REDTab: A Relation Extraction Dataset for Knowledge Extraction from Web Tables

Siffi Singh [1], Alham Fikri Aji [2], Gaurav Singh [3,*], and Christos Christodoulopoulos [2]
[1]Amazon AWS AI, [2]Amazon Alexa AI
{siffis,afaji,chrchrs}@amazon.com
[3]Binance
gaurav.singh@binance.com

Abstract

Relational web-tables are significant sources of structural information that are widely used for relation extraction and population of facts into knowledge graphs. To transform the web-table data into knowledge, we need to identify the relations that exist between column pairs. Currently, there are only a handful of publicly available datasets with relations annotated against natural web-tables. Most datasets are constructed using synthetic tables that lack valuable metadata information, or are limited in size to be considered as a challenging evaluation set. In this paper, we present REDTab, the largest natural-table relation extraction dataset. We have annotated ~9K tables and ~22K column pairs using crowd sourced annotators from MTurk, which has 50x larger number of column pairs than the existing human-annotated benchmark. Our test set is specially designed to be challenging as observed in our experiment results using TaBERT.

1 Introduction

Web-tables contain a lot of knowledge (Cafarella et al., 2008; Wang and Hu, 2002) that can be utilized for various tasks such as question answering (Chakrabarti et al., 2020; Zhang et al., 2020), building knowledge graphs (Sekhavat et al., 2014; Wang et al., 2012), web-search (Sun et al., 2016; Kopliku et al., 2011), data-mining (Gatterbauer et al., 2007) and so on. Relation Extraction (RE) is one way to extract table information by capturing relations between two table columns (Figure 1). However, the current RE datasets available are either synthetic, or very small in size. In this paper, we propose a new human-annotated dataset for the evaluation of RE models.

The existing benchmark datasets for column pair RE are T2Dv2 and SemTab. However, T2Dv2 (Ritze and Bizer, 2017) only contains 236 tables and 435 column pairs annotated with 118 relations where about 50% of relations have only 1 column pair sample. The highest number of samples for a relation is only 28. Hence, due to its small size and less diversity in column pairs per relation, it is difficult to be treated as a reliable evaluation set. In contrast, the SemTab 2020 challenge (Jiménez-Ruiz et al., 2020a) provides a large benchmark dataset, but synthetically generated, which is arguably not representative of the real web-tables. Other table-based datasets (Limaye et al., 2010; Efthymiou et al., 2017; Zhang, 2017; Kacprzak et al., 2018) are designed for a binary relation extraction task.

REDTab[1] is a human annotated dataset consisting of 9,149 tables and 22,236 column pairs with 23 relations. The table was collected from Web Data Commons (WDC) (Lehmberg et al., 2016) relational tables for Music and Literature domains. These domains are two of most popular domains for question answering (Rajpurkar et al., 2016; Mihaylov et al., 2018; Kočiskỳ et al., 2018; Serban et al., 2016). The dataset also contains metadata such as table title, page title, and text before and after the table as extra context. The key differences between our proposed dataset and other benchmarks are shown in Table 1. We also provide benchmark results on REDTab using state-of-the-art table embedding approach called TaBERT (Yin et al., 2020). We get an F1-score of 0.58, which shows that our dataset is challenging and the performance on the test set can be further improved.

*Work done while at Amazon

[1]We publicly release the REDTab dataset at https://github.com/alexa/alexa-dataset-redtab.

Proceedings of the 29th International Conference on Computational Linguistics, pages 2319–2327
October 12–17, 2022.

Dataset	Tables	Col pairs	Relations	Annotation
T2Dv2	236	435	118	Human
SemTab	34,294	1,31,289	-	Synthetic
REDTab	**9,149**	**22,236**	23	**Human**

Table 1: Key Differences in REDTab over other RE benchmark datasets. The number of relations in SemTab is unknown as it is part of the blind test set and not provided by the organisers. Note that REDTab has 23 relations covering diverse set of column pairs for two domains.

2 Data Collection and Annotation

In this section, the construction of the REDTab dataset is described in detail. The discussion is divided into three subsections: Data Source, Annotation Protocol and Dataset Validation.

2.1 Data Source

REDTab dataset is constructed from Web Data Commons or WDC (Lehmberg et al., 2016) web-tables. Table selection is done by randomly sampling ~20k tables from WDC and manually identifying the suitable Music and Literature domain column headers. Based on these column headers, the tables for each domain are extracted from the entire WDC corpus. Approximately 5k tables are randomly sampled for both domains.

