Project 5: Statement Verification and Evidence Finding with Tables

Aditya Jindal[1], **Vishesh Kaushik**[2], **Vijit Malik**[3]
[1]170048, [2]170805, [3]170791
[1]ECO, [2]ME, [3]ME
{adityaji, kvishesh ,vijitvm}@iitk.ac.in

Abstract

Recently, there has been an increase in research on factual verification or prediction over structured data like tables and graphs. To circumvent any false news incident, it is necessary to not only model and predict over structured data efficiently but also to explain those predictions. In this paper, we tackle the problem of fact verification and evidence finding over tabular data. We make a comparison of the baselines and state of the art approaches over the given SemEval dataset. We also propose a novel approach CellBERT to solve the task of evidence finding, as a form of Natural Language Inference task.

1 Introduction

Verification of a textual fact, whether entailed or refuted by the given evidence, is a fundamental problem in natural language understanding. Due to the unprecedented amount of false news and rumors spreading through the internet, Fact-checking has recently become an essential research topic (Goodrich et al., 2019), (Nakamura et al., 2019), (Vaibhav et al., 2019), (Kryściński et al., 2019). Moreover, online misinformation may manipulate people's opinions and significantly influence critical social events like political elections. Tables are ubiquitous in documents and presentations for concisely conveying important information. Fact-checking on structured data like tables or graphs is much more difficult compared to simple text format. This task provides another critical perspective in finding out which cells of the table are relevant for verifying the given fact.

Describing all the information provided in this table in a readable manner would be lengthy and considerably more difficult to understand. The task is related to whether a textual hypothesis holds based on the given information in the table. There has been a lot of work done on unstructured data as an NLI task but verification under structured evidence, such as tables, graphs, and databases, has not been explored much. Misunderstanding of tabular data can lead to fake news which we can see all too prevalent today. Two reasons are important while doing fact-checking from table (Chen et al., 2019). The first one is linguistic reasoning which includes the semantic level of understanding of the text in the table. The second type of reasoning is symbolic reasoning in which we need a symbolical understanding of the structure of the table and doing both symbolic and linguistic reasoning simultaneously is a difficult task.

In this paper, we tackle the problem of Statement Verification and Evidence finding over tables. We also propose a novel idea for the subtask of Evidence finding from the tables by approaching it as an individual cell-based NLI task. For the Subtask A We also provide a comparison of baseline results of TableBERT and other Table transformers with the current state-of-the-art TAPAS model. We have made all our code available on github[1].

2 Problem Defintion

The problem statement is articulated around two related subtasks. The first subtask consists of Table and Statement Support in which given a table and statement we have to determine whether the statement is supported by the table. In the second subtask Relevant Cell Selection we need to found which table cells form relevant evidence for the statement. Mathematically, the problem can be described as, given a table $\mathbb{T}$ that consists of cells x_{ij} corresponding to row r_i and column c_j, and a fact $\mathbb{F}$. We need to perform two subtasks as follows:

[1] https://github.com/vijit-m/TablEval

2.1 Subtask A - Table Statement Support

Given a statement/fact, some of which will be directly adapted from the linking text, and a table, determine whether the statement is supported by the table. In this classification problem, a statement is assigned one of the following labels:

- **Fully Supported:** Statement is supported by data found within the table. We denote the label by 1.
- **Refuted:** Statement is contradicted by table. Label denoted by -1.
- **Unknown:** Not enough information in table to assess statement veracity. Denoted by 0.

This can be denoted as a classification task, in which given $\{\mathbb{T}, \mathbb{F}\}$, we need to perform the mapping to the output y (corresponding to $\mathbb{F}$), where $y \in \{-1, 0, 1\}$.

2.2 Subtask B - Relevant Cell Selection

Given a statement and a table, find which table cells form relevant evidence for the statement (if any). A table cell is evidence for a statement if it helps support or refute a part of the statement:

- **Relevant:** the cell must be included. Denoted by 1.
- **Ambiguous:** the cell is allowed to be either included or not included. Denoted by 0.
- **Irrelevant:** the cell must not be included. Denoted by -1.

