Table-based fact verification with self-labeled keypoint alignment

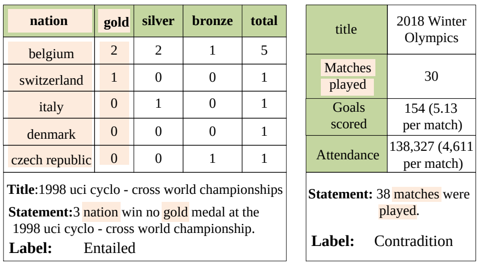
Guangzhen Zhao[†], **Peng Yang**[†*]
† School of Computer Science and Engineering, Key Laboratory of Computer Network
and Information Integration, Ministry of Education, Southeast University, China
{zhaogz, pengyang}@seu.edu.cn

Abstract

Table-based fact verification aims to verify whether a statement sentence is trusted or fake. Most existing methods rely on graph feature or data augmentation but fail to investigate evidence correlation between the statement and table effectively. In this paper, we propose a self-**L**abeled **K**eypoint **A**lignment model, named **LKA**, to explore the correlation between the two. Specifically, a dual-view alignment module based on the statement and table views is designed to discriminate the salient words through multiple interactions, where one regular and one adversarial alignment network cooperatively character the alignment discrepancy. Considering the interaction characteristic inherent in the alignment module, we introduce a novel mixture-of-experts block to elaborately integrate the interacted information for supporting the alignment and final classification. Furthermore, a contrastive learning loss is utilized to learn the precise representation of the structure-involved words, encouraging the words closer to words with the same table attribute and farther from the words with the unrelated attribute. Experimental results on three widely-studied datasets show that our model can outperform the state-of-the-art baselines and capture interpretable evidence words.

1 Introduction

Table-based fact verification aims to uncover the factuality attribute of the sentence relying on the available structured (Chen et al., 2020b; Wang et al., 2021b; Gupta et al., 2020) textual evidence. Current methods can be divided into two groups. The first one exploits the logical form of the statement with graph neural networks (Zhong et al., 2020; Shi et al., 2020, 2021). The other focuses on extending table-aware pre-trained language models (PLMs) (Eisenschlos et al., 2020; Herzig et al., 2020).



nation	gold	silver	bronze	total
belgium	2	2	1	5
switzerland	1	0	0	1
italy	0	1	0	1
denmark	0	0	0	1
czech republic	0	0	1	1

Title:1998 uci cyclo - cross world championships
Statement:3 nation win no gold medal at the 1998 uci cyclo - cross world championship.
Label: Entailed

(a) one example in TABFACT

title	2018 Winter Olympics
Matches played	30
Goals scored	154 (5.13 per match)
Attendance	138,327 (4,611 per match)

Statement: 38 matches were played.
Label: Contradition

(b) one example in INFOTABS

Figure 1: Two examples of table-based fact verification task. Keypoints are highlighted in yellow. (a) is a relational web table. (b) is an entity web table.

In PLM-oriented fact verification, the majority of methods treat the statement-table pair as plain text and then further capture latent essential information relying on multiple Transformer (Vaswani et al., 2017) blocks. It is intuitive that only partial table cells are associated with the statement while other cells are redundant (Wang et al., 2021a; Yin et al., 2020). Due to the lack of explicit guidance signals in the statement-table pair, the capability of checking various statements is hindered for the PLMs, deteriorating model performance and interpretability. Taking Figure 1 as an example, the clues are derived from the statement and some scattered table cells. If the model pays attention to the unrelated words (e.g., “sliver”, “bronze”, “total”), the prediction would not be able to convincingly correct. In other words, failing to align the latent salient words, which are denoted as **keypoints**, may lead to some misleading information being focused on as evidence. Despite impressive process, we empirically find that few methods are committed to the keypoint alignment across the statement words and table cells. The main reasons are: 1) There are alignment discrepancies in the alignment space, where one statement is associated with one table. However, one table may be involved with several statements; 2) The essence of alignment is to perceive salient evidence for the final classifi-

[*] Corresponding author.

Proceedings of the 29th International Conference on Computational Linguistics, pages 1401–1411
October 12–17, 2022.

cation, which requires a well-designed interaction network to aggregate the statement words and table cells; 3) Flexible table structures hinder the representation of words since the significant cost of designing a general structure-aware PLM for table-based fact verification task. In summary, exploring keypoint alignment feature in the statement-table pair is a major challenge.

