Project 15: Semantic Verification and Evidence Finding with Tables

Ankur Gupta[1], **Preeti Menghwani**[2], **Jaya Srivastava**[3]
[1]180109 [2]180550 [3]180326
[1]EE, [2]EE, [3]ME
{ankugupt, priti, jayasri}@iitk.ac.in

Abstract

In this report, we describe Task 9 of SemEval 2021, which involves Scientific Table understanding and proper interpretation of surrounding data. Tables are an easy and concise way to convey important information in documents using rows and columns. The organisers propose two subtasks for Table understanding first, statement verification, second, evidence finding in tables. We make use of the dataset provided to us by the organizers and the popular Wikipedia tables dataset, TABFACT (Wenhu Chen and Wang, 2020), presently for our task. We have produced results on the state-of-the-art models and our proposed model for for both the datasets and subtasks. The Data and Code of the Dataset are provided in : https://github.com/jaihonikhil/Group-15

1 Introduction

Textual Inference, also known as natural language inference (Bowman et al., 2015), plays an important role in the study of natural language understanding and semantic representation. Textual Inference is well explored. However, the current works are mainly dealing with unstructured evidence like comprehension (Dagan et al., 2005). Verification under structured and semi-structured evidence, such as tables, graphs, and databases, remains unexplored. Though recently, there have been work on Tabular Inference problems (Zhong et al., 2020; Cho et al., 2018; Sun et al., 2018; Wenhu Chen and Wang, 2020; Eisenschlos et al., 2020; Pasupat and Liang, 2015; Wang et al., 2018), evidence finding is still an unexplored area. Through this task, we aim to solve the problem of 1. *Statement verification* and 2. *Evidence finding* using tables and emphasize the importance of structured evidence in a document summarizing, analysis, fact verification, and fake news detection.

The current models majorly use semantic parsing approaches to create logical forms. On the other hand, we have decided to follow the works of Eisenschlos et al., 2020 and Herzig et al., 2020 to encode tables with BERT base model and predict inference decision. We propose a new approach inspired by the previous work on question answering systems and table pruning methods for evidence finding.

The experimentation has been done both on our dataset and the TABFACT Dataset. This paper will be going through our relevant works, experimentation outcomes and, our proposed approach in detail.

2 Problem Defintion

The proposed task by the organisers has two subtasks to explore table understanding:
A. Table Statement Support Does the table support the given statement?
It is a multi-class classification problem, to assign one of the label to each input statement using information from linking table and surrounding data:

- ***Fully Supported***: Statement is supported by data found within the table.
- ***Refuted***: Statement is contradicted by table.
- ***Unknown***: Not enough information in table to assess statement veracity.

Body Sensation	Agoraphobic	Pleasant
number	museum	lovely
palpitation	shop	happiness
heartbeat	boat	Joyous

Statement	Label
Palpitation is a bodily sensation	Supported
Joyous and boat have same strength	Unknown
Lovely is an agoraphobic situation	Refuted

Table 1[1]: A sample table and statement with correct results for subtask A and B. violet: Supported, red: Refuted, grey: Unknown

B. Relevant Cell Selection Which cells in the table provide evidence for the statement?
Each table cell is labelled any one in the following label depending upon its involvement in reasoning of a statement:

- ***Relevant***: the cell must be included.
- ***Ambiguous***: the cell is allowed to be either included or not included.
- ***Irrelevant***: the cell must not be included.

3 Related Work

TAPAS (Herzig et al., 2020) proposes an approach to question-answering over tables without generating logical forms. It uses weak supervision and predicts the answer by selecting table cells and optimally applying aggregation operator to the selected cells. The input to the model is a question and a flattened table. The statement and the table are tokenized into word pieces and concatenated using the `[CLS]` and `[SEP]` tokens.

In addition to BERT embeddings extra row, column and rank embeddings are added with two classification layers for selecting cells and predicting a corresponding aggregation operator. Since TAPAS handles tables as strings, it fails to capture very large tables or databases containing multiple tables, as these do not fit into the memory. The expressivity of the model is limited to a form of aggregation over a few cells of the table hence, questions with multiple aggregations are not handled properly.

