Table Fact Verification with Structure-Aware Transformer*

Hongzhi Zhang†, Yingyao Wang◇, Sirui Wang†, Xuezhi Cao†, Fuzheng Zhang†, Zhongyuan Wang†
† Meituan Dianping Group, Beijing, China
◇Harbin Institute of Technology, China
{zhanghongzhi03,wangsirui,caoxuezhi,zhangfuzheng}@meituan.com
yywang@hit-mtlab.net, wzhy@outlook.com

Abstract

Verifying fact on semi-structured evidence like tables requires the ability to encode structural information and perform symbolic reasoning. Pre-trained language models trained on natural language could not be directly applied to encode tables, because simply linearizing tables into sequences will lose the cell alignment information. To better utilize pre-trained transformers for table representation, we propose a **S**tructure-**A**ware **T**ransformer (**SAT**), which injects the table structural information into the mask of the self-attention layer. A method to combine symbolic and linguistic reasoning is also explored for this task. Our method outperforms baseline with 4.93% on TabFact, a large scale table verification dataset.

1 Introduction

Table fact verification aims at classifying whether a textual hypothesis is entailed or refuted by the given table. It could benefit downstream tasks such as fake news detection, misinformation detection, etc. Compared to fact verification over textual evidence (Dagan et al., 2006; Bowman et al., 2015), verification on semi-structured data further requires 1) the ability to encode and understand structural information of tables, and 2) the ability to perform symbolic reasoning over structured data, such as counting, comparing, and numerical calculation. Although large-scale pre-trained language models (Devlin et al., 2019; Yang et al., 2019) achieved dominant results on textual entailment datasets (Wang et al., 2019), they could not be directly used to encode semi-structured data as they are pre-trained on unstructured natural language.

Wenhu et al. (2020) eliminate the discrepancy by serializing tables into word sequences, and then table fact verification could be processed as a natural language inference task. The most straightforward method for table serialization is linearizing the table contents via horizontal scan. However, this would destroy structural information within tables, i.e. the alignments between table cells. In Figure 1, the value "533" and "733" is meaningless digits without the column name "core clock", and it is hard for the model to recover the alignments from the flattened word sequence. Therefore, Table-BERT (Wenhu et al., 2020) includes the column name into cell representation using natural language templates during the linearization. However, comparing or counting column contents of different rows over the flattened word sequence remains a hard task, and simply duplicating the column name multiple times does not achieve satisfying results.

To better utilize the transformer architecture for table representation, we propose to inject the table's structural information into the mask of the self-attention layer. Figure 2 illustrates the pattern commonly adopted when human read or write a table. Usually, each table row describes a record, and cell $c_{1,2}$ describes a record property with the attribute name clarified in the corresponding column name $c_{0,2}$. Besides, values of the same column are usually compared or aggregated for analysis. So, the colored row and column are most crucial to the representation of cell $c_{1,2}$. In the long flattened sequence obtained by horizontal/vertical scan, the alignments between table cells would be disturbed by other unimportant words. To tackle this problem, we have the representation of cell $c_{1,2}$ only depend on the colored cells in Figure 2 by zeroing the attention weights to other ones. Figure 3 illustrates the representation of cell $c_{1,2}$ utilizing transformer. Through masking, only two pseudo sentences, i.e. the corresponding row and column, that share some common words are considered in the representation of each cell. That is, the flattened word sequence is implicitly decomposed into a series of small readable sentences so as to unleashes the power of large pre-trained language model.

* The first two authors contribute equally to this work.

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 1624–1629,
November 16–20, 2020.

Comparison of intel graphics processing units

cpu	market	core clock (mhz)	execution units	memory bandwidth
celeron g1101 pentiumg69xx	desktop	533	12	17 gb/s
core i3 - 5x0 core i5 - 655k	desktop	733	12	21.3 gb/s
core i7 - 620le core i7 - 6x0lm	mobile	266-566	12	17.1 gb/s

Entailed Statement

1. each cpu have 12 execution unit
2. core i3 – 5x0 have faster core clock than core i7 -620le

Refuted Statement

1. core i7 - 620le is designed for mobile market and its memory bandwidth is 21.3 gb/s.
2. There are three series of cpu designed for desktop market.

Figure 1: Examples of table fact verification, the right boxes provide entailed and refuted statements respectively.

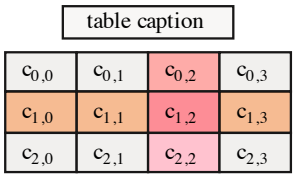

Figure 2: Illustration of table understanding. The colored row and column are crucial to understanding cell $c_{1,2}$.