To tackle the above deficiencies, we propose a model called self-**L**abeled **K**eypoint **A**lignment (**LKA**) for table-based fact verification, focusing on aligning salient evidence and aggregating essential information between the statement and table. Specifically, we design a Dual-view Alignment module (DA) for dealing with the discrepancy of the alignment characteristics. An interaction network is first applied for aggregating the interacted statement and the table representation in multiple steps. The DA then employs a regular alignment network to learn keypoint correlation from the enhanced statement view and force another adversarial alignment network to perceive the correspondingly reverse correlation (i.e., unrelated words) from the table view. For providing aggregated information in the interaction network, we design an Adaptive Aggregation Experts (AAE) block. The AAE employs a mixture-of-experts (MoE) network (Jacobs et al., 1991) that incorporates multiple operating units to sufficiently aggregate the statement and table information. Besides, inspired by the contrastive learning theory (Chen et al., 2020a; Pan et al., 2021), we adopt a structure-aware contrastive learning loss to obtain precise representation for the structure-involved words. The amended representation can force the statement and table closer to its local sub-structure zone (e.g., statement, row, column, etc.) and farther away from others. Our contributions are summarized in three folds:

- We explore a table-based fact verification model integrating keypoint alignment from the statement and table views, which can convert the alignment task into the optimization of two opposite goals and effectively integrate essential information with the MoE network.
- The contrastive learning theory is introduced to enhance structure-aware word representation, which provides a simple and general way to address various structured tabular data.
- We conduct experiments on three benchmark datasets TABFACT, INFOTABS and SEM-TAB-FACTS. Experimental results demonstrate that our model bring performance gains by 0.67%/3.63%/3.07% compared with several state-of-the-art models, and the captured salient words can be interpreted.

2 Related Work

Unlike FEVER (Thorne et al., 2018) utilizing textual evidence or FEVEROUS (Rami Aly and Mittal, 2021) using textual-table mixed evidence, table-based fact verification (Chen et al., 2020b) concentrates on structured or semi-structured evidence text. The currently popular methods employ a tree-style neural network (Zhong et al., 2020) or graph network (Yang et al., 2020; Shi et al., 2020) to encode the logical form of statements. However, labeling massive accurate logical forms is labor-intensive. Meanwhile, structure-aware models (Eisenschlos et al., 2020; Zhang et al., 2020; Dong and Smith, 2021) have been investigated to deal with the table-based fact verification task. Among these approaches, TaPaS++ (Eisenschlos et al., 2020) projects structural information of tables to a pre-trained language model by importing row, column, numeric features into the embedding layer. Some researchers also design novel data augmentation strategies to enhance TaPaS++, such as decomposing complex statements (Yang and Zhu, 2021), replacing non-salient tokens (Wang et al., 2021a), or generating massive question-answer pairs (Liu et al., 2021). However, upgrading vanilla PLMs for precise representation requires a considerable expense of pre-training or data augmenting.

In addition to the rational structured table data, there are entity tables (Gupta et al., 2020) and matrix PDF tables (Wang et al., 2021b) in table-based fact verification scenarios. Mainstream approaches (Gautam et al., 2021; Müller et al., 2021) employ TaPaS++ and TABFACT data to check matrix PDF tables while various scanning (Gupta et al., 2020) and filtering (Neeraja et al., 2021) methods are proposed to deal with entity tables. Different from the aforementioned works, our model exploits a novel alignment of salient words for the statement-table pair and a structure-oriented loss for precise representation.

3 Methodology

3.1 Formulation

In the table-based fact verification task, there are a statement S, its corresponding structured or semi-structured evidence table T and label Y. All samples are denoted as $\boldsymbol{\mathcal{D}} = \{(\boldsymbol{S}_i, \boldsymbol{T}_i, \boldsymbol{Y}_i) | 0 \leq i < I\}$, $I = |\boldsymbol{\mathcal{D}}|$. The task can be formalized as searching for the best mapping $y^* = f_\theta(S, T)$ to minimize the error:

$$Err_{\mathcal{D}} = \boldsymbol{E}_{(s_i,t_i,y_i)\sim\mathcal{D}}\mathcal{L}(f_\theta(s_i,t_i),y_i), \tag{1}$$

where y^* is the predicted label, y is the ground-truth label, f_θ is a specified model in the hypothesis space $\boldsymbol{F}$ with parameter θ.