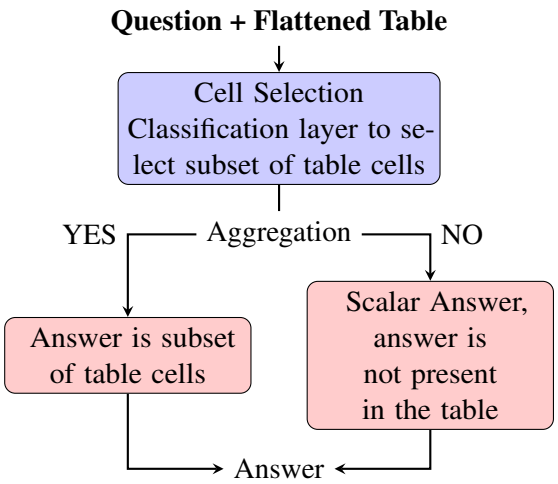

Figure 1: Flow chart explaining the functioning of TAPAS model using the two classification layers

Recently Eisenschlos et al., 2020 adapted TAPAS to classify statements based on tables as ENTAILED/REFUTED. They introduced two intermediate pre-training tasks learned from the MASKLM model, one being based on counterfactual statements generated by creating one positive and one negative from every relevant statement extracted from Wikipedia statements and tables and the other, based on synthetic statements that generates a sentence by sampling from a set of logical expressions, by defining context-free grammar.

Let S and T represent the statement and the table respectively which are given as input to the model and E_S and E_T represent their corresponding input embeddings. The sequence of statement and the table given by $E = [E_{[CLS]}; E_S; E_{[SEP]}; E_T]$ is passed through the transformer model which we denote by f and a contextual representation is obtained for every token. The entailment probability P(S|T) is modeled using a single hidden layer neural network obtained by computing the output of `[CLS]` token:

$$P(S|T) = \texttt{MLP}(f_{[CLS]}(E))$$

They have also used table pruning using Heuristic exact match (HEM) to handle large size tables. In this method, the columns are ranked by a relevance score and added in order of decreasing relevance. Columns that exceed the maximum input length are skipped.

TABFACT(Wenhu Chen and Wang, 2020) presents a dataset with 16k Wikipedia tables and 118k human-annotated natural language statements,labeled as either ENTAILED or REFUTED. TABFACT also presents two models TABLE-BERT and LPA. Table-BERT views the task as natural language inference and doesn't use any special embedding to encode the table. It linearizes the table into a sequence and then concatenates the table cells using horizontal scan, adopting a simple natural language templates to transform a table into a "somewhat natural" sentences . Next it feeds into the 12 layer BERT model and computes the matching probability between the (table, statement) pair,classifying it as ENTAILED statement if it is greater than 0.5. LPA parses statements into programs and executes them against the tables using weak supervision to obtain the returned binary value for verification.

[1]based on example 0001.html with light edits

4 Corpus/Data Description

The tables are extracted from open-source scientific articles using APIs provided by Science Direct. The Dataset contains two kinds of data, one being **Manually** annotated which is also validated by another round of validation. Other is the **Auto-Generated** Annotations that are the statements auto-generated using random paraphrase and table understanding parser. The input consists of tables, table caption, Legend Text, footnotes, statements, labels and cell labeling in XML structured format. Each XML file consists of one or more tables.

Table 2 below shows the basic statistics of the data provided to us by the organizers. It contains a total of **1980** Autogenerated Tables and **981** Manually generated Tables, the Autogenerated Tables being **more complex** than the Manually Annotated Tables. A few errors like Classification label errors, Grammatical errors in natural statements about the table, and few Autogenerated statements with no labeling due to potential ambiguity are there in the Dataset. Only the Autogenerated XML files provide labeling to individual cells which are specific to Task 2.