For each table, we also preserve metadata containing title, page title, as well as text before and after the table to provide more context. From the metadata information, we obtain the 'keyColumnIndex', that denotes the main column (referred to as Subject column). This Subject column will then be paired with other columns from the same table to form a column pair.

The relations in the column pairs are manually identified by inspecting the table headers. Our relation names hold more meaning than the relations from existing datasets. For example, instead of using "author", we use "is the author of", since it encodes more information than just "author". The full verb phrase construction encodes the argument order (Subject to the left, Object to the right), "Is" implies it is currently true, "the" implies uniqueness (compared to "is an author of"). It makes the relations far more human readable and interpretable without needing additional tooling to present the information to annotators. It also made the understanding connection to natural language questions easier with more naturalistic phrasing. The number of relations in our dataset is data-driven, this means, if we had encountered more relations in the data, we would have added them to the set of relations presented to annotators. Additionally, our dataset has higher number of relations per domain as compared to existing human-annotated datasets. Note that our dataset cannot be used to build a knowledge graph from scratch but rather to serve as an input to an information extraction system, that, with the right grounding (mapping), can also be used for knowledge graph construction. See Table 3 for examples of relation names. After filtering, there are 36 and 34 relevant relations in Music and Literature domains respectively.

[2] https://music.apple.com/ca/artist/jenny-silver/189196098/see-all?section=top-songs, accessed on Feb 16th 2022

2.2 Annotation Protocol

REDTab dataset is constructed by crowd source annotators from Amazon Mechanical Turk (MTurk) (Chen et al., 2011). The qualification requirement is set to locations of United States, United Kingdom, Australia, Ireland, Canada, and Singapore, and HIT Approval Rate of 98% with number of approved annotations as at-least 1000.

The annotation process is as follow. A table is shown to annotator. Then, the Subject column is paired with all other columns in that table. For each pair, the annotator must annotate the relation of the pair. The interface is shown in Figure 2. Let us consider an example, where <Name, Artist, Time, and Price> are the table columns and 'Name' is the Subject column. The annotators must annotate the relation between (Name - Artist), (Name - Time), and (Name - Price).

The table is displayed with the relation names specific to the table's domain. There is an additional 'UNKNOWN' relation that is provided to be selected in a scenario when the two column do not have a relation, or, when none of the given relations are applicable.

The annotator must also decide the directionality of the relation, i.e. whether it is 'Tinku' `<is a song performed by>` 'Holy Near & Inti Illimani', or 'Holy Near & Inti Illimani' `<is a song performed by>` 'Tinku' (as shown in Figure 2). Annotators are also provided with a space to leave their comments which has been useful to improve the guidelines and user experience.

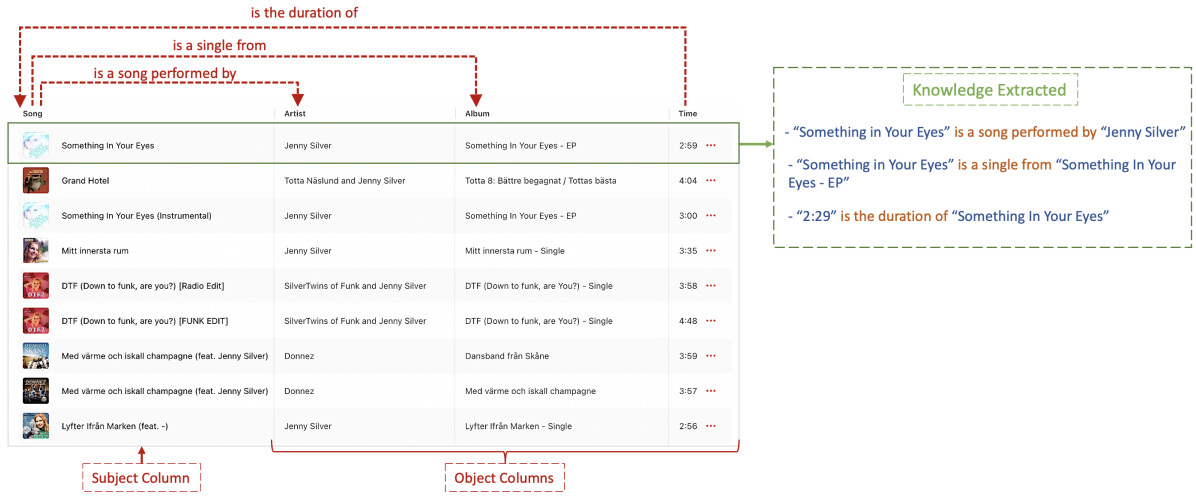

Figure 1: Example table[2] with columns 'Song', 'Artist', 'Album' and 'Time'. The main column, called the Subject column, is 'Song'. The relations between column pairs are labeled at the top. On the right, the extracted knowledge from one row of the table is shown.