Furthermore, as shown in Figure 1, it can be observed that only partial cells in table T are relevant to the statement S, which means these salient words are keypoints. The alignment problem is defined as how to identify these keypoints in the statement-table pair. Formally, given an example $e \in \mathcal{D}$ with m labeled tokens $e = \{(x_j, y_j)\}_{j=1}^m$ from $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ denotes an input space and $\mathcal{Y}$ denotes an output space, $y_j \in \{0, 1\}$ indicates token x_j should be aligned or not. We consider the binary classification as the *alignment classifier* $f_a : \mathcal{X} \rightarrow \mathbb{R}^{|y|}$. The accuracy of the alignment classifier is given by:

$$Acc(f_a) \;=\; E_{(x_j,y_j)\in e}\nVdash(f_a(x_j) = y_j), \tag{2}$$

where $\nVdash(.)$ is the indicator function. Then we can define the alignment distance from statement x^s and table x^t as:

$$\begin{aligned} &d_{f_a,f_a'\in\boldsymbol{F}}(S,T) = Acc_S(f_a) + Acc_T(f_a') \\ &= \frac{1}{m}\sum_{j}^{m}\nVdash(f_a(x_j^s) = y_j^s) + \nVdash(f_a'(x_j^t)! = y_j^t)' \end{aligned} \tag{3}$$

where f_a' is an adversarial alignment function that discriminates the unrelated tokens. Thus the total objective can be defined as:

$$\min_{f_\theta\in\boldsymbol{F}} Err_{\mathcal{D}} + \max_{f_a,f_a'\in\boldsymbol{F}} d(S,T). \tag{4}$$

In this manner, the proposed model learns to minimize the error performance and maximize the alignment distance jointly.

3.2 Model Overview

The architecture of LKA is shown in Figure 2. It consists of an encoder, a dual-view alignment module and an MoE-level interaction network. The encoder maps a statement-table pair x into a hidden representation with a vanilla PLM. Inspired by the way humans solve the table-based fact verification task, we design an alignment module to align the underlying keypoint from the statement and table, respectively. Meanwhile, an interaction network driven by MoE is designed to aggregate interactive information for further supporting keypoint alignment and final classification. Additionally, we utilize a contrastive loss on the PLM to yield a more precise structure feature.

3.3 Encoder

The statement and flattened table are formatted as $e = \{\text{[CLS]}, state, \text{[SEP]}, head, ..., r_w, \text{[SEP]}\}$, where $state$ indicates the statement, $head$ indicates the headline of the table, and r_w indicates the w-th row tokens. After encoding, we can obtain the overall statement-table pair representation $\boldsymbol{H} = \text{PLM}(e)$, including the statement representation $\boldsymbol{H}_s$ and the table $\boldsymbol{H}_t$, where $\boldsymbol{H} \in \mathbb{R}^{m\times d}$, $\boldsymbol{H}_s \in \mathbb{R}^{s\times d}$, $\boldsymbol{H}_t \in \mathbb{R}^{t\times d}$, d is the dimension of the hidden representation, m, s, and t are the length of the statement-table pair, the statement and the table, respectively.

3.4 Dual-view Alignment Module

As keypoints are derived from the interaction of the statement and the table, an interaction network is designed to explore the correlation between the two representations. We alternate attentive memory accesses to the statement and the table for multiple steps. From the statement view, we formulate a query glimpse $\boldsymbol{q}_s^\tau$ at step τ:

$$\boldsymbol{q}_s^\tau = \underset{j=1,\dots,s}{\text{softmax}}(\boldsymbol{H}_s^\tau\mathbf{W}_q^s\cdot(\boldsymbol{H}_{s_j}\mathbf{W}_k^s + b_j)^T)\boldsymbol{H}_s\mathbf{W}_v^s, \tag{5}$$

where $\mathbf{W}_q^s, \mathbf{W}_k^s, \mathbf{W}_v^s \in \mathbb{R}^{d\times d}$ are projection matrices, $\boldsymbol{H}_{s_j} \in \mathbb{R}^d$ is the j-th token vector in the statement, b_j is a bias term. $\boldsymbol{H}_s^\tau$ is initialized with $\boldsymbol{H}_s$ when $\tau = 0$.