The test set is expected to be only manual annotations that have been verified and will require evidence for all cells if participating in the evidence finding subtask.The Data was preprocessed and cleaned before feeding it into the models. During the Data Cleaning, statements with no labels were removed and subcolumns were interpolated to handle the complex header structure of the tables provided. For subtask 1, there are no sentences with the unknown label in the train, dev and test split hence currently, UNKNOWN label is not taken into consideration for evaluation purpose.

Manual	**Train(Avg)**	**Dev(Avg)**	**Test(Avg)**
Total	3549(4.5)	501(5)	456(4.65)
Entail	2194(2.8)	318(3.1)	306(3.12)
Refute	1355(1.7)	183(1.8)	150(1.53)
Autogen.	**Train(Avg)**	**Dev(Avg)**	**Test(Avg)**
Total	145539(91.5)	16967(87)	16839(86.8)
Entail	74862(47.8)	8696(44.59)	8578(44.2)
Refute	70677(44.4)	8271(42.41)	8261(44.5)

Table 2: Basic statistics of the Autogenerated and Manual Annotations.

From the Above table, we can infer that the data provided is a bit skewed with Entailed statements roughly being 1.5 times the Refuted statement in case of Manual Annotation. In addition to that the Manually Annotated data is fairly simple, with less number of statements and tables with decent size unlike the Autogenerated ones with complex tables and a large number of statements per table.

We also used **TABFACT** dataset (Wenhu Chen and Wang, 2020), as to perform experiments earlier. It is a large-scale dataset with 16k Wikipedia tables as the evidence for 118k human-annotated natural language statements, with labels as ENTAILED/REFUTED, combining semantic-level understanding and Symbolic Reasoning, in which the verification requires symbolic execution on the table structure.

5 Proposed Approach

5.1 Fact verification from table information

The model architecture for manually annotated dataset is an ensembled model using the fine-tuned TableBERT and TAPAS. TableBERT (Devlin et al., 2018) has been used by replacing the BERT base model by pretrained SciBERT(Beltagy et al., 2019) and then training on the given dataset. The input to the model is linearized table concatenated with the statement using `[SEP]` and `[CLS]` tokens and the sequence relationship logits are obtained as outputs which are further used for class prediction. Finetuned TAPAS(Herzig et al., 2020), consisting of extra positional embeddings to encode the tabular structure was used as the second model for ensembling by obtaining classification logits from the output layer of the model.

The Label prediction on the Manual dataset was done by taking these logit scores from both the TableBERT + SciBERT and TAPAS model as:.

$$\text{LABEL} = \begin{cases} |score(TAPAS)| > 13.5 & \text{TAPAS} \\ |score(TAPAS)| < 13.5 \text{ and } |score(TableBERT)| > 0.80 & \text{TableBERT} \\ Otherwise & \text{TAPAS} \end{cases}$$