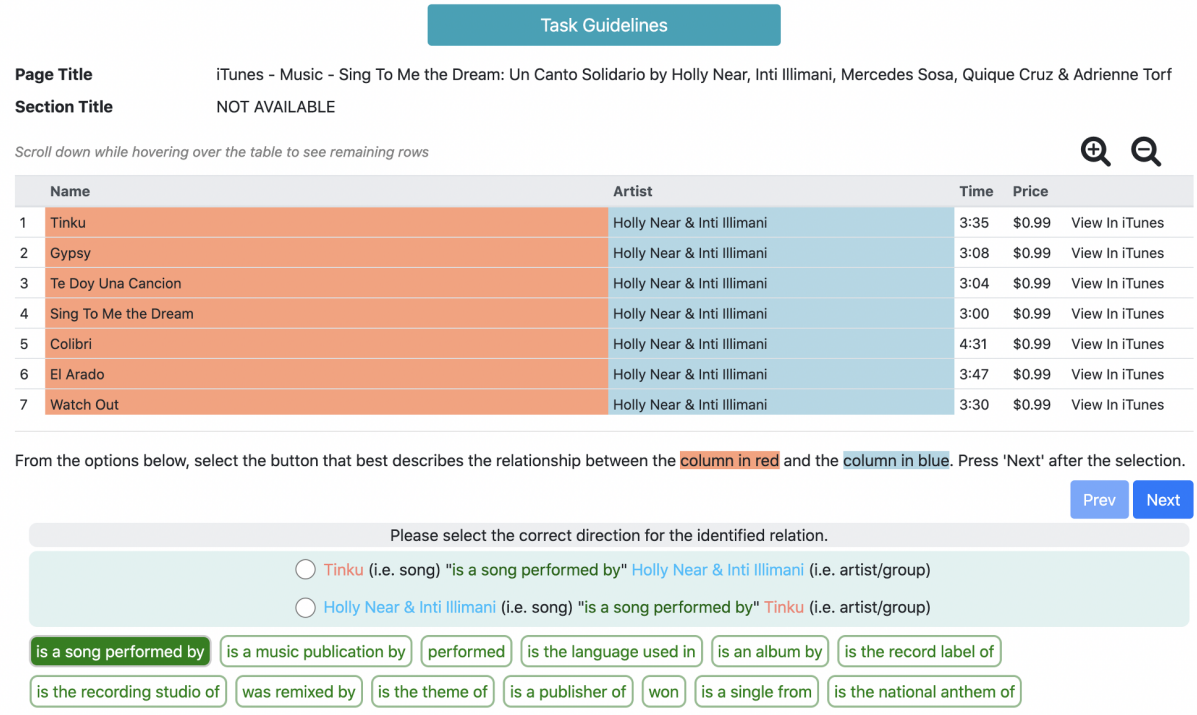

Figure 2: The annotation user interface for creating REDTab dataset. On top the 'Task Guidelines' are given. The 'Task Guidelines' is followed by page title, table title of the table, along with all the row values of table. A list of relations are provided, and upon selection of a relation (example shown 'is a song performed by'), the two options are provided showing the directionality of Subject column.

2.3 Dataset Validation

There are two main stages in the validation phase. The first stage is analysis of annotated data and the second stage is manual inspection to eliminate incorrect annotations.

Dataset Analysis: Annotations are carried out in several batches. Each table is annotated by 3 annotators. The first analysis is counting the number of annotated tables for each batch. This is followed by calculating the inter-annotator agreement using the Fleiss' kappa Score (McHugh, 2012). For Music,

the inter-annotator agreement is 0.75 whereas for Literature, the agreement is 0.40. The low agreement value in Literature is attributed to tables that are mistakenly identified into the Literature domain. For instance, 'Publisher' is a column header for tables in both Game and Literature domains. As 'Publisher' is also a column header used to filter tables as Literature, some of the tables from Game domain are mistakenly included during the filtering process. These tables create confusion for annotators as the remaining column pairs in the table cannot be identified with the given relations list.

In the final set, only those column pairs are included which have the same relation selected by at least two annotators. The co-occurrences of relations are then looked at to know which relation pairs are most commonly selected. For instance, 'is a publisher of' and 'is the record label of' have more than 50 co-occurrences in a batch of 1000 tables.