After interacting with the statement, the alternative attention probes the target table. The table attention weights are calculated based on the table $\boldsymbol{H}_t$ and the currently selected query glimpse $\boldsymbol{q}_s^\tau$:

$$\boldsymbol{q}_t^\tau = \underset{j=1,\dots,t}{\text{softmax}}(\boldsymbol{q}_s^\tau\mathbf{W}_q^t\cdot(\boldsymbol{H}_{t_j}\mathbf{W}_k^t + b_j)^T)\boldsymbol{H}_t\mathbf{W}_v^t, \tag{6}$$

where $\mathbf{W}_q^t, \mathbf{W}_k^t, \mathbf{W}_v^t \in \mathbb{R}^{d\times d}$ are projection matrices, $\boldsymbol{H}_{t_j} \in \mathbb{R}^d$ is the j-th token vector in the table. The interaction network then updates the statement on the basis of the attentive information gathered from the current step τ, i.e., $\boldsymbol{H}_s^{\tau+1} = \psi([\boldsymbol{q}_s^\tau, \boldsymbol{q}_t^\tau])$, where $\psi(.)$ is a non-linear aggregation function.

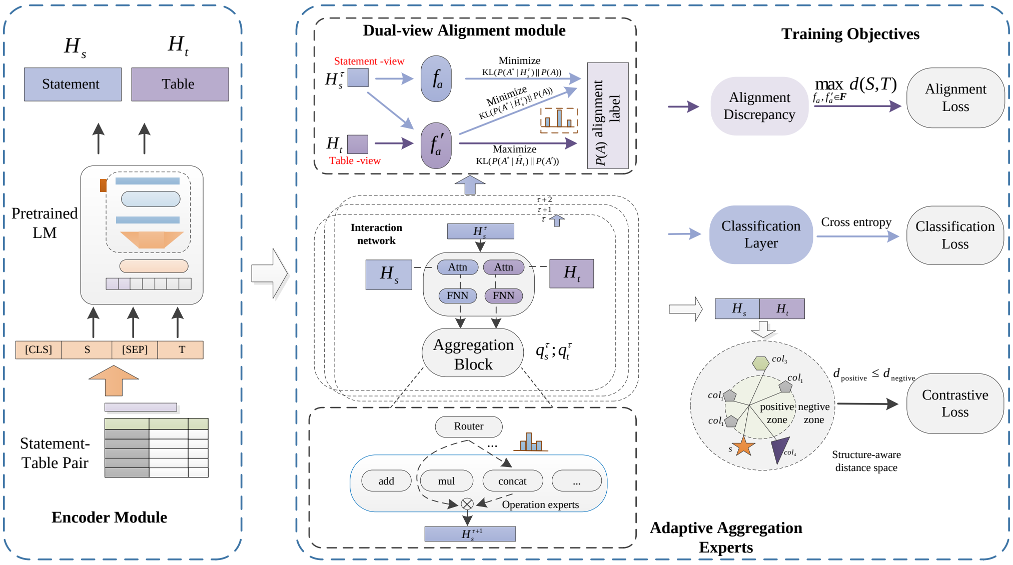


Figure 2: The framework of the proposed model LKA for table-based fact verification.

The updated statement representation aggregates the interacted information from the statement and table for multiple steps. By this means, the updated statement can benefit better alignment and the final verification.

The alignment module tries to make sense of the critical evidence cells so as to provide interpretable evidence for the verification. One noticeable difficulty is how to determine keypoints in the table since accurate labeling of these keypoints is labor-intensive. To this end, we use the same content (i.e., salient tokens) appearing in both the statement and the table as weak supervised keypoints. Given a statement-table pair example $e = \{t_0, t_1, ..., t_m\}$, the alignment label of each token is $A(t_i) \in \{0, 1\}$, where 0 means that token is not essential and vice versa. For token t in example e, the alignment module produces a likelihood probability distribution $\mathcal{A}(t)$ and thus predicts the corresponding alignment label. Since the keypoints are primarily determined by the statement, $\mathcal{A}(t)$ can be predicted with the guidance of the statement representation $\boldsymbol{H}_s^\tau$. The $\mathcal{A}(t)$ is implemented by:

$$\mathcal{A}(t) = \text{Sigmoid}(\text{MLP}((\boldsymbol{H}_s^\tau)) \in \mathbb{R}^{m\times 1}. \quad (7)$$