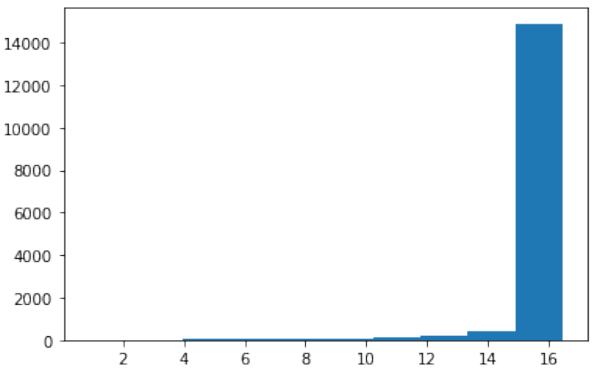

Figure 2: Absolute Score distribution of TAPAS on correctly predicted labels

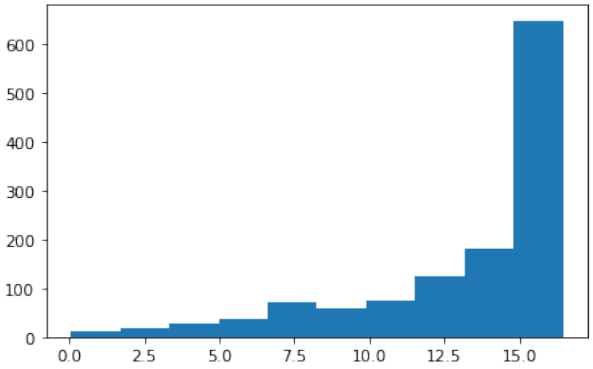

Figure 3: Absolute Score distribution of TAPAS on incorrectly predicted labels

As seen from figure 2 and 3 the score distribution on correctly predicted labels is concentrated on over a score of 14 while for incorrectly labels as the model was unsure of its prediction a significant area of the curve is below the threshold. We use this pattern for ensembling tapas with TableBERT.

The Label prediction on the Autogenerated dataset was taken by finetuning TAPAS model on both the Manual and Autogenerated Dataset. Table pruning method using Heuristic exact match (HEM) was applied for the large complex tables in the case of Autogenerated dataset.

The different model architectures were chosen keeping in mind the complexities in the datasets.For extending to 3 labels, we plan to test by two methods: A. one-hot encoding based multi-label classification B. Binary classification followed by threshold-based sorting for the UNKNOWN label.

5.2 Relevant evidence finding from table

Motivated by the previous works on Content Snapshot (Yin et al., 2020) and Table Pruning (Eisenschlos et al., 2020) , we designed an automated and scalable method of relevant cell selection. We implemented a pipeline leveraging lexical similarity and cell scoring mechanism based on aggregation prediction and iterative occlusion over each cell, as described in Figure 4.

The algorithm used for table pruning is as follow:

Row Selection : Using n-gram overlap, we select the top-K rows in the input table for each input statement. Here K, is a hyper parameter of our model.

Column Selection : Following the work of Eisenschlos et al., 2020, we use publicly available Glove (Pennington et al., 2014) model to get one embedding for each token. For a given token c the relevance with respect to the statement S is evaluated as:

$$f(S,c) = \lambda_{f(s,c)>\tau} f(s,c) \quad s \in Token(S)$$

Where τ is the threshold used to control noise and λ is the aggregation function (average, maximum, sum, etc). The final score between the statement S and Column C is computed as

$$f(S,C) = max\left[f(S,c)\right]$$

Based on the computed column score as above and a threshold, the relevant column can be extracted corresponding to a statement.

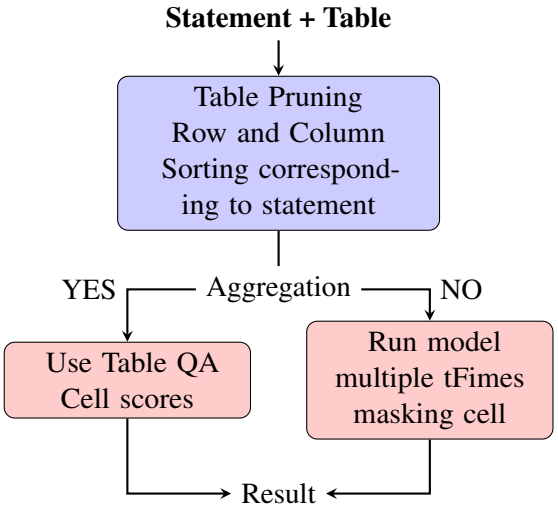

Figure 4: Pipeline for subtask 2

Though table pruning reduces the computational complexity of the task by a huge percentage but it also results in a reduces 2-3 points in accuracy particularly for the small sized tables. Therefore, on final results we used table pruning only for big table sizes.

Cell Score we iteratively mask each cell of the table and check for any substantial change in the statement's interference score(Task1). A substantial change indicated the presence of the cell in reasoning. During Experimentation, we found that this method doesn't work well for cell in the header. We tried the following approaches particularly for the header cell.

N-gram Overlap: We computed the n-gram overlap score for each cell-statement pair and cells above a threshold were marked as relevant, other irrelevant.