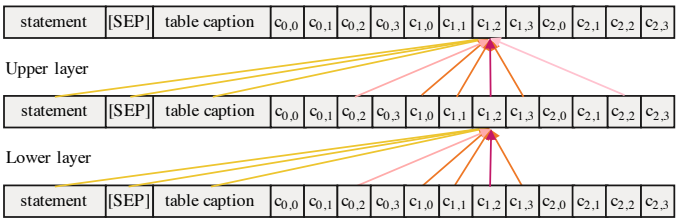

Figure 3: Illustration of masked self-attention for representation of cell $c_{1,2}$. Attentions among cells of the same column are enabled in upper layers to support cross-row reasoning, e.g. $c_{1,2} \sim c_{2,2}$.

Pre-trained transformers are good at semantic-level understanding, i.e. capturing the identical meaning between different expressions. However, one limitation is that they are not doing perfectly in symbolic reasoning (Asai and Hajishirzi, 2020). To tackle this, we perform first-order aggregation over each column and append the result as a special row into the table. An improvement of 1% is achieved, indicating that the ability of hard symbolic reasoning requires further studying.

Our contributions are summarized as follows:

- A **S**tructure-**A**ware **T**ransformer (**SAT**) is devised to better represent semi-structured tables, which injects structural information into attention mask of pre-trained transformers.
- For statements that require symbolic reasoning, we explore a method to combine symbolic reasoning and semantic matching.
- Experimental results show that our method outperforms the state-of-the-art method by 4.93%. Our code is available at https://github.com/zhhongzhi/sat.

2 Methodology

As the examples shown in Figure 1, given a statement S, table fact verification aims to classify whether the statement is entailed or refuted by the evidence table T. The table T consists of a caption t and cells $\{c_{i,j}\}$ of $m \times n$, where m and n are the numbers of rows and columns. Since pre-trained transformer could only take word sequences as input, we feed it with a concatenation of the statement S, the [SEP] token, the table caption t, and the flattened table T_f. The table could be serialized by the horizontal or vertical scan. Figure 3 shows an example of horizontal scanning.

The representation of the word sequence follows the general encoding procedure of the pre-trained transformers (Devlin et al., 2019), so we only describe the self-attention layer in which an attention mask is introduced for table representation. As illustrated in Figure 2, understanding the table requires both horizontal and vertical views. That is, if the table is flattened by a horizontal scan, the vertical alignment information will be lost, and vise versa. For example, the column name $c_{0,2}$ is crucial to the representation of $c_{1,2}$, but its signal could be perturbed by other cells in grey, since all $c_{0,*}$ and $c_{2,*}$ cells are far from $c_{1,2}$ in the flattened sequence and are processed equally.

Therefore, we propose to recover the alignment information by masking signals of unimportant cells during self-attention. The attention mask $\boldsymbol{M} \in \mathbf{R}^{L \times L}$ is defined as:

$$M_{i,j} = \begin{cases} 0 & w_i \sim w_j \\ -\infty & w_i \not\sim w_j \end{cases} \quad (1)$$

where L is the sequence length, and $w_i \sim w_j$ denotes that w_j is attended to when generating representation of w_i, while $w_j \not\sim w_i$ means the opposite. Denote the input of l-th self-attention layer as $\boldsymbol{H}^l \in \mathbf{R}^{L \times d}$, where d is the hidden size. The

attention mask is then applied to the self-attention layer as follows:

$$\boldsymbol{Q}^l, \boldsymbol{K}^l, \boldsymbol{V}^l = \boldsymbol{H}^l\boldsymbol{W}_q, \boldsymbol{H}^l\boldsymbol{W}_k, \boldsymbol{H}^l\boldsymbol{W}_v$$
$$\boldsymbol{A}^l = \text{softmax}(\frac{\boldsymbol{Q}^l{\boldsymbol{K}^l}^T + \boldsymbol{M}}{\sqrt{d_k}}) \quad (2)$$

where $\boldsymbol{W}_* \in \mathbf{R}^{d \times d_k}$ are trainable parameters. The output of self-attention layer is then calculated as:

$$\boldsymbol{H}^{l+1} = \boldsymbol{A}^l\boldsymbol{V}^l \quad (3)$$

It could be observed that if $w_j \not\sim w_i$, then $A_{i,j}$ is reset to zero and $\boldsymbol{H}_j^l$ will not contribute to the representation of w_i, i.e. $\boldsymbol{H}_i^{l+1}$.