The annotated set consists of 10,284 tables and 39,514 column pairs. The number of selected relations is 52 as compared to 70 relations in the original set given to the annotators.

Dataset Cleanup: In this stage, the dataset is manually inspected. From the annotated tables, the header and subject column combination are collected. E.g. <Author, Title, Description, Date, Rank, 1>. This is called a 'template'. The given example is a table with five columns and a subject column of index '1'. Such a template is collected for all tables.

Considering these unique templates, a total of 1026 different sets are identified among 10,284 tables. The annotations for these tables are verified to have the correct relation and directionality. In the verification process, the contents of the table column are also taken into consideration. Using this approach, a mapping between a template and its corresponding annotation is created. The mapping is further used to update the annotations and create a clean dataset. The relations removed as part of cleanup are the ones which have incorrect annotations, or cannot be split into train-test set (see Section 3.1). The final dataset is created by collecting majority annotations, filtering the 'UNKNOWN' relation selections, and removing the incorrect annotations.

Dataset	# relations	# train	# test
REDTab-standard	23	4682	17554
REDTab-simple	10	4431	16176

Table 2: Statistics from the train and test set in REDTab Standard and REDTab simple dataset.

3 The Resulting Corpus

REDTab consists of 9,149 tables and 22,236 column-pairs annotated with 23 relations for Music and Literature domain. The overall time spent for annotation was ~300 hours and the total cost is ~6000 USD (i.e. 20 USD per hour). The number of tables in REDTab is an order of magnitude higher than T2Dv2 as shown in Table 1. Similarly, the number of column pairs in REDTab is two orders of magnitude higher than T2Dv2. Although, T2Dv2 has more relations, the number of column pair samples per relation is very small.

3.1 Train-Test Split

On both T2Dv2 and REDTab, it is observed that most of the time, the relation can be easily determined by looking at the column headers. For example, the column pair ('Title', 'Author') is annotated as `<is the author of>` all of the time. Only few column headers have some ambiguity to it. Therefore, if the train and test data is split randomly, the relation can be determined by simply memorizing the header. This hypothesis is tested and a simple header memorization is shown to achieve high accuracy without looking at any other context.

In order to construct a challenging test set that can be used to benchmark model's capability on understanding structured context, the training and testing data is split such that they do not share the same header pairs. For example, given a relation `<is a book about>`, the possible column header pairs are for example ('Title', 'Subjects'), ('Name', 'Description'), ('Title', 'Synopsis') etc. Data with ('Title', 'Subjects') are assigned as training whereas ('Name', 'Description') are assigned as testing. Specifically, for every relation, we group all data based on their header pair. However, some relations were annotated to only have 1 unique header pair, therefore these relations are excluded from the dataset.

3.2 REDTab dataset

Our final set consists of 23 relation classes, divided into training and testing split as shown in Table 3.

