Volta at SemEval-2021 Task 9: Statement Verification and Evidence Finding with Tables using TAPAS and Transfer Learning

Devansh Gautam[*]   **Kshitij Gupta**[*]   **Manish Shrivastava**

International Institute of Information Technology Hyderabad

{devansh.gautam,kshitij.gupta}@research.iiit.ac.in,
m.shrivastava@iiit.ac.in

Abstract

Tables are widely used in various kinds of documents to present information concisely. Understanding tables is a challenging problem that requires an understanding of language and table structure, along with numerical and logical reasoning. In this paper, we present our systems to solve Task 9 of SemEval-2021: *Statement Verification and Evidence Finding with Tables* (SEM-TAB-FACTS). The task consists of two subtasks: (A) Given a table and a statement, predicting whether the table supports the statement and (B) Predicting which cells in the table provide evidence for/against the statement. We fine-tune TAPAS (a model which extends BERT's architecture to capture tabular structure) for both the subtasks as it has shown state-of-the-art performance in various table understanding tasks. In subtask A, we evaluate how transfer learning and standardizing tables to have a single header row improves TAPAS' performance. In subtask B, we evaluate how different fine-tuning strategies can improve TAPAS' performance. Our systems achieve an F1 score of 67.34 in subtask A three-way classification, 72.89 in subtask A two-way classification, and 62.95 in subtask B.

1 Introduction

There has been extensive work on verifying if a given textual context supports a given statement. Even though tables are also widely used to convey information, especially in scientific texts, there has been comparatively less work on verifying if a given table supports a statement. To this end, SemEval 2021 Task 9 (Wang et al., 2021) focuses on statement verification and evidence finding for tables from scientific articles in the English language. The task is divided into two subtasks - A and B. The aim of subtask A is to classify whether a given

statement is entailed or refuted according to the given table and associated table metadata (such as captions and legends) or whether the statement's truth is unknown as it cannot be determined from the table. The aim of subtask B is to classify each cell in the table as relevant or irrelevant in determining whether the statement is entailed or refuted from the tabular evidence (the truth value of the statement is also provided).

Our systems use TAPAS (Herzig et al., 2020) trained with intermediate pre-training (Eisenschlos et al., 2020) for both the subtasks. For subtask A, we fine-tune TAPAS after adding a three-way classification head on top for classifying the statement as entailed/refuted/unknown. We also evaluate how transfer learning and standardizing tables to have a single header row can improve TAPAS' performance. Due to the similarity between subtask B and table question-answering (which involves cell selection or cell selection followed by aggregation), we use the TAPAS architecture previously used for table question-answering and fine-tune it to select the relevant cells. We also evaluate how different fine-tuning strategies can improve TAPAS' performance on evidence finding.

Our systems achieve an F1-micro score of 67.34 in subtask A and 72.89 in subtask A if the unknown statements are not considered while calculating the metrics (however, classifying entailed/refuted statements as unknown is still penalized). Our submitted system achieves an F1 score of 62.95 in subtask B. During the post-evaluation phase, we modified our system and achieved an F1-score of 65.48 in subtask B.

The code for our systems is available at https://github.com/devanshg27/sem-tab-fact.

---

[*]The authors have contributed equally.

	(1)	(2)	(3)	(4)
	English Language		English	
	Frequency	Percent	Frequency	Percent
AQA	241539	61.6	84742	55.7
WJEC	83219	21.2	39650	26.1
Pearson	37194	9.5	18815	12.4
OCR	30061	7.7	8818	5.8
Total	392015		152025	

Caption: Number of GCSE Full Course entries by Awarding Body (KS4 Results tables, 2014)

Legend: Note. Number of GCSE Full Course entries in the summer season of the academic year 2012-2013. AQA (The Assessment and Qualifications Alliance); WJEC (Welsh Joint Education Committee); OCR (Oxford, Cambridge and RSA Examinations); CCEA (Council for the Curriculum, Examinations and Assessment). We do not show the information of an additional awarding body that accounts for almost no entries.