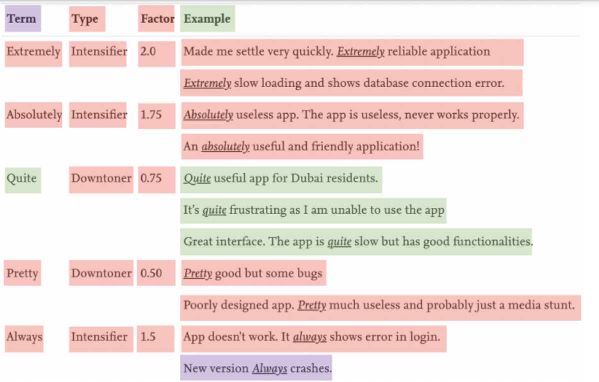
Mathematically, each cell $x_{ij} \in \mathbb{T}$, each x_{ij} needs to be mapped to a value $y \in \{-1, 0, 1\}$ for the corresponding $\mathbb{F}$.

These categories allow us to differentiate between cells that can be objectively categorized as relevant or irrelevant, and cells that, while not necessarily required to come to a conclusion about the statement, are also not entirely unrelated.

We have also provided a sample output for our task in Figure[2] 1.

2.3 Evaluation Measures

The evaluation of the results considers different strategies and metrics for Subtasks A and B in order to allow more fine-grained scores.



Term	Type	Factor	Example
Extremely	Intensifier	2.0	Made me settle very quickly. *Extremely* reliable application
			Extremely slow loading and shows database connection error.
Absolutely	Intensifier	1.75	*Absolutely* useless app. The app is useless, never works properly.
			An *absolutely* useful and friendly application!
Quite	Downtoner	0.75	*Quite* useful app for Dubai residents.
			It’s *quite* frustrating as I am unable to use the app
			Great interface. The app is *quite* slow but has good functionalities.
Pretty	Downtoner	0.50	*Pretty* good but some bugs
			Poorly designed app. *Pretty* much useless and probably just a media stunt.
Always	Intensifier	1.5	App doesn't work. It *always* shows error in login.
			New version *Always* crashes.

(a) Example Subtask B output for the statement in bold in Figure 1(b)

Statement	Label
The polarity score of the opinion word that follows the downtoner “quite” is multiplied by the factor (0.75)	Supported
“New version Always crashes” is an example for “Quite”	Refuted
The “Extremely” term has the highest factor.	Supported
The polarity score of the opinion word that follows the intensifier “very” is multiplied by the factor (1.25)	Unknown

(b) Example Subtask A output

Figure 1: Example Table and output for Subtask A & B

Subtask A The test set will be balanced between the three categories and the final ranking will be based on accuracy averaged across all three categories. There will be two evaluation methods. The first will be a standard precision/recall evaluation of a multi-class classification that evaluates whether each table was classified correctly as True / False / Unknown. This will test whether the classification algorithm understands cases where there is insufficient information to make a determination. The second, simpler evaluation will not penalize misclassifying an Unknown table as True or False, while still treating a True/False misclassification as an error for calculating precision and recall.

Subtask B The evaluation will calculate aggregated precision and recall for a statement-table pair, weighting each cell equally, with ambiguous cells not influencing the results.

3 Related Work

Most of the previous studies in fact-checking mainly focused on making better use of the meaning of words or linguistic reasoning while rarely considered symbolic reasoning about logical op-

[2] img src: https://sites.google.com/view/sem-tab-facts/semeval-proposal

erations. Modeling logical operations (like count, greater, etc.) is an essential step towards the modeling of complex reasoning and semantic compositionality, which is also known as symbolic reasoning. (Chen et al., 2019) proposed the TabFact dataset along with two baseline approaches. The TabFact dataset is very similar to the task's Subtask A but with only two labels, Entailed and Refuted. They proposed TableBERT, in which they linearize the given table along with the fact and perform the fact-checking as a simple NLI task using BERT (Devlin et al., 2018). They also proposed LPA (Latent Program Algorithm) to formulate the table fact-checking as a program synthesis problem by using reinforcement learning to directly optimize the task reward of this structured. prediction problem, as was done in Neural-Symbolic machines introduced in (Liang et al., 2016). They also used A collection of these two types of reasonings integrated with an understanding of a given table's structural format that was performed in (Zhong et al., 2020). prediction problem, as was done in Neural-Symbolic machines introduced in (Liang et al., 2016).They also used A collection of these two types of reasonings integrated with an understanding of a given table's structural format was performed in (Zhong et al., 2020). Recently, the TAPAS model (Herzig et al., 2020) was proposed by Google for the task of weakly supervised question answering over tabular data. They proposed a new pre-training scheme different than BERT to make symbolic reasoning possible to some extent. The Tapas model has also been tweaked for fact-checking tasks as well and is currently state-of-the-art in the task of fact-checking on tables. On some similar lines, TABERT, a pre-trained LM that jointly learns representations for NL sentences and (semi-)structured tables were proposed in (Yin et al., 2020).