Then the predicted label $a(t)$ is gained by an alignment softmax function σ:

$$a(t) = \sigma(\mathcal{A}_o(t)) = \frac{\exp(\mathcal{A}_o(t))}{\sum_{o=0}^{o=1} \exp(\mathcal{A}_o(t))}, \quad (8)$$

where $\mathcal{A}_1(t) = 1 - \mathcal{A}_0(t)$. To alleviate the noise from the yielded label probability distribution $A(t_i)$, we add a tolerance item β on $A(t_i)$ with random sampling and revise the alignment probability distribution as $\tilde{A}(t)$. The formula is defined as below:

$$\tilde{A}(t) = \frac{A_o(t) + G(0, \beta)}{\sum_{o=0}^{o=1} A_o(t) + G(0, \beta)}, \quad (9)$$

where $A_o(t) \in \{0, 1\}$ is the original probability, G is the Gaussian sampling function that can revise $A(t)$'s 0-1 hard label to be more tolerant. Sequentially, we use the Kullback-Leibler (KL) divergence (Kullback and Leibler, 1951) to measure the difference between the predicted alignment probability $\mathcal{A}(t)$ and the ground truth alignment label $A(t)$:

$$L(A(t), \mathcal{A}(t)) = \frac{1}{m} \sum_{i=0}^{m} \text{KL}(\tilde{A}(t_i)||a(t_i)). \quad (10)$$

Moreover, the keypoints can be recognized from the table view by first scanning the table and then searching the relevant statement. Thus, the alignment module also could align the keypoints with the table representation. However, as shown in Figure 3, the statement is only related to one table, but one table has involved in different statements. If the alignment module directly aligns the different statements from the table perspective, the process of optimization becomes more difficult due to the multiple alignment decision bounds brought by these statements. Under this consideration, the alignment module concentrates on the non-salient tokens when using table representation. In other words, an adversarial network $\mathcal{A}'(t)$ is designed to make the misalignment with the table representation $\boldsymbol{H}_t$ and

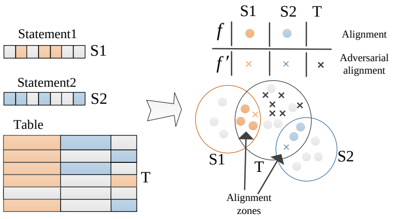


Figure 3: Alignment discrepancy between the statement and table. The non-salient tokens, alignment tokens of the S1, alignment tokens of S2 are highlighted in gray, yellow and blue.

predict correct alignment with the statement representation $\boldsymbol{H}_s^{\tau}$. The above parallel network working mode can be viewed as dual-view alignment. The details of the dual-view alignment can be clarified in Figure 3. Focusing on the alignment zones for the $\mathcal{A}^{'}(t)$ with statement representation is helpful to learn the potential bound of alignment and non-alignment. Specific ally, we first generate the false alignment label $A^{'}(t) = 1 - A(t)$ and then use the statement representation $\boldsymbol{H}_s^{\tau}$ to predict the self-labeled alignment probability $a_s^{'}(t) = \sigma(\mathcal{A}_s^{'}(t))$ by the adversarial network. Meanwhile, we use the table representation $\boldsymbol{H}_t$ to predict the false alignment probability $a_t^{'}(t) = \sigma(\mathcal{A}_t^{'}(t))$.

$$\mathcal{A}_s^{'}(t) = \text{Sigmoid}(\text{MLP-adv}(\boldsymbol{H}_s^{\tau})), \tag{11}$$

$$\mathcal{A}_t^{'}(t) = \text{Sigmoid}(\text{MLP-adv}(\boldsymbol{H}_t)). \tag{12}$$

In short, the adversarial alignment can be summarized as:

$$L_{adv}(A^{'}(t), \mathcal{A}^{'}(t)) = \frac{1}{m}\sum_{i=0}^{m} \text{KL}(a_s(t_i)||a_s^{'}(t_i)) + \text{KL}(A^{'}(t_i)||a_t^{'}(t_i)). \tag{13}$$

3.5 Adaptive Aggregation Experts Module

In this subsection, we implement the aggregation function ψ inspired by the mixture-of-experts (MoE) (Shazeer et al., 2017) mechanism. The aggregation function consists of multiple parallel neural layers, which indicate different kinds of interactions for each attentive statement $\boldsymbol{q}_s^{\tau}$ and attentive table $\boldsymbol{q}_t^{\tau}$. The idea of mixture-of-experts is derived from a group of networks ("experts") that jointly make decisions with dynamical weights. Unlike previous approaches that treat each expert as a uniform structure unit (Shazeer et al., 2017; Fedus et al., 2021), we regard the experts as a series of operation units, which take the $(\boldsymbol{q}_s^{\tau}, \boldsymbol{q}_t^{\tau})$ as input $\{\boldsymbol{x}_1, \boldsymbol{x}_2\}$ and effectively aggregate it in various manners.