Figure 3 sketches the representation learning of tokens in cell $c_{1,2}$ leveraging the masked self-attention. In the lower layers, the token representation of each cell considers information from four aspects: a) tokens of the same row that describe the same entry, b) its column title that clarifies the attribute name, c) the table caption which provides global background, and d) the statement for verification. In the upper layers, cross row attention among cells of the same column is further enabled. In this manner, lower layers focus on capturing low-level lexical information and upper layers are capable of simple cross-row reasoning. Note that tokens of the statement S and the table caption receive information from all cells.

Another preferred ability of SAT is to perform symbolic reasoning such as counting, comparing, and numerical calculation. Pre-trained models like BERT are good at semantic-level understanding, but not symbolic reasoning (Geva et al., 2020; Asai and Hajishirzi, 2020). We explore to enhance the performance of counting verification by converting the counting problem into a semantic matching problem. Specifically, for every column, the frequency of duplicate cell contents is counted as a summary cell, leading to a summary row which is then appended to the table. For example, the summary cell of the second column in Figure 1 is "count desktop:2", so the second refuted statement could be verified via semantic matching.

3 Experiments

3.1 Dataset

Experiments are carried out using TabFact[1] (Wenhu et al., 2020), a large scale table fact verification dataset. The basic statistics of TabFact are listed in Table 1. The dataset contains both simple and complex statements. Simple statements only involve a single row/record, while the complex ones require higher-order semantics (argmax, count, etc.), and the statements are rephrased so more ability on linguistic reasoning is required.

Split	#Statement	#Table	Simple/Complex
Train	92,238	13,182	–
Val	12,792	1,696	–
Test	12,779	1,695	4,230/8,609

Table 1: Basic statistics of TabFact.

3.2 Experimental Settings

Model weights are initialized using BERT-base model trained on English corpus. The first 6 layers are regarded as lower layers, and the other 6 layers are taken as upper layers. We finetune the model with a batch size of 10 and a learning rate of 2e-5. It usually takes 15-18 epochs until convergence.

The flatten sequence is usually longer than the sequence limit of BERT, which requires more memory and training time. Hence, we only retain the top 5 table rows according to the number of words shared with the statement. During experiments, the maximum sequence length is set to 256.

3.3 Results and Ablation Study

The experimental results on TabFact are listed in Table 2. Our method achieves an accuracy of 73.23% on the test set and outperforms Table-BERT by 4.93%. The improvement on complex statements is even larger, which achieves 5.75%.

Effect of Attention Mask Without the attention mask, test accuracy is 67.67% and 64.27% for horizontal and vertical scans respectively, namely a decrease of 5.15% and 8.96% compared to the complete SAT. An interesting finding is that the horizontal scan outperforms the vertical scan when removing the mask, which is consistent with our intuition that each row describes an entry and thus horizontal alignment information is more important. With the cell alignment information recovered by the attention mask, the gap is rather small when using SAT, demonstrating its robustness towards different scan directions.

The last two rows of Table 2 present two variants of the masks, where we adopt an identical mask matrix for all transformer layers instead of using different ones for low/high layers. Results indicate

[1] https://github.com/wenhuchen/Table-Fact-Checking

Model	Val	Test	Test(simple)	Test(complex)
LPA(Wenhu et al., 2020)†	65.1	65.3	78.7	58.5
Table-BERT(Wenhu et al., 2020)†	66.1	65.1	79.1	58.2
Table-BERT tuned*	68.38	68.30	82.35	61.48
BERT with cell position encoding	59.31	59.44	63.24	57.58
SAT with Horizontal scan	72.96	72.82	85.44	66.62
- w/o visible matrix	68.41	67.67	75.93	63.61
- w/o summary row	72.00	72.09	85.53	65.49
- w/o visible matrix w/o summary row	66.84	66.01	74.37	61.90
SAT with Vertical scan	**73.31**	**73.23**	**85.46**	**67.23**
- w/o visible matrix	64.21	64.27	68.77	62.06
- w/o summary row	71.71	71.59	84.70	65.15
- w/o summary row and w/o visible matrix	63.03	62.34	66.71	60.19
- all layers w/o cross row attention	72.83	72.26	84.61	66.11
- all layers w cross row attention	72.02	71.82	83.45	66.10

Table 2: The accuracy (%) of different models. The results annotated with † are cited from literature, and *Table-BERT tuned** denotes results obtained by changing the leaning rate from 5e-5 to 1e-5.

that designing different mask matrix for low/high layers, with the intention to model low-level lexical information and high-level cross-row reasoning, has indeed achieved better performance.