Entailed:	1. The highest Frequency, not counting the Total, is 84742.
	2. The highest English Percent is for AQA
Refuted:	1. The highest Percent value for OCR is 5.8
	2. The lowest total is 392015
Unknown:	1. First, this is due to technical problems in providing Unique Candidate Numbers (UPN) for all candidates.
	2. This is for four main reasons.

Figure 1: An example from the SEM-TAB-FACTS dataset: Table `A1` From `10262.xml` along with its caption and legend. Some example statements of each class associated with this table are also shown. The highlighted cells are the relevant cells for entailed statement 2.

2 Background

Verifying if the given textual evidence supports a given statement is a fundamental natural language processing problem. It has been extensively studied under different tasks such as RTE (Recognizing Textual Entailment) (Dagan et al., 2006), NLI (Natural Language Inference) (Bowman et al., 2015), FEVER (Fact Extraction and VERification) (Thorne et al., 2018). In recent years, large-scale pre-trained models (Devlin et al., 2019; Peters et al., 2018; Yang et al., 2019; Liu et al., 2019) have dominated these tasks and have achieved close-to-human performance. NLVR (Suhr et al., 2017) and NLVR2 (Suhr et al., 2019) focus on verifying a statement given an image as evidence. TAB-FACT (Chen et al., 2020) focuses on verifying a statement given a table from Wikipedia[1] as evidence.

Along with releasing TABFACT, Chen et al. (2020) also discuss two promising approaches for tabular fact verification, Latent Program Algorithm(LPA) and Table-BERT. LPA is a semantic parsing approach that parses statements into programs (logical forms) and executes the programs against the table to predict the entailment decision. Most of the current models (Zhong et al., 2020; Shi et al., 2020; Yang et al., 2020) for TABFACT are semantic parsing approaches similar to LPA. Table-BERT encodes the linearized tables and statements using BERT-based models and directly pre-

dicts the entailment decision. Zhang et al. (2020) inject table structural information into the mask of the self-attention layer of BERT-based models, which helps the model learn better table representations. TAPAS (Herzig et al., 2020) extends BERT's architecture to capture the tabular structure, and it showed competitive performance on various table question answering datasets: SQA (Iyyer et al., 2017), WTQ (Pasupat and Liang, 2015) and WikiSQL (Zhong et al., 2017). Eisenschlos et al. (2020) add an intermediate pre-training step before the fine-tuning step to TAPAS and show that it achieves state-of-the-art results on TABFACT and SQA (Iyyer et al., 2017). Their model is still 8 points behind human performance on TABFACT since tabular fact verification involves table understanding and complex reasoning.

While TABFACT also focuses on fact verification using tables as evidence, it focuses on tables from Wikipedia, whereas SemEval-2021 Task 9 (SEM-TAB-FACTS) instead focuses on tables from scientific articles and has a subtask related to evidence finding. Also, TABFACT did not have a neutral/unknown class, which they left out because of low inter-worker agreement due to confusion with refuted class. Figure 1 shows an example of a table from the SEM-TAB-FACTS dataset and the labels for the two subtasks.

---

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

	Train (Auto) Set	Train (Manual) Set	Dev Set
Total number of tables with `<thead>` tag	1977	980	52
Number of tables with correct header prediction	1855(93.83%)	918(93.67%)	51(98.08%)
Number of tables with header prediction error is ≤ 1	1966(99.44%)	972(99.18%)	52(100%)

Table 1: Header Prediction Statistics

	(1) English Language Frequency	(2) English Language Percent	(3) English Frequency	(4) English Percent
AQA	241539	61.6	84742	55.7
WJEC	83219	21.2	39650	26.1
Pearson	37194	9.5	18815	12.4
OCR	30061	7.7	8818	5.8
Total	392015		152025	

(a) Converting multi-row/multi-column cells to single cells

	(1) English Language Frequency	(2) English Language Percent	(3) English Frequency	(4) English Percent
AQA	241539	61.6	84742	55.7
WJEC	83219	21.2	39650	26.1
Pearson	37194	9.5	18815	12.4
OCR	30061	7.7	8818	5.8
Total	392015		152025	

(b) Standardizing the header rows of the table with single cells

Figure 2: Pre-processing and header standardization applied to the table shown in Figure 1.