Our task has dissimilarities with the above mentioned related works. The above work mainly focused on subtask A and no prior work has been performed on subtask B yet. Both the subtasks have three labels but prior work has been done on just two labels which is also a point to ponder.

4 Corpus/Data Description

Following the large-scale dataset called TabFact[3] with 16k Wikipedia as evidence for 118k human-annotated statements to study fact verification tables, SemEval produces its own dataset with around 1k tables and relatively more number of statements per table for verification. Further Semeval has captions of table unlike TabFact, which helps us in providing background knowledge for table data.

4.1 Corpus Collection

The training and testing data will be sourced from open-access scientific articles with tables using APIs provided by Science Direct for data mining. The format that the data will be procured is in XML so that the tables will be structured. With XML files we also have the table in image format since the size and styling of table contents can be useful in table understanding. The statements will be from a combination of automatic-generated and manual sources. The text will be from the same scientific articles. We have 957 tables for training, 100 tables for testing. and at least 20 statements per each table. For Subtask B, we have 200 of the 1000 tables in the training set and all of the test set with relevance ground truth.

4.2 Annotation Process

Each statement in SemEval will be adapted from existing text and verified by at least one human reader. A smaller proportion will be verified by multiple readers to assess inter-rater agreement. To perform the annotation each table requires around 15 minutes to produce 10 statements. With a goal of 1100 tables, it will take approximately 280 hours to generate the entire set for Subtask A. With Subtask B, it takes an extra 1 minute per statement. In this process, some annotations are Manual as they are sourced from human annotators with 957 tables and 3764 statements. Besides these, some statements are auto-generated using a random paraphraser, and table understanding parser that includes 174948 statements.

5 Data Analysis

Since the dataset consists of both manual annotations and auto-generated evaluations, we analyzed them separately. See table 1. For a more visual representation for dataset statistics see fig. 3 and fig. 2.

Data Changelog given by the organisers:
V1. Initial release
V1.1. Small expansion of training data, correct xml structure to have table id be an attribute and all tables inside a document tag.

[3] https://tabfact.github.io/

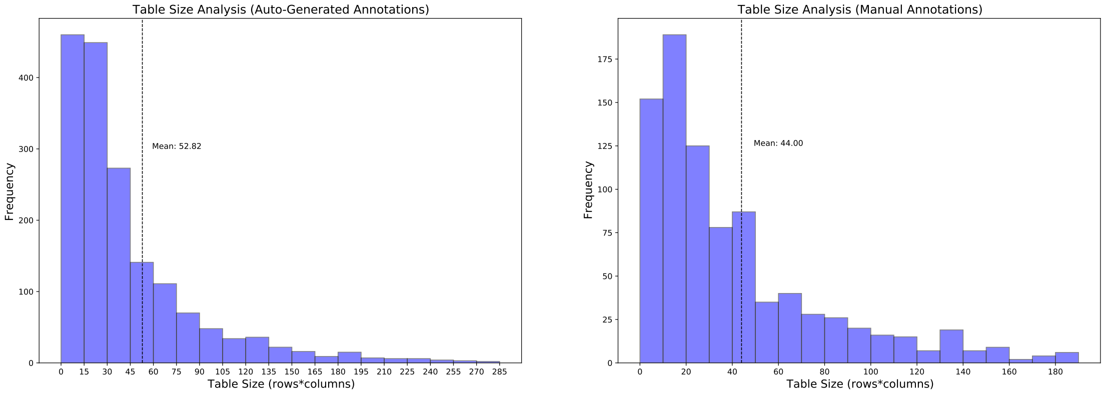


Figure 2: Table Size statistics of manual vs auto generated statements dataset.