Essentially, by masking signals of unimportant cells, SAT implicitly segments the unnatural long sequence into a series of meaningful subsequences. Such sub-sequences are more friendly to pre-trained language models, so the power of large pre-trained transformer can be unleashed.

The Summary Row Appending a summary row to the table brings a stable improvement of 1%, which mainly contributes to the complex test set. This indicates that although pre-trained transformer is dominant on semantic understanding, its ability on symbolic reasoning is limited. With the counting problem in scope, experimental results show that it is promising to combine both symbolic reasoning and semantic understanding abilities by feeding symbolic reasoning results into SAT.

SAT vs Table Position Embeddings Experiments are further carried out to identify whether the table position encoding method introduced in TaPaS(Herzig et al., 2020) is better than the proposed SAT on table encoding. Row and column positional embeddings are added to the original positional embeddings of BERT to identify the table alignment information. The experimental results are listed in the fourth row of Table 2. An accuracy of 59.8% is observed while the accuracy of the BERT baseline is 68.30%. The results show that BERT is perturbed by the additional table positional embeddings and the model did not converge well. Though the table position information is appended to the inputs, the following transformer layers are not ready to accept and propagate the signal without pre-training. It is demonstrated that simply providing positional information without pre-training is not sufficient for Transformer to encode tables.

3.4 Case study

We analyzed samples that are fixed by SAT compared to baselines. It is observed that a large portion (43/80) of them are statements involve multiple facts/table cells that do not requires logic reasoning. Besides, several problems (9/80) that requires simple count and comparison are fixed. The model both fixed (the other 38) and failed on some samples that require complex symbolic logical reasoning, such as argument sort, conditional aggregation and then comparison. The behavior is most likely random guess for both SAT and baselines. The results show that SAT mainly contributes to the general table representation and enhance the linguistic reasoning, and the summary row appended helps to solve some count problems.

4 Related Work

To encourage the study on table fact verification, Wenhu et al. (2020) construct a large scale table fact checking dataset and study two promising approaches, Table-BERT and Latent Program Algorithm (LPA) respectively. Table-BERT transforms the problem into a natural language inference task to leverage the power of the pre-trained language models. LPA formulates the task as a program

synthesis problem and it is good at symbolic reasoning. Our work aligns with the direction of Table-BERT. Inspired by existing work Weijie et al. (2020); Nguyen et al. (2020); Dong et al. (2019); Yang et al. (2019) that manipulates self-attention masks, we devise a structure-aware transformer to attain better table representation.

There are several recent works that table fact verification could benefit from. Geva et al. (2020) and Asai and Hajishirzi (2020) study to improve the pre-trained model in numerical reasoning and logical comparisons. The enhanced pre-trained model could be directly used in our approach. Herzig et al. (2020) extend BERT's architecture to encode tables for the table question answering task (Iyyer et al., 2017), where additional embeddings identifying the row and column number are added. The proposed architecture is potentially applicable to table fact checking but requires expensive pre-training.

5 Conclusion

We propose SAT to enhance the pre-trained transformer's ability on table representation by injecting structural information into the mask of self-attention layers. Significant improvements on TabFact demonstrate its effectiveness. We further enhance SAT by appending a summary row to the table, the results show that it is promising to solve the fact verification that requires both symbolic reasoning and semantic understanding by feeding symbolic reasoning results into SAT. Overall, an improvement of 4.93% is achieved compared to the state-of-the-art method. The proposed method can further contribute to other semi-structured data (table, graph, etc.) related tasks, e.g. WikiTableQuestions (Pasupat and Liang, 2015) and CommonsenseQA (Talmor et al., 2019). There still exists plenty of potentials that require future studies in this direction.

References

Akari Asai and Hannaneh Hajishirzi. 2020. Logic-guided data augmentation and regularization for consistent question answering.

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.

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.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 13063–13075. Curran Associates, Inc.

Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting numerical reasoning skills into language models.

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*.

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.

Xuan-Phi Nguyen, Shafiq Joty, Steven Hoi, and Richard Socher. 2020. Tree-structured attention with hierarchical accumulation. In *International Conference on Learning Representations*.

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.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. 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 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 3266–3280. Curran Associates, Inc.

Liu Weijie, Zhou Peng, and Qi Ju Haotang Deng Ping Wang Zhe Zhao, Zhiruo Wang. 2020. K-BERT: Enabling language representation with knowledge graph. In *Proceedings of AAAI 2020*.

Chen Wenhu, Wang Hongmin, Hong Wang Shiyang Li Xiyou Zhou Jianshu Chen, Yunkai Zhang, 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.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 5753–5763. Curran Associates, Inc.