3 System Overview

In this section, we provide a general overview of our systems for the two subtasks. We use TAPAS for both subtasks.

3.1 Subtask A: Statement Verification

Pre-processing Since TAPAS only works on tables with single cells (cells which do not span multiple columns/rows) only, we first convert the tables with multi-row/multi-column cells to tables with only single cells by duplicating the value of the cell in every single cell the multi-row/multi-column cell spans. An example of the pre-processing is shown in Figure 2a.

Header Standardization We experiment with standardizing the pre-processed tables with multi-row headers to tables with a single header row since TAPAS was pre-trained on single header tables and TABFACT (which we want to use for transfer learning) also contains single header tables. We first predict the number of header rows using the following rules:

1. In many pre-processed tables, we found that the left-most column contained row names, and either (a) all the header cells in the left-most column were empty, or (b) the cell value at the top-left corner was repeated in all the header cells below it, or (c) the cell at the top-left corner was not empty, but the header cells below it were empty. Based on these cases, we initially estimate the number of header rows as the number of rows at the top, such that all cells in the left-most column in those rows are either empty or have the same value as the cell at the top-left corner.

2. We also found that in many cases, there were multi-column cells in the header, which had more specific sub-headers in the rows below. To handle these cases, we increment the estimate of header rows until no two adjacent columns have the same header cell values.

We merge the predicted header rows into a single row by joining each column's header cell values into a single cell with a newline as a separator. An example of header standardization is shown in Figure 2b. We were provided with HTML versions of the tables in the training and development set. We compare our predictions against the `<thead>` tags in the HTML tables to analyze our header prediction system's performance. The results are shown in Table 1. We also find that in almost all of the cases, the predictions are either correct or have an error of ± 1.

To study the effect of header standardization, we will train all our systems with and without header standardization.

Model Our model takes the following input: `[CLS] <statement> [SEP] <flattened table>`, which is tokenized using the standard BERT tokenizer. We compute the class probabilities using a linear layer with a softmax activation function on top of the output of the `[CLS]` token, as shown in Figure 3a. We use the weighted cross-entropy loss, which helps in handling imbalance in the class sizes:
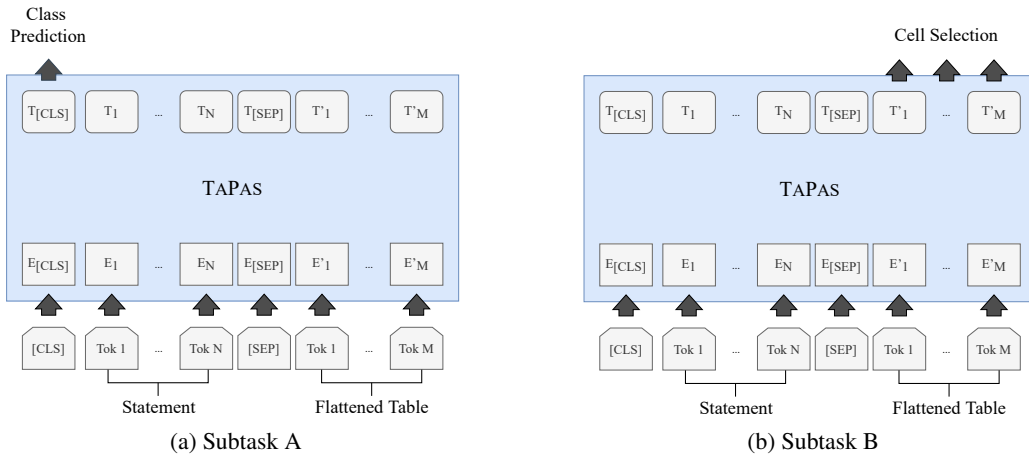
Figure 3: The architecture of our models

$$H_y(y') = -\sum_i \sum_{k=1}^{K} w_k y_{ik} \cdot \log(y'_{ik})$$

Where y_{ik} denotes the ground truth label, it is 1 if k is the true class label of the ith token, and 0 otherwise, y'_{ik} is the corresponding model probability prediction and w_k is the weight for class k. We set w_k as the size of the biggest class divided by the size of class k.