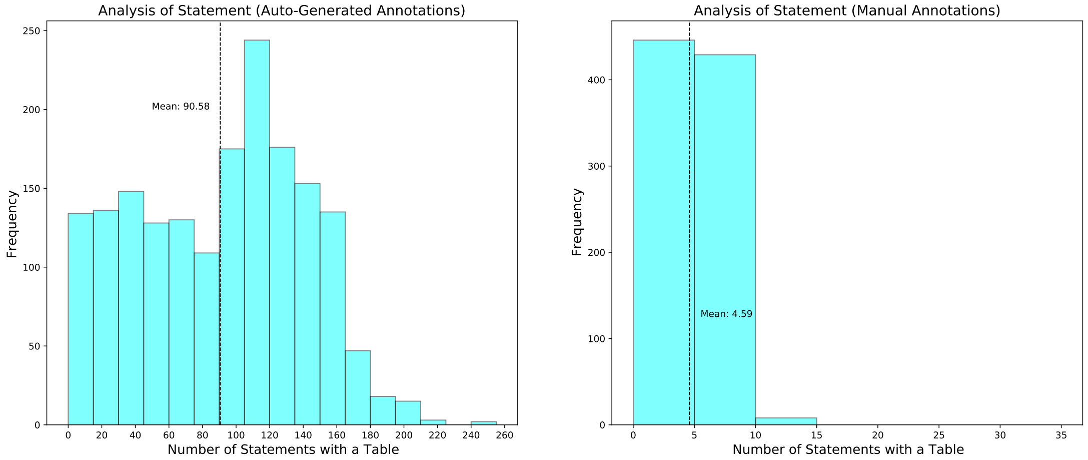


Figure 3: Number of statements for a table statistics in manual vs auto generated data.

V1.2. Added legend tag to xml file in training data.

6 Proposed Approach

In this paper, for Subtask A we experiment with using TAPAS and TableBERT mutations on the given SemEval dataset, while also encoding the surrounding information. For subtask A, we have fine tuned TableBERT on the given train set. Similar to TableBERT we have implemented other table transformers like TableRoBERTa (Liu et al., 2019) and TableSciBERT (Beltagy et al., 2019). We also experimented with implementing BiGRU layers on top of TableRoBERTa. For subtask B we have done it as an individual cell tweaked Natural Language Inference task, where the premise is taken as the combination of row header, column header and cell contents and the hypothesis is taken as the statement provided. Please note that not all tables have row header or in some cases column headers as well. For such cases the row header gives us the cell in the first column in the same row and for the column header we get the cell in the same column in the first row.

7 Experiments and Results

7.1 Subtask A

Following the TabFact's TableBERT, we fine-tuned our own TableBERT model on the given SemEval dataset for Subtask A. We also experimented with mutations of TableBERT by using RoBERTa

Dataset	Number of tables			Mean no.	Mean no.	Mean no.	Mean no,.
Type	Train	Dev	Test	of rows	of columns	of cells	of statements
Auto	1591	195	194	10.23	4.79	52.82	90.58
Manual	783	100	98	9.23	4.65	44	4.59

Table 1: Data statistics

Model	Train	Dev	Metrics (On Dev set)			
	set	set	Precision	Recall	F1	Accuracy (%)
TableBERT	**Auto**	**Auto**	**0.875**	**0.859**	**0.867**	**86.00**
TableRoBERTa	Auto	Auto	0.635	0.631	0.633	64.05
TableRoBERTa + BiGRU	Auto	Auto	0.647	0.634	0.660	67.13
TableSciBERT	Auto	Auto	0.710	0.698	0.704	70.74
TAPAS	TabFact	Auto	0.667	0.620	0.643	74.76
TAPAS	**TabFact**	**Manual**	**0.778**	**0.761**	**0.771**	**72.26**
TableBERT	Auto	Manual	0.588	0.582	0.585	58.95
TableRoBERTa	Auto	Manual	0.529	0.507	0.518	51.95
TableRoBERTa + BiGRU	Manual	Manual	0.542	0.520	0.531	53.28
TableSciBERT	Manual	Manual	0.614	0.606	0.610	61.07

Table 2: Our Results for Subtask A

(TableRoBERTa) as well. Since the dataset consisted of various scientific statements, we figured that using SciBERT might be relevant. Therefore, we implemented the TableSciBERT, with the same motivation as TableBERT and fine-tuned it on our SemEval data as well. We also experimented with implementing BiGRU layers on top of TableRoBERTa as well. Please note that all the table transformers were fine-tuned on the Auto-generated dataset of SemEval data. This is because we were unable to fine-tune them on manual data due to its small size. For the manual dataset we also evaluated results using the pretrained TAPAS model fine-tuned on TabFact dataset. See table 2 for results of our experiments on the development set. The transformers trained on auto-generated dataset were used to evaluate for the manual dataset as well. We can see that the best performance was given by TableBERT on the auto-generated Development dataset, therefore, for the evaluation on Test set, we proceeded with using TableBERT for the Auto-generated dataset and TAPAS for the Manual dataset. Using TableBERT we got an F1 score of 0.8717 and TAPAS performs best on the Manual dataset with an F1 score of 0.7671. For comparison of our results with Group 15, see table 3. In comparison to their results we have underperformed by 0.09 F1 on auto-generated dataset and by 0.04 F1 on manual dataset.