Token Similarity: We used the common selection algorithms stated above treating a row length to be unity and Glove Tokenizer.

Rule Based: We ran the occlusion on all cells ignoring header cells and then assigned labels to header cell as:

$$Table[0][j] = relevant \quad if Table[i][j] \in relevant$$

For this dataset, we found that the rule based gives the best result particularly due to the pattern used by the annotators but we believe that the Token similarity method gives a more generalized result.

6 Experiments and Results

Due to the late arrival of data from the organizers, all the experiments before midsems were done with the TABFACT dataset on the Baseline models: TableBERT, LPA, GNN, and TAPAS. Table 3 briefly shows the results obtained on the dataset.

Model	Val	Test
TableBERT	66.1	65.1
GNN	72.1	72.2
LPA	64.9	64.5
TAPAS	**81.0**	**72.2**

Table 3:Baseline Results drived on the TABFACT dataset for Subtask-1

Below are the baseline results and our model for Task1 on the Dev and Test dataset provided to us. The results reported in the table below doesn't include the **UNKNOWN** labels for the course (as decided by the course instructor and both the groups) since no such labels are provided in the dataset. However, we have included the **UNKNOWN** Label in the model and will be used in the future. Drastic improvement in accuracy was achieved using robust pre-processing (see Appendix) and proper fine-tuning.

Manual Annotations	Acc	F1
LPA (Finetuned)	54.64	0.51
TableBERT (Finetuned)	69.5	0.74
TableBERT + SciBERT	**73.4**	**0.74**
TableBERT + BioBERT	**60.1**	**0.70**
TAPAS	52.7	0.6
TAPAS (Finetuned)	**78.67**	**0.72**
Ensemble Model	**75.43**	**0.82**

Table 4: Results on Manual Dev Data

Autogen. Annotations	Acc	F1
LPA (Finetuned)	60.54	0.63
TableBERT (Finetuned)	83.13	0.84
TableBERT + SciBERT	87.3	0.84
TableBERT + BioBERT	**76.22**	**0.73**
TAPAS	55.9	0.58
TAPAS (Finetuned)	**94.88**	**0.93**

Table 5: Results on Autogenerated Dev data

Datasets	Dev Acc	F1
TableBERT	**80.02**	**0.79**
TAPAS	**91.6**	**0.90**

Table 6: Results on combined Dev data before merging multiple headers

Datasets	Dev Acc	F1
TableBERT	**82.71**	**0.82**
TAPAS	**92.26**	**0.92**

Table 7: Results on combined Dev data after merging multiple header

Datasets	Acc	F1
Autogenerated Dataset	**96.23**	**0.82**
Manual Dataset	**75.43**	**0.96**

Group 15

Datasets	Acc	F1
Autogenerated Dataset	87.17	0.87
Manual Dataset	70.83	0.76

Group 5

Table 8: Results derived on Test dataset

The fine-tuning for TAPAS was done on TPUv2 8GB per-core provided by Google Colab. For other experimentation, we used the standard K80 GPU present in the Colab. TAPAS outperforms all the models and decent performance by TableBERT and SciBERT on both the manual and Autogenerated Datasets. We analyzed the scope of ensembling on the models and found an increase of about 2 percent for manual annotations on the developement dataset. The same(1.94) increase was observed in the test dataset too.

For the Task 2, we did intense experiments with the different algorithm for header and Table pruning needs, which have been stated in the table and described in earlier sections(All numbers are for development set):

Method	F1
Occlusion with Table Pruning	0.59
Occlusion without Table Pruning	0.61
TAPAS logit cell-wise score(avg over tokens)	0.37
N-gram overlap(header) + occlusion	0.63
Glove based matching(header) + occlusion	0.63
Rule based(header) + occlusion	0.65

Table 9:Experiments on Validation Set

Team	F1
Group 15	**0.58**
Group 5	0.70

Table 10:Results on Test Set

7 Error Analysis

TAPAS outperforms other models primarily due to pre-training on the corpus of **synthetic** and **counterfactual statements** as described in Section 3 and Table Pruning techniques based on **heuristic entity linking (HEL)**, which was very evident for this dataset too.