To analyze how transfer learning can improve performance, we compare the following approaches:

- **TAPAS-stf:** We use the publicly available TAPAS checkpoint which has been pre-trained with a masked language modeling objective and fine-tune it on the SEM-TAB-FACTS dataset provided by the task organizers.

- **TAPAS-tf:** As a baseline, we directly use the publicly available TAPAS checkpoint, which had been fine-tuned on TABFACT without any further fine-tuning on SEM-TAB-FACTS. Since TABFACT has only entailed/refuted labels, this model is a binary classifier and does not predict the unknown class's probabilities.

- **TAPAS-tf-stf:** We use the publicly available TAPAS checkpoint, which had been fine-tuned on TABFACT and further fine-tune it on the SEM-TAB-FACTS dataset released by the task organizers. This is our submitted model for subtask A.

3.2 Subtask B: Evidence Finding

Pre-processing and Header Standardization
We convert the multi-row/multi-column cells and standardize the header rows as discussed in Section 3.1. The relevant/irrelevant labels of the multi-row/multi-column cells are duplicated to all the single cells they span. We consider the relevant/irrelevant labels only for the cells of the non-header rows as TAPAS does not make predictions for header cells. Based on the performance of header standardization in subtask A (which we will discuss in Section 5), we standardize headers for all our models in this subtask.

Model Our model takes the following input: `[CLS] <statement> [SEP] <flattened table>`, which is tokenized using the standard BERT tokenizer. We show the architecture of our model in Figure 3b. Our model computes token-level logits using a linear layer on top of each token's last hidden state output, which are used to compute cell-level logits by averaging the logits of the tokens in each cell. The probability of selection for each cell is calculated from the cell-level logits using the sigmoid function. We use the weighted binary cross-entropy loss which helps in handling class imbalance:

$$H_y(y') = -\sum_i w_p y_i \cdot \log y'_i + (1 - y_i) \cdot \log\left(1 - y'_i\right)$$

Where y_i denotes the ground-truth label, it is 1 if the ith token is part of any relevant cell, and 0 otherwise, y'_i is the corresponding model probability prediction, and w_p denotes the weight of the positive (relevant) class. We set w_p to 10.

1265

	#Tables	#Entailed statements	#Refuted statements	#Unknown statements
Train (Auto-generated)	1980	92136	87209	0
Train (Manually annotated)	981	2818	1688	0
Train (with unknown statements)	981	2818	1688	4506
Validation	52	250	213	93
Test	52	274	248	131

(a) Subtask A

	#Tables	#Entailed statements	#Refuted statements	#Relevant cells	#Irrelevant cells
Train (auto-generated)	1980	92136	87209	1039058	15467957
Validation	51	233	191	3048	28495
Test	52	251	219	3458	26724

(b) Subtask B

Table 2: Dataset Statistics for each subtask

Due to the similarity of evidence finding with table question-answering, we use the publicly available TAPAS checkpoint, which was fine-tuned in a chain on SQA, WikiSQL, and finally WTQ. We compare the following fine-tuning strategies:

- **WTQ-base:** As a baseline, we fine-tune our model directly for relevant cell selection on SEM-TAB-FACTS.

- **WTQ-statement:** We again fine-tune the model for relevant cell selection on SEM-TAB-FACTS, but we try to include the information on whether the statement was entailed/refuted by modelling the statement as '*Which cells entail "<statement>"?*' or '*Which cells refute "<statement>"?*'. <statement> denotes the original statement.

- **WTQ-separate:** We fine-tune two separate models, one which predicts the relevant cells for entailed statements and another one for refuted statements. This is our submitted system for subtask B.

During the post-evaluation phase, we experimented with the publicly available TAPAS checkpoint, which was fine-tuned on TABFACT. Similar to the systems described above, we compare three systems based on this checkpoint: TABFACT-base, TABFACT-statement, and TABFACT-separate.

Post-Processing We further apply post-processing steps to obtain the final prediction from the cell classification. To predict the header's relevant cells, we select the header cells for any column with cells selected as a relevant cell. We label multi-row/multi-column cells as relevant if

any of the single cells they span are predicted as relevant.

4 Experimental Setup

4.1 Data Description

We used the dataset provided by the task organizers for both subtasks. We did not use the table metadata in our systems.