7.2 Subtask B

For Subtask B, we used the approach as in fig. 4 and implemented a BERT base model. We call our approach CellBERT since we perform NLI task on each cell. As the number of statements given to us in the auto-generated dataset is large and also accounting for the fact that each cell in the table is a separate input example, the number of tuples of (cell, statement) were very large. Due to computational limitations, we only trained CellBERT on 0.1% of such tuples and discarded the rest of the data. See table 4 for results on the auto-generated dataset. Also, we have outperformed the results by Group 15 in subtask B by an F1 score of 0.12.

Model	Train Set (Auto)	Dev Set (Auto)
	F1	F1
CellBERT	0.7772	0.6783

Table 5: Results for Subtask B on Train and Dev Set

[CLS] + [Row_Header] + [Cell Content] + [Column Header] + **[SEP]** + [Statement Text] + **[SEP]**

Figure 4: Approach for Subtask B

Model	Test Dataset	Precision	Recall	F1	Accuracy (%)
TableBERT	Auto	0.8717	0.8718	0.8717	87.17
Group 15	**Auto**	**0.9696**	**0.9559**	**0.9627**	**96.23**
TAPAS	Manual	0.8264	0.7157	0.7671	70.83
Group 15	**Manual**	**0.7975**	**0.8497**	**0.8227**	**75.44**

Table 3: Overall Results for Subtask A on Test Set

Model	Train set	Test set	F1
CellBERT	**Auto**	**Auto**	**0.7047**
Group 15	Auto	Auto	0.5789

Table 4: Overall Results for Subtask B

8 Error Analysis

We noticed that TableSciBERT was underperforming compared to TableBERT on Subtask A even when the dataset had a lot of scientific statements. TAPAS was able to perform the best on the Manual dataset. Note that the F1 score of TAPAS is greater than the accuracy for Subtask A suggesting that we've got a good amount of sensitivity for a class. It is possible that our model is over predicting a class. Unfortunately due to time constraints we were unable to fine-tune TAPAS on Subtask A SemEval data, which we will do as a future work. We are confident that this would provide us with better results as it did in case of Group 15. Also, due to unavailability of computational resources we were unable to train LPA as well.

For Subtask B we have used a considerably less amount of data to train CellBERT and even that provides us with promising results. Also, due to time constraints were unable to experiment with using other models or using larger versions of the transformers.

9 Conclusion

This paper tries to look for a solution to an underexplored but important problem: Statement Verification and Evidence Finding with Tables. We verified the existing models like TableBERT and TAPAS. Also, We implemented TableSciBERT and TableRoBERTa by putting Bi-GRU layers on top of it. In the future, we plan to progress in the direction of implementing new models that can tackle both linguistic and symbolic reasoning. We are planning to fine tune TAPAS on the manual and auto generated dataset and to use an ensemble of these models to achieve a boost in accuracy for subtask A. In case of subtask B we are planning to experiment on other NLI techniques and models. In addition we will be looking into using more data for training CellBERT.

References

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2019. Tabfact: A large-scale dataset for table-based fact verification. *arXiv preprint arXiv:1909.02164*.

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*.

Ben Goodrich, Vinay Rao, Peter J Liu, and Mohammad Saleh. 2019. Assessing the factual accuracy of generated text. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 166–175.

Jonathan Herzig, Paweł Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Martin Eisenschlos. 2020. Tapas: Weakly supervised table parsing via pre-training. *arXiv preprint arXiv:2004.02349*.

Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Evaluating the factual consistency of abstractive text summarization. *arXiv preprint arXiv:1910.12840*.

Chen Liang, Jonathan Berant, Quoc Le, Kenneth D Forbus, and Ni Lao. 2016. Neural symbolic machines: Learning semantic parsers on freebase with weak supervision. *arXiv preprint arXiv:1611.00020*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Kai Nakamura, Sharon Levy, and William Yang Wang. 2019. r/fakeddit: A new multimodal benchmark dataset for fine-grained fake news detection. *arXiv preprint arXiv:1911.03854*.

Vaibhav Vaibhav, Raghuram Mandyam Annasamy, and Eduard Hovy. 2019. Do sentence interactions matter? leveraging sentence level representations for fake news classification. *arXiv preprint arXiv:1910.12203*.

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*.

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. *arXiv preprint arXiv:2004.13659*.