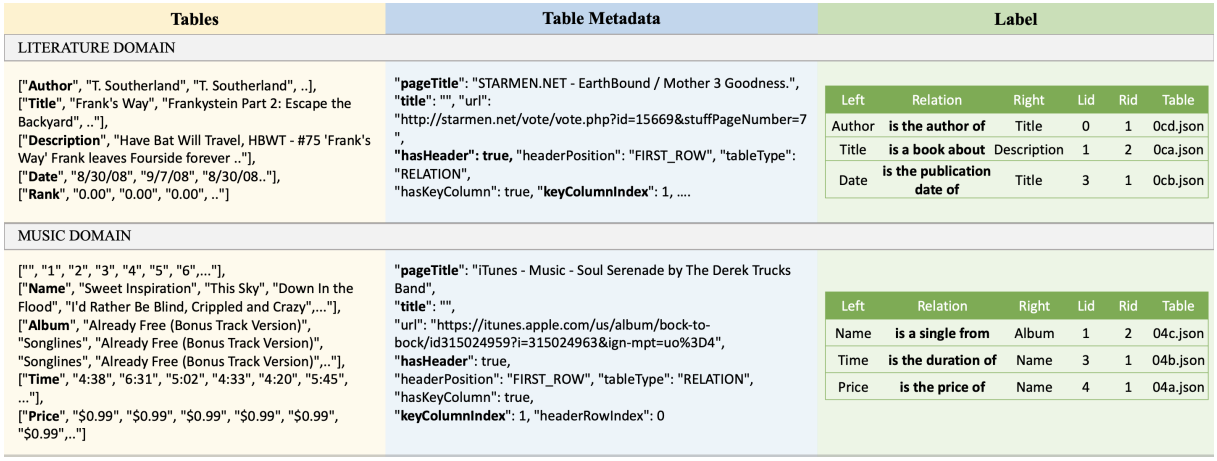

Figure 3: The first column shows the Table row data. The column headers are highlighted to show the column pairs. The second column is Table Metadata which includes 'keyColumnIndex' i.e. the index of the Subject column. In the first example, 'Title' is the Subject column with index 1. The third column shows the human-annotated relation label. The label has 'Left' i.e. left entity column, 'Relation' i.e. annotated relation and 'Right' i.e. right entity column. The result also contains the indexes of the two columns, direction information, and table filenames.

Note that some of the tail relation classes has minimal training or testing examples, which are the difficult samples. However, we argue that such cases reflect the real-world scenario. Therefore, we decide to release a simpler version of REDTab, which only focus on the top 10 most common relations. The statistics of our REDTab dataset can be seen in Table 2.

The train-test splits for both REDTab standard and simple category are publicly released for researchers to evaluate. An example annotation from REDTab can be seen in (Figure 3). The 'Left' and the 'Right' column is based on the directionality selected by annotators.

4 Experimental Setup

We use TaBERT (Yin et al., 2020) as a baseline in our dataset. TaBERT is a pre-trained language model that jointly learns representations for natural language sentences and tables. TaBERT represents each cell by the column name, column datatype and cell value, which is followed by using the Transformer (Vaswani et al., 2017; Devlin et al., 2018) model to generate row-level vectors. To allow for information flow across cells of different rows, TaBERT uses a vertical self-attention mechanism. TaBERT introduces two pre-training tasks, Masked Column Prediction and Cell Value Recovery to generate powerful contextualised representations. TaBERT has been tested only on semantic parsing tasks.

TaBERT model requires 4 inputs: Header, data, ID, and context. We feed the table header and row data as header and data respectively. The table title is assigned as the TaBERT ID, while the page title is assigned as the TaBERT context. We then concatenate the column embeddings for the current input, i.e. `concat(embedding of the left column, embedding of the right column)`. This is connected with a classification layer to predict the relation (see Figure 4).

Relation	# training	# testing
is a song performed by	1065	2702
is the author of	669	3322
is the price for	629	2197
is the duration of	549	2523
is the publication date of	427	2098
is the record label of	301	549
is a single from	271	1355
is a publisher of	202	430
is a book about	198	889
was written on	120	111
is an album by	81	624
is the composer of	47	17
is the genre of	42	175
is a narrative set in the location	39	248
is the isbn of	13	73
is the date of creation of	7	30
is the genre of music played by	5	3
is the number of pages in	4	35
is published as part of	4	39
is an edition of the published work	4	123
was released on an album at timepoint	2	3
is written in the language	2	7
wrote the lyrics to	1	1

Table 3: The relation names in REDTab standard dataset. We show the number of column pairs corresponding to each relation in the train and test set.

Model	REDTab-standard			REDTab-simple		
	P	R	F1	P	R	F1
Majority class	0.008	0.043	0.013	0.020	0.100	0.034
TaBERT base	0.628	0.621	0.580	0.876	0.900	0.883
TaBERT large	0.677	0.637	0.593	0.928	0.866	0.871

Table 4: Baseline performance showing macro precision, recall and F1-score, using REDTab dataset.

We experiment on both base and large TaBERT model. Our model is fine-tuned on 4 GPUs with a batch-size of 16 for 3 epochs. The model is trained with AdamW optimizer (Loshchilov and Hutter, 2019). Each experiment takes less than an hour to finish.

5 Result

5.1 Baseline

We start by exploring a TaBERT model as a baseline for both of our standard and simple REDTab set. Our baseline utilize all of the available context: table header, table data, and titles. We also introduce a "majority" baseline, where the model simply predicts the majority relation class all the time, which is "is the author of". The majority baseline is used as our lower-bound performance.

As shown in Table 4, our model achieved a F1 score of 0.58 on REDTab-standard, showing room for improvement for future studies on this dataset. Table 5 also shows the F1-score per relation. Better performance is achievable in REDTab-simple, since this set is designed to be easier compared to its standard counterpart. TaBERT large does not significantly outperform TaBERT base, despite using more computational resource. For further experiments, we use TaBERT base architecture.