On analyzing the test set results, we can see that even though the Manual dataset is reasonably simple to process and predict, it still has relatively less accuracy and F1 score than very complex Autogenerated tables. The reason being; first, the dataset has more than ten times the Annotated statement than the manual ones, leading to the model being trained on the Autogenerated files better than the manual ones. Secondly, as shown in Table 2, the manual set is positively skewed (On the test set, the ratio is 2:1), so F1 should be considered a better measure than accuracy, which favors our results on the test set too.

For Task2, we observed that our model performs very excellently well for statements that require direct aggregation or direct answer. Still, it fails for statements involving a comparison between two cells (for example, heartbeat and soap belong to the same class, Table1). Since the overall score is a mean of all the statement-table pair, this brings down our F1 score. The algorithm also gives a good recall value (more than 0.8 for both development and test set) but comparatively low precision values due to the higher number of irrelevant labels in the table, leading to higher false-positive numbers.

8 Individual Contribution

Work Done	Member
Obtained Baseline Results for TABFACT dataset on GNN Model, TableBERT, LPA	Ankur Jaya Preeti
Data Pre-Processing and LPA	Jaya Preeti
TableBERT and BioBERT	Preeti
TableBERT pretrained on Scibert	Preeti Jaya
TAPAS on the provided Dataset TAPAS on Tabfact Dataset	Jaya Ankur
Task 2	Ankur

9 Future Work

For the subtasks, our approach for the competition is as follows:

Task 1: We plan to upgrade TAPAS for the Manual Annotated dataset using **hierarchical softmax** and expand the number of aggregation functions. The aggregation parameters used in the model are not agnostic, as shown in section 7. We will also be experimenting with RoBERTa and, depending on the results, including it in the ensembling model to increase the Bottleneck efficiency.

We will also be deriving our results by including unknown label statements manually and from the new release development dataset for the competition since no such labels are provided in the training dataset.

Task 2: We saw some recent work on graph-based approaches, which can be beneficial for evidence finding. We plan to get results on them to compare with this approach.

10 Conclusion

There have been various works related to a binary classification of statements. Still, evidence finding for these classifications is a difficult novel challenge and, at the same time, interesting to work. This paper has discussed the previous works, exciting models, data analysis, and results on the baseline models using the TABFACT and our dataset. For task1, we plan to use TAPAS, TableBERT, and SciBERT Ensembled model for the Manually annotated dataset and TAPAS for the Autogenerated Dataset for task2; though it being an first time work in this area which look ahead to try out more experiments to get the best performing model.

References

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference.

Minseok Cho, Reinald Kim Amplayo, Seung won Hwang, and Jonghyuck Park. 2018. Adversarial tableqa: Attention supervision for question answering on tables.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. pages 177–190.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Julian Martin Eisenschlos, Syrine Krichene, and Thomas Müller. 2020. Understanding tables with intermediate pre-training.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, 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, Online. Association for Computational Linguistics.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480, Beijing, China. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Yibo Sun, Duyu Tang, Nan Duan, Jingjing Xu, Xiaocheng Feng, and Bing Qin. 2018. Knowledge-aware conversational semantic parsing over web tables.

Hao Wang, Xiaodong Zhang, Shuming Ma, Xu Sun, Houfeng Wang, and Mengxiang Wang. 2018. A neural question answering model based on semi-structured tables. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1941–1951, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Jianshu Chen Yunkai Zhang Hong Wang Shiyang Li Xiyou Zhou Wenhu Chen, Hongmin Wang and William Yang Wang. 2020. Tabfact : A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations (ICLR)*, Addis Ababa, Ethiopia.

Pengcheng Yin, Graham Neubig, Wen tau Yih, and Sebastian Riedel. 2020. TaBERT: Pretraining for joint understanding of textual and tabular data. In *Annual Conference of the Association for Computational Linguistics (ACL)*.