$$E(\boldsymbol{x}_1, \boldsymbol{x}_2) = \{\boldsymbol{x}_1 \circ \boldsymbol{x}_2, \boldsymbol{x}_1 \otimes \boldsymbol{x}_2, \boldsymbol{x}_1 \oplus \boldsymbol{x}_2, \boldsymbol{x}_1 \ominus \boldsymbol{x}_2\}, \tag{14}$$

where $\circ, \otimes, \oplus, \ominus$ denote the concatenation, the element-wise multiplication, the element-wise addition and the element-wise subtraction operations, respectively. The MoE block routes the token pair $(\boldsymbol{x}_1, \boldsymbol{x}_2)$ to the determined expert from an expert set $\{E_i(\boldsymbol{x}_1, \boldsymbol{x}_2)\}_{i=1}^{N}$ by an MLP neural network. The output of the MLP $h(\boldsymbol{x}_1, \boldsymbol{x}_2)_j$ is normalized via a softmax function over the available N experts. The gate-value for expert i is given:

$$p_i(\boldsymbol{x}_1, \boldsymbol{x}_2) = \frac{e^{h(\boldsymbol{x}_1, \boldsymbol{x}_2)_j}}{\sum_j^N e^{h(\boldsymbol{x}_1, \boldsymbol{x}_2)_j}}. \tag{15}$$

Accordingly, the output of the MoE block is the linearly weighted combination of each selected expert's computation on each token by the gate value:

$$\boldsymbol{H}_s^{\tau+1} = \sum_{i \in N} p_i(\boldsymbol{q}_s^{\tau}, \boldsymbol{q}_t^{\tau}) E_i(\boldsymbol{q}_s^{\tau}, \boldsymbol{q}_t^{\tau}). \tag{16}$$

By this means, the updated statement representation $\boldsymbol{H}_s^{\tau+1}$ can aggregate the attentive information and flow into the next interaction step. Moreover, we average the final step $\boldsymbol{H}_s^{\tau}$ and concatenate it with the overall statement-table pair representation $\boldsymbol{H}_{[\text{CLS}]}$ to predict the label $y^* = \text{MLP}(\text{mean-pooling}(\boldsymbol{H}_s^{\tau}); \boldsymbol{H}_{[\text{CLS}]})$.

3.6 Training Objectives

Objective 1. To capture the alignment features, we minimize the KL-divergence from the statement and table views, respectively.

$$L_{align} = L(A(t), \mathcal{A}(t)) + L_{adv}(A^{'}(t), \mathcal{A}^{'}(t)) \tag{17}$$

Objective 2. Inspired by the contrastive learning theory (Chen et al., 2020a), we design a structure-aware loss, enabling the PLM to grasp the structural attributes of the statement-table pair.

The definitions of the positive correlation for these different tables are different. The reason is that the rational and matrix PDF table cells in the same column have a similar natural attribute. Recognizing the column relationship is helpful for table encoding (Yin et al., 2020; Chen et al., 2020b). For the entity table, a row is comprised of a property cell and its corresponding content. There is no structural correlation among the rows. (Gupta et al., 2020) also confirms that modeling the link between the property and the content could provide a more accurate representation. Subsequently, the structure features can be learned by the objective of contrastive learning:

$$L_{cl} = \frac{1}{m}\sum_{i=1}^{m}\left[d\left(\boldsymbol{t}_i^a, \boldsymbol{t}_i^p\right) - d\left(\boldsymbol{t}_i^a, \boldsymbol{t}_i^n\right) + \xi\right]_+, \tag{18}$$

where m is the length of the example, a, p, n are the anchor, positive and negative token features, respectively. $d(\cdot,\cdot)$ is the distance function, ξ is a margin parameter, and $[x]_+$ is the $max(x, 0)$ function. In a nutshell, the contrastive loss L_{cl} helps to enhance the distance space's intra-structure compactness and inter-structure discreteness.

Objective 3. Finally, we use cross-entropy loss:

$$L_{ce} = \sum_{x,y\in D} -logP_\theta(y|x). \tag{19}$$

Total objective. The overall loss consists of the above three objectives with hyperparameters λ_1 and λ_2, as well as a balance loss L_{moe} (Shazeer et al., 2017) of adjusting the ratio of selected experts:

$$L_{total} = L_{ce} + \lambda_1 L_{align} + \lambda_2 L_{cl} + L_{moe}. \tag{20}$$

4 Experiments

4.1 Dataset and Metrics

To evaluate the validity of LKA, we adopt three standard datasets with various table structures[1]. For labels, each statement in TABFACT and SEM-TAB-FACTS is labeled as *entailed* or *refuted*[2], while INFOTABS divides statements into three kinds: *entailment*, *contradiction* and *neutral*. We leverage accuracy (Acc.) as the evaluation metric on TABFACT and INFOTABS, as well as microF1 score for SEM-TAB-FACTS.