For subtask A, dataset statistics and the official splits are shown in Table 2a. The provided training sets do not have any statements of the unknown class. So, we used the manually annotated training set to create a training set with unknown statements. Each statement of the manually annotated training set was added as an unknown statement to a different table chosen randomly. We used this dataset for training all our models for subtask A.

For subtask B, dataset statistics and the official splits are shown in Table 2b. We use the auto-generated training set for training all our models in subtask B.

4.2 Implementation

For the implementation of our systems, we used the HuggingFace Transformers[2] library(Wolf et al., 2020) and we used the AdamW optimizer available in PyTorch[3] (Paszke et al., 2019) with the default parameters (learning rates are specified below). All models were fine-tuned using a single Nvidia GeForce RTX 2080 Ti GPU.

We used the base variant of TAPAS, which has a hidden dimension of 768 in all our models. All the

---

[2] 🤗 Transformers, v4.2.0, https://huggingface.co/transformers/

[3] PyTorch, v1.7.1, https://pytorch.org/

F1 Score	Validation Set			Test Set		
	TAPAS-stf	**TAPAS-tf**	**TAPAS-tf-stf**	**TAPAS-stf**	**TAPAS-tf**	**TAPAS-tf-stf**
Without header standardization						
2-way micro	**72.1** $_{\pm 0.43}$	69.42	71.01 $_{\pm 0.99}$	68.01 $_{\pm 0.28}$	70.97	**72.97** $_{\pm 1.37}$
3-way micro	**66.41** $_{\pm 0.48}$	58.97	65.76 $_{\pm 0.37}$	61.59 $_{\pm 0.02}$	57	**65.15** $_{\pm 0.81}$
Refuted	67.95 $_{\pm 0.98}$	64.31	**70.32** $_{\pm 0.91}$	62.04 $_{\pm 0.45}$	64.05	**69.13** $_{\pm 0.74}$
Entailed	67.8 $_{\pm 0.36}$	58.94	**68.09** $_{\pm 1.24}$	64.89 $_{\pm 0.49}$	61.9	**67.23** $_{\pm 1.18}$
Unknown	**49.76** $_{\pm 0.73}$	0	47.52 $_{\pm 3.52}$	**47.58** $_{\pm 0.8}$	0	46.43 $_{\pm 1.88}$
With header standardization						
2-way micro	71.34 $_{\pm 0.96}$	72.78	**74.35** $_{\pm 1.14}$	68.67 $_{\pm 0.9}$	73.79	**73.87** $_{\pm 0.87}$
3-way micro	66.16 $_{\pm 0.64}$	61.11	**69.16** $_{\pm 0.58}$	61.99 $_{\pm 0.8}$	59.32	**66.95** $_{\pm 0.27}$
Refuted	68.22 $_{\pm 0.29}$	65.98	**73.2** $_{\pm 0.83}$	61.42 $_{\pm 1.9}$	65.7	**70.39** $_{\pm 0.44}$
Entailed	67.98 $_{\pm 0.43}$	63.67	**70** $_{\pm 1.69}$	65.67 $_{\pm 0.21}$	65.38	**68.9** $_{\pm 0.48}$
Unknown	49.9 $_{\pm 3.07}$	0	**50.91** $_{\pm 3.99}$	48.27 $_{\pm 1.55}$	0	**50.89** $_{\pm 3.93}$

Table 3: Performance on subtask A: Mean and standard deviation of the metrics from 3 independent runs. In the case of TAPAS-tf, we calculate the metrics using the publicly available TAPAS checkpoint fine-tuned on TABFACT.