Wanjun Zhong, Duyu Tang, Zhangyin Feng, Nan Duan, Ming Zhou, Ming Gong, Linjun Shou, Daxin Jiang, Jiahai Wang, and Jian Yin. 2020. Logicalfactchecker: Leveraging logical operations for fact checking with graph module network.

Appendix

A Other Experiments

In the Tabfact Paper(Wenhu Chen and Wang, 2020), along with the LPA and TableBERT, GNN (Graph neural Network) was the third model proposed by the owner, leverage the idea proposed in NumGNN into the encoding of tabular data. This model had no paper, and it was taking a few input labels which were derived as proposed in the NumGNN model. It would use cross-Attention Between Table and Statement to obtain the Representation. But this model had few drawbacks, and the code has few problems, according to the author, so we decided to leave this model for our Task.

We also experimented with the LPA model, which excels at the symbolic reasoning aspects by executing database queries and counterpart to the TableBERT model, working on the Linguistic Reasoning and semantic level understanding. Unfortunately, the model's vocab file deciding the fine-tuned model's embedding was fixed and based on the Tabfact dataset. We had to interpolate contents in our vocab file to match their model's embedding for Finetuning the LPA model on our dataset. This method was advised to us by the author himself, but the accuracy we report after finetuning on the model is less than other models. Hence, we had to leave this model for our Task as well.

We also experimented with integrating SciBERT in TAPAS. It has BERT's architecture (Devlin et al., 2019) extended with additional embeddings that capture the tabular structure and two classification layers for selecting cells and predicting a corresponding aggregation operator. Since it is an extended BERT model, it was not possible in the limited time to change the entire architecture of the model from scratch. It is very computationally expensive to train the TAPAS model as well. Hence, we had integrated Scibert in the TableBERT model, which uses standard BERT to improve the model.

B Preprocessing of the Multiple Header files

Our dataset had many Tables with multiple headers and subcolumns, as shown below.

ExperMatter	UserB		UserC	
	Base1	Base2	Base1	Base2
Gold	5.6		7	8
Silver	3.4	6.7	8.0	6.7

Since most of our models take single header as input only and with equal number of columns in every row, we had to convert such tables to suit our input type.There were two processes involved for preprocessing such Tables.

Intrapolation: For one table we would calculate the maximum numbers of columns in the Table and then intrapolate all other rows with less columns to eventually have equal number of columns in every row.The below table is the end result obtained by preprocessing the above table by intrapolation.

ExperMatter	UserB	UserB	UserC	UserC
	Base1	Base2	Base1	Base2
Gold	5.6		7	8
Silver	3.4	6.7	8.0	6.7

Header Merging: Since the input our model has to be a single header file we had to merge such rows as shown below.The below table is the final table obtained after Preprocessing.

ExperMatter	UserB Base1	UserB Base2	UserC Base1	UserC Base2
Gold	5.6		7	8
Silver	3.4	6.7	8.0	6.7

C Training

Finetuning the TAPAS model on the Manual and Autogenerated dataset was quite computationally expensive. Therefore we had to Train TAPAS in eight batches of 200 files for Autogenerated Data and two batches of 400 files for the Manual Dataset.TAPAS model used in predicting Manually annotated Dataset was finetuned on 18 epochs of each batch of Manual Dataset and two epochs of each batch of half Autogenerated Files. The TAPAS model used in predicting Autogenerated Dataset was finetuned on 18 epochs of each batch of Manual Dataset and three epochs of each batch of all the Autogenerated Files. Generating different checkpoints and preprocessing the batches took about eight to nine days using the Colab's TPU. For task2, since we must compute inference predictions for all the cells in the table, we used a GPU instance by AWS (using AWS student credit) for the same. We used the trained task 1 model for making the prediction, and for getting the prediction, the compute time was nearly 30 hours for the validation dataset.

We are deeply grateful to Prof. Ashutosh Modi and all the Teaching Assistants for mentoring us in the project. We also thank all students for their time, constructive feedback, useful comments, and suggestions about the Work.