5.2 Ablation Study

In this part, we perform an ablation study to see the effect of each data feature on performance. Specifically, we divide REDTab data into 3 contexts: 1) The header context, which includes the table header itself, 2) The row data context, which includes every row data of the table, and, 3) The title context, which includes the table and web page titles. This experiment uses REDTab-standard. Figure 4 illustrates the contexts.

Our baseline feeds all 3 contexts to TaBERT. Here, we remove one or more contexts by replacing it with empty strings before passing it to TaBERT. The experiment result can be seen in Table 6.

Header context is the most prominent feature.

Relation	P	R	F1
is a single from	0.95	1	0.97
is the duration of	0.83	0.98	0.9
is the price for	0.77	0.49	0.6
is the author of	0.74	0.88	0.8
is a book about	0.5	0	0
is the publication date of	0.98	1	0.99
is a publisher of	0.52	0.27	0.36
is a narrative set in the location	0.85	0.89	0.87
is a song performed by	1	0.98	0.99
is an album by	0.97	1	0.98
is the record label of	0.62	0.99	0.76
is an edition of- the published work	1	0.18	0.3
is the isbn of	0.97	0.99	0.98
is the genre of	0.7	0.71	0.71
was written on	0.99	0.66	0.79
is the composer of	0.89	0.94	0.91
is published as part of	0	0	0
is the date of creation of	0.01	0.2	0.02
is the number of pages in	0.51	1	0.68
is the genre of- music played by	0	0	0
is written in the language	0.58	1	0.74
was released on an album at timepoint	0	0	0
wrote the lyrics to	0.25	1	0.4

Table 5: Baseline performance showing precision, recall and F1-score per relation using TaBERT base on REDTab-standard dataset.

Our model can perform as good as the baseline only by looking at the table header (Header only). Similarly, removing the header significantly reduces the performance (No header). We argue that many table headers are representative of their content, therefore in most cases, it can be utilized to determine the relation of two columns without additional context.

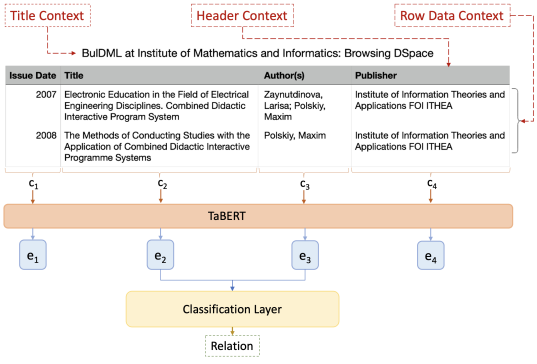

Figure 4: TaBERT for column-pair relation extraction. $\{c_1, c_2..\}$ are the columns, $\{e_1, e_2..\}$ are the TaBERT column embeddings for each column. The diagram shows title context, header context, and the row data content. Relation is classified between c_2 and c_3 in the table.

Config Name	Context			Performance		
	header	row-data	title	P	R	F1
Baseline	✓	✓	✓	0.628	0.621	0.580
Header only	✓			0.633	0.636	0.571
Row only		✓		0.417	0.318	0.290
Title only			✓	0.140	0.158	0.115
No header		✓	✓	0.392	0.367	0.336
No row	✓		✓	0.599	0.632	0.559
No title	✓	✓		0.664	0.644	0.615
Zero context				0.075	0.108	0.075

Table 6: Baseline results showing impact of metadata information on macro precision, recall and F1-scores for relation extraction using REDTab dataset.

We see a significant drop in performance (0.29 F1-score) when the model has to rely on row data alone. Therefore, it is challenging to understand a table data if we only see the data without additional context. Title context is the least prominent feature, as the model performed poorly (0.115 F1-score) if only the table and page titles are shown. In contrast, we gain performance over the baseline by removing the title context (No title), suggesting that the title feature might be too noisy if we have a better context such as table header or row data.

Interestingly, model with zero context (i.e. only see an empty table) can perform better than our lower-bound model (majority relation class). Therefore, the model is capable of guessing the relation even without any explicit context at all.

5.3 Column Ordering Bias

We also observe in our experiments that the table content can be indirectly inferred by the column location. For example, the first column is mostly expressing the main subject of the table, such as album name, whereas the later columns are mostly expressing the information of columns, such as price, or date. Therefore, the relation between two columns can be inferred by their position alone. For example, the relation between the 2^{nd} and the 1^{st} column is 'is a song performed by' 77% of the time, or the relation between the 3^{rd} and the 1^{st} column is 'is the price of' 68% of the time. We find that a model can exploit this implicit information, thus can gain some performance in zero context scenario.