Model	Validation Set			Test Set		
	F1	**F1$_{entailed}$**	**F1$_{refuted}$**	**F1**	**F1$_{entailed}$**	**F1$_{refuted}$**
WTQ-base	55.39 $_{\pm 0.53}$	64.07 $_{\pm 0.65}$	48.66 $_{\pm 0.47}$	61.36 $_{\pm 1.47}$	68.47 $_{\pm 2.49}$	**52.75** $_{\pm 1.15}$
WTQ-statement	55.18 $_{\pm 1.78}$	63.36 $_{\pm 3.16}$	48.45 $_{\pm 0.8}$	58.93 $_{\pm 2.49}$	65.22 $_{\pm 4.38}$	51.27 $_{\pm 0.54}$
WTQ-separate	**56.46** $_{\pm 0.43}$	**66.91** $_{\pm 0.3}$	**48.74** $_{\pm 1.01}$	**62.26** $_{\pm 0.79}$	**71.87** $_{\pm 1.2}$	50.79 $_{\pm 1.86}$
During Post-Evaluation Phase						
TABFACT-base	58.41 $_{\pm 0.84}$	64.88 $_{\pm 1.37}$	54.02 $_{\pm 0.91}$	61.46 $_{\pm 0.33}$	67.32 $_{\pm 1.01}$	54.47 $_{\pm 0.55}$
TABFACT-statement	58.92 $_{\pm 1.69}$	65.41 $_{\pm 1.95}$	**54.18** $_{\pm 1.69}$	62.78 $_{\pm 1.71}$	68.44 $_{\pm 2.34}$	**55.8** $_{\pm 1.36}$
TABFACT-separate	**59.47** $_{\pm 0.23}$	**68.06** $_{\pm 0.79}$	53.16 $_{\pm 1.18}$	**65.01** $_{\pm 0.6}$	**74.18** $_{\pm 0.6}$	54.48 $_{\pm 0.58}$

Table 4: Performance on subtask B: Mean and standard deviation of the metrics from 3 independent runs

TAPAS checkpoints we used had been trained with intermediate pre-training and used relative position embeddings (the position index reset when a new cell starts).

For subtask A, we first fine-tuned the classifier head with the TAPAS layers frozen for 3 epochs with a learning rate of 1^{-5} and then fine-tuned the whole model for 10 epochs with a learning rate of 1^{-6}. We used a batch size of 8. We saved a checkpoint every 100 steps and selected the best checkpoint based on the validation set performance.

For subtask B, we fine-tuned the whole model for 5000 steps with a learning rate of 1^{-6}. We used a batch size of 8. We saved a checkpoint every 50 steps and selected the best checkpoint based on the validation set performance.

4.3 Evaluation Metrics

In subtask A, two evaluation metrics are used. The first evaluation metric used is the standard F1-micro score for three-way classification. The second metric again calculates the F1-micro score but does not consider statements with their ground truth label as the unknown class for evaluation; however, classifying the entailed/refuted statements as unknown is penalized.

In subtask B, the evaluation metric used is the standard F1 score with relevant cells as the positive class. If multiple minimal sets of cells can be used to determine the statement's truth value, the dataset contains all of these versions. The score for that statement is calculated by comparing the prediction against each ground truth version and considering the highest score.

	Validation Set		Test Set	
	Length(≤ 512)	**Length**(> 512)	**Length**(≤ 512)	**Length**(> 512)
Distribution - Number of samples				
Subtask A	431(77.52%)	125(22.48%)	616(94.33%)	37(5.67%)
Subtask B	345(81.37%)	79(18.63%)	442(94.04%)	28(5.96%)
Performance of each task's best model				
Subtask A 2-way F1-micro	$77.83_{\pm0.57}$	$65.49_{\pm3.13}$	$73.83_{\pm0.83}$	$74.44_{\pm1.57}$
Subtask A 3-way F1-micro	$73.13_{\pm1.13}$	$55.53_{\pm1.4}$	$66.71_{\pm0.35}$	$54.74_{\pm1.12}$
Subtask B F1	$62.79_{\pm0.39}$	$45.91_{\pm0.68}$	$65.38_{\pm0.63}$	$58.98_{\pm1.22}$

Table 5: Results on long sequences

5 Results

Subtask A The performance of the various systems we considered in subtask A is shown in Table 3. Header standardization improves the performance of all the systems we compared. Transfer learning from TABFACT also improves the performance of our systems. Surprisingly, TAPAS-tf without any fine-tuning on SEM-TAB-FACTS has a better two-way F1-micro score than TAPAS-stf. This shows us the potential of transfer learning from TABFACT in subtask A.