4.2 Experimental Details

The computation environment is implemented with Python 3.6, PyTorch 1.8.0, CUDA 10.2 and cuDNN 8.0. Recall that all the experiments are running on a CentOS 7 server with the Intel(R) Xeon(R) Gold 6240 @ 2.60GHz CPU and one NVIDIA TESLA V100 GPU.

The optimizer is AdamW and the warmup rate is 0.06. Following traditional natural language understanding task GLUE[3], we fine-tune the DeBERTaV1[4] backbone for the DeBERTaV1 baseline and our LKA with the MultiNLI[5] corpus in one epoch before formal training. The hyperparameters are adjusted depending on the performance of the validation dataset. We set the word embedding and the hidden embedding size of the PLM to 1024. For TABFACT, we run five epochs with a batch size of 4, an initial learning rate of 1e-5, an attention head of 16 and each head of 64 in the attentive interaction network. Three epochs with a batch size of 8 and a learning rate of 1e-5 are adopted in INFOTABS and SEM-TAB-FACTS. In the three datasets, the Dropout is set to 0.1, the number of steps in the interaction network is 3. The tolerant item β, and the balanced factor λ_1, λ_2 are set to be 0.1, 0.08, 0.1, respectively. We set the $d(\cdot,\cdot)$ with Euclidean Distance and the margin parameter ξ with 0.1 in the contrastive learning loss. In the interaction network, we search the number of steps $\mathcal{T}$ in [1, 2, 3, 4, 5, 6]. According to the best results of these different parameters settings, we chose the $\mathcal{T}$=3.

4.3 Baseline Models

We compare our model LKA with the advanced baselines for TABFACT, i.e., **TaPaS++** (Eisenschlos et al., 2020), **Decomp.** (Yang and Zhu, 2021), **SalienL.** (Wang et al., 2021a), **TaPEx** (Liu et al., 2021). For INFOTABS, we employ the baselines **TabFact**, **TabAttn** proposed in (Gupta et al., 2020) and **KG_Info**(Neeraja et al., 2021) to estimate our model. We utilize the advanced baselines **Volta** (Gautam et al., 2021) and **TAPAS** (Müller et al., 2021) for SEM-TAB-FACTS.

4.4 Results and Analysis

Table 1 presents the results of various verification models on the TABFACT dataset. From Table 1, we can observe that our model LKA surpasses matrices from 0.16% to 1.50% compared to TaPEx, illustrating the boosted ability brought from the proposed alignment learning strategy. Moreover, our

[1] Dataset statistics are attached to Appendix.

[2] Since *neutral* examples are not given, we conduct the 2-way experiment for a fair comparison.

[3] GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding

[4] https://github.com/huggingface/transformers/blob/master/src/transformers/models/deberta

[5] https://cims.nyu.edu/~sbowman/multinli/

Models	Val	Test	Test simple	Test complex	Test small
TaPaS++	81.1	81.1	92.6	75.7	84.2
Decomp.	82.7	82.7	93.6	77.4	84.7
SalienL.	82.7	82.1	93.3	76.7	84.3
TaPEx(BART)	81.6	81.2	91.9	75.6	83.9
TaPEx	84.6	84.2	93.9	79.6	85.9
DeBERTaV1	83.28	83.26	92.53	79.15	85.14
Ours	**84.77**	**84.87**	**94.06**	**80.31**	**87.40**
Human	N/A	N/A	N/A	N/A	92.1

Table 1: Comparisons on the TABFACT (%)

Models	Val	Testα_1	Testα_2	Testα_3
TabAttn	63.63	62.94	49.37	49.04
TabFact	77.61	75.06	69.02	64.61
KG_Info	79.44	78.42	71.97	70.03
TaPaS++	74.94	73.22	61.83	60.88
TaPEx	77.38	76.50	67.55	66.38
DeBERTaV1	81.16	80.88	73.61	72.77
Ours	**82.66**	**82.05**	**74.94**	**73.55**
Human	79.78	84.04	83.88	79.33

Table 2: Comparisons on the INFOTABS (%)

Models	Val	Test
Volta	74.35	73.87
TAPAS	78.33	75.33
TaPEx	77.53	75.47
DeBERTaV1	79.12	75.94
Ours	**80.34**	**78.54**

Table 3: Comparisons on the SEM-TAB-FACTS(%)