To remove column ordering bias, we pre-process the data by shuffling the columns. As shown in Table 7. The zero-context model on shuffled columns has similar performance as majority.

Config Name	Performance		
	P	R	F1
Zero context	0.075	0.108	0.075
Zero context - shuffled column	0.018	0.052	0.025
Majority class	0.008	0.043	0.013

Table 7: Zero context performance on REDTab dataset by shuffling the column and removing the column ordering bias.

5.4 Discussion and Future Work

In REDTab, we see that some relations only have few examples, it is in fact reflective of the real-world scenario where there are column pairs which occur more frequently with one another, while some column pairs are uncommon and hence fewer tables contain these samples. Our baseline system is not handling these tail relation cases well, demonstrated by the low F1 score in REDTab-standard scenario. For future, we should investigate a model, that is capable of handling such tail cases. We have also observed that row data in tables alone cannot give good performance, and use of metadata improves results. In our experiments, we have explored the use of metadata such as headers, table title and page title. In future, we can explore models that can make effective use of other metadata information present with tables in the dataset (e.g. text before table) which might show further improvement in performance. In future, we want to add relation mappings to public knowledge graphs, expand REDTab dataset to include several other domains and cover diverse set of tables with variety of column pairs.

6 Related Work

We briefly describe most major benchmark datasets that contain relation annotations for web-tables.

SemTab Challenge

The SemTab challenge was launched to encourage comparisons between different automated table alignment techniques over a benchmark dataset. In 2019, SemTab (Jiménez-Ruiz et al., 2020a) released 3 tasks, namely: 1) Column-Type Annotation (CTA), 2) Cell-Entity Annotation (CEA), and 3) Columns-Property Annotation (CPA). They build their dataset by automatically generating labelled tables from facts stored in the the DBpedia (Auer et al., 2007) Knowledge Graph (KG). Despite its large size, the dataset consists of synthetically generated tables, which suffer from draw-

back such as: lack of associated metadata, noisiness, heterogeneity, incompleteness and ambiguity in the data generation process. These drawbacks are not unique to synthetic tables, but can cause them to significantly diverge from human generated web-tables. In 2020, they released a new dataset called Tough Table (2T) dataset (Jiménez-Ruiz et al., 2020b). It consists of a mix of small and large catalog-like tables to make the dataset more challenging. However, unlike other datasets, it also consists of real web-tables. Unfortunately though, it does not contain annotations for the relation extraction task (CPA). Also, a very small number of real web-tables are contained in this dataset, while the majority of other tables are still synthetic.

TURL

TURL (Deng et al., 2020) consists of Wikipedia tables annotated via distant supervision based on overlap of facts with Freebase KG (Bollacker et al., 2008). However, our own investigations revealed that these relations and tables were too easy, and therefore did not represent the real task of relation extraction from web-tables. In addition to our investigations, the model in the paper achieves extremely high F1 scores for the task, further evidence that the tables and relations are simple. Also, the method can not be used to create a larger dataset of more tables and relations, since the overlapping relations have already been exploited. Additionally, it would be harder to find direct overlaps of unpopular relations or facts in the Freebase KG.

T2Dv2

T2Dv2 (Ritze and Bizer, 2017) is a dataset of real web-tables manually labelled against DBpedia KG. To the best of our knowledge, it is the only other publicly available real web-tables dataset that was manually labelled. Of all the other web-tables datasets, this one is the closest to ours because it was manually created and consists of real web-tables. However, its one drawback is that it consists of only 236 web-tables with annotated relation. In contrast, our dataset consists of over 9K manually annotated web-tables. Therefore, our dataset is the largest publicly available and manually annotated dataset.

Additionally, there are other table alignment datasets created in the past (Limaye et al., 2010; Efthymiou et al., 2017). But these datasets either consists of very few tables or the tables were annotated using automated distant supervision techniques, making them unreliable for task evaluation. For example, Limaye et al. consists of 400 manually annotated Web-tables with entity, class, and property-level correspondences, where single cells are mapped to entities and Efthymiou et al. only mapped entities in tables to DBpedia KG.

7 Conclusion

We present REDTab, the largest natural web-tables dataset for column pair relation extraction. The dataset is annotated by crowd sourced annotators from MTurk. REDTab includes more than 9K tables and 22K column pairs making it the largest human-annotated relation extraction dataset to our knowledge. We evaluate state-of-the art table embedding model TaBERT and find that our dataset is challenging over the strategically created train-test split set. Our final set contains more diverse set of column pairs per relation that is ideal for testing models. Furthermore, our analysis shows that predicting relations solely based on table row data is challenging and use of some metadata information significantly improves the performance. We expect this dataset can contribute to facilitate further progress in the field of information extraction research.