From the confusion matrix shown in Figure 4a, we observe that our model struggles with the unknown class and often misclassifies it as refuted.

Subtask B The performance of the various systems we considered in subtask A is shown in Table 4. Modifying the statement to include entailed/refuted class information leads to a small drop in performance for the models fine-tuned on question-answering earlier and led to a small increase in performance in models fine-tuned on TABFACT. Separate models for entailed/refuted statements perform the best among the systems we considered. It significantly improves the performance on entailed statements, with a little drop in performance on refuted statements. Surprisingly, we observe that transfer learning from TABFACT performs better than transfer learning from WTQ, even though it is a cell selection task. We believe this is because the model has to predict the cells that can be used as evidence for table entailment. The token-level embeddings of the model fine-tuned on TABFACT are better for this task than the model fine-tuned on WTQ, which is instead a question-answering dataset.

(a) Subtask A   (b) Subtask B

Figure 4: Confusion matrices of the test set predictions by our best model for each subtask. The percentages show the ratio of the target class, which was predicted as that class.

Long Inputs The maximum number of tokens supported by our system is 512. In sequences longer than 512 tokens, the tables are truncated row by row to fit in 512 tokens. We compare our system's performance on these long sequences and sequences that fit within 512 tokens. The results are shown in Table 5. We find a significant drop in performance on sequences longer than 512 tokens which had to be truncated.

6 Conclusion

In this paper, we presented our approach for fact verification and evidence finding for tabular data in scientific documents. We show that transfer learning from TABFACT and standardization of the tables to have a single header helps improve our system's performance. We also show that having separate evidence finding models for entailed/refuted statements helps improve our system's performance in the second subtask.

We also find that our model has a significant

drop in performance on large tables since they are truncated to fit in the 512 tokens, the maximum number of tokens supported by TAPAS.

In future work, we would like to experiment with table pruning methods like Heuristic entity linking (Chen et al., 2020) or Heuristic exact match (Eisenschlos et al., 2020) so that the statement and table can fit in 512 tokens. Our systems did not use the table metadata while making the predictions. In the future, we would also like to explore extending the model to encode table metadata along with the table.

Acknowledgments

We thank the organisers of the shared task for their effort, and the anonymous reviewers for their insightful comments.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

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

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190, Berlin, Heidelberg. Springer Berlin Heidelberg.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Julian Eisenschlos, Syrine Krichene, and Thomas Müller. 2020. Understanding tables with intermediate pre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 281–296, Online. Association for Computational Linguistics.

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.

Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. 2017. Search-based neural structured learning for sequential question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1821–1831, Vancouver, Canada. Association for Computational Linguistics.

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.

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.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Qi Shi, Yu Zhang, Qingyu Yin, and Ting Liu. 2020. Learn to combine linguistic and symbolic information for table-based fact verification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5335–5346, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Alane Suhr, Mike Lewis, James Yeh, and Yoav Artzi. 2017. A corpus of natural language for visual rea-

soning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 217–223, Vancouver, Canada. Association for Computational Linguistics.

Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. 2019. A corpus for reasoning about natural language grounded in photographs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6418–6428, Florence, Italy. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Nancy Xin Ru Wang, Diwakar Mahajan, Marina Danilevsky, and Sara Rosenthal. 2021. SemEval-2021 Task 9: A fact verification and evidence finding dataset for tabular data in scientific documents (SEM-TAB-FACTS). In *Proceedings of the 15th international workshop on semantic evaluation (SemEval-2021)*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Xiaoyu Yang, Feng Nie, Yufei Feng, Quan Liu, Zhigang Chen, and Xiaodan Zhu. 2020. Program enhanced fact verification with verbalization and graph attention network. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7810–7825, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Hongzhi Zhang, Yingyao Wang, Sirui Wang, Xuezhi Cao, Fuzheng Zhang, and Zhongyuan Wang. 2020. Table fact verification with structure-aware transformer. In *Proceedings of the 2020 Conference on*

Empirical Methods in Natural Language Processing (EMNLP), pages 1624–1629, Online. Association for Computational Linguistics.

Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2SQL: Generating structured queries from natural language using reinforcement learning.

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. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6053–6065, Online. Association for Computational Linguistics.