References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

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

Kaushik Chakrabarti, Zhimin Chen, Siamak Shakeri, and Guihong Cao. 2020. Open domain question answering using web tables. *arXiv preprint arXiv:2001.03272*.

Jenny J Chen, Natala J Menezes, Adam D Bradley, and T North. 2011. Opportunities for crowdsourcing research on amazon mechanical turk. *Interfaces*, 5(3):1.

Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2020. Turl: Table understanding

through representation learning. *arXiv preprint arXiv:2006.14806*.

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*.

Vasilis Efthymiou, Oktie Hassanzadeh, Mariano Rodriguez-Muro, and Vassilis Christophides. 2017. Matching web tables with knowledge base entities: from entity lookups to entity embeddings. In *International Semantic Web Conference*, pages 260–277. Springer.

Wolfgang Gatterbauer, Paul Bohunsky, Marcus Herzog, Bernhard Krüpl, and Bernhard Pollak. 2007. Towards domain-independent information extraction from web tables. In *Proceedings of the 16th international conference on World Wide Web*, pages 71–80.

Ernesto Jiménez-Ruiz, Oktie Hassanzadeh, Vasilis Efthymiou, Jiaoyan Chen, and Kavitha Srinivas. 2020a. Semtab 2019: Resources to benchmark tabular data to knowledge graph matching systems. In *European Semantic Web Conference*, pages 514–530. Springer.

Ernesto Jiménez-Ruiz, Oktie Hassanzadeh, Vasilis Efthymiou, Jiaoyan Chen, Kavitha Srinivas, and Vincenzo Cutrona. 2020b. Results of semtab 2020. In *CEUR Workshop Proceedings*, volume 2775, pages 1–8.

Emilia Kacprzak, José M Giménez-García, Alessandro Piscopo, Laura Koesten, Luis-Daniel Ibáñez, Jeni Tennison, and Elena Simperl. 2018. Making sense of numerical data-semantic labelling of web tables. In *European Knowledge Acquisition Workshop*, pages 163–178. Springer.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Arlind Kopliku, Karen Pinel-Sauvagnat, and Mohand Boughanem. 2011. Attribute retrieval from relational web tables. In *International Symposium on String Processing and Information Retrieval*, pages 117–128. Springer.

Oliver Lehmberg, Dominique Ritze, Robert Meusel, and Christian Bizer. 2016. A large public corpus of web tables containing time and context metadata. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 75–76.

Girija Limaye, Sunita Sarawagi, and Soumen Chakrabarti. 2010. Annotating and searching web tables using entities, types and relationships. *Proceedings of the VLDB Endowment*, 3(1-2):1338–1347.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.

Mary L McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia medica*, 22(3):276–282.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Dominique Ritze and Christian Bizer. 2017. Matching web tables to dbpedia-a feature utility study. In *EDBT*.

Yoones A Sekhavat, Francesco Di Paolo, Denilson Barbosa, and Paolo Merialdo. 2014. Knowledge base augmentation using tabular data. In *LDOW*.

Iulian Vlad Serban, Alberto García-Durán, Caglar Gulcehre, Sungjin Ahn, Sarath Chandar, Aaron Courville, and Yoshua Bengio. 2016. Generating factoid questions with recurrent neural networks: The 30m factoid question-answer corpus. *arXiv preprint arXiv:1603.06807*.

Huan Sun, Hao Ma, Xiaodong He, Wen-tau Yih, Yu Su, and Xifeng Yan. 2016. Table cell search for question answering. In *Proceedings of the 25th International Conference on World Wide Web*, pages 771–782.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Jingjing Wang, Haixun Wang, Zhongyuan Wang, and Kenny Q Zhu. 2012. Understanding tables on the web. In *International Conference on Conceptual Modeling*, pages 141–155. Springer.

Yalin Wang and Jianying Hu. 2002. Detecting tables in html documents. In *International Workshop on Document Analysis Systems*, pages 249–260. Springer.

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. Tabert: Pretraining for joint understanding of textual and tabular data. *arXiv preprint arXiv:2005.08314*.

Xingyao Zhang, Linjun Shou, Jian Pei, Ming Gong, Lijie Wen, and Daxin Jiang. 2020. A graph representation of semi-structured data for web question answering. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 51–61.

Ziqi Zhang. 2017. Effective and efficient semantic table interpretation using tableminer+. *Semantic Web*, 8(6):921–957.