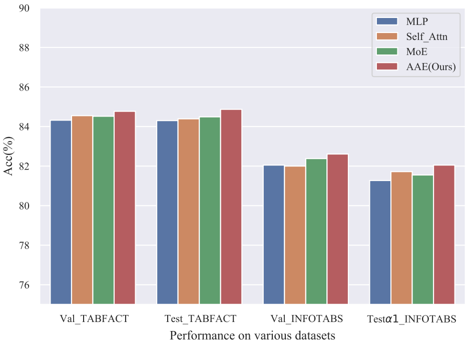


Figure 4: Different aggregation methods in the Interaction network. (%)

model reduces the gap between the machine and human performance to 4.7% on the small test dataset. Meanwhile, LKA achieves the best performance without complicated data augmentation compared with TaPEx. Since most approaches in TABFACT do not have results on INFOTABS and SEM-TAB-FACTS, we run the best approach TaPEx for comparison. As shown in Table 2, LKA outperforms the up-to-date baseline KG_Info from 2.97% to 3.63% on various evaluation subsets. Simultaneously, LKA improves the verification scores on the DeBERTaV1 backbone. Furthermore, we find that TaPEx and TaPaS++ do not perform as well on entity table data INFOTABS as they do on TABFACT. The reason is that the two models are designed to handle rational tables and they have difficulty in adapting to tabular data with varying structures. Considering that approaches on SEM-TAB-FACTS mainly use ensemble models for prediction, we only report the single-model performance in their paper to ensure evaluation fairness. LKA outperforms TaPEx by 3.07% on the test dataset.

In summary, LKA achieves the best results in the three scenarios, which indicates the prominent generalization ability of LKA. Besides, although data augmentation is important to boost performance, the results of DeBERTaV1 demonstrate that a stronger pre-trained language model has potential to tackle various table data and the structure-aware loss further enhances the advantage. Owing to the table structure similarity between TABFACT and SEM-TAB-FACTS, we then only conduct experiments and analyses on TABFACT and INFOTABS.

4.5 Effort of DA

We take a closer look at the dual-view alignment (DA) module by exploiting how the minimization (↓) and maximization (↑) of the KL-divergence between the prediction and the self-labeled alignment affect the final verification performance. As shown in Table 4, we adopt various alignment settings to investigate the alignment discrepancy. We can conclude that: 1) Comparison of the first three rows indicates the dual-view alignment networks are generally superior to the single ones, since the two views can provide complementary alignment information to be aware of salient words. 2) Performance of the third row is mostly lower than the last four rows, which demonstrates the alignment network f cannot resist the negative effect of alignment discrepancy as well as implies the rationality of the adversarial alignment network $f^{'}$. 3) The last row represents the performance of the dual-view alignment module. The highest metrics indicate that using the adversarial network and table representation to align unimportant points, and using the adversarial network and statement representation to align important points are effective to alleviate alignment discrepancy.

Statement	Table	TABFACT		INFOTABS	
		Val	Test	Val	Testα_1
$\downarrow L_f$	–	84.23	84.31	81.83	81.22
–	$\downarrow L_f$	83.95	84.03	81.66	81.05
$\downarrow L_f$	$\downarrow L_f$	84.31	84.25	82.05	81.38
$\downarrow L_f + \downarrow L_{f'}$	$\downarrow L_{f'}$	84.45	84.21	**82.77**	81.22
$\downarrow L_f + \uparrow L_{f'}$	$\uparrow L_{f'}$	84.68	84.59	82.17	81.56
$\downarrow L_f + \uparrow L_{f'}$	$\downarrow L_{f'}$	84.62	84.65	82.21	81.23
$\downarrow L_f + \downarrow L_{f'}$	$\uparrow L_{f'}$	**84.77**	**84.87**	82.66	**82.05**

$\downarrow L_*$ means to align salient tokens, while $\uparrow L_*$ is to align non-important tokens, f and $f^{'}$ are the alignment network and the adversarial networks, respectively.

Table 4: Efforts of various settings under the alignment module(%)

4.6 Effort of AAE

To further exhibit the superiority of the MoE-level aggregation module, we compare it with the following three aggregation methods: 1) **MLP** (Multilayer Perceptron) acts as a fuse block to concatenate the attentive representation from the statement and table; 2) **Self-Attn** (Self-Attention network) adopts the attentive statement representation as query, the attentive table as key and value for aggregation; 3) regular **MoE** employs multiple MLP layers to fuse the representations.

The comparison is illustrated in Figure 4. MoE performs better than MLP and Self-Attn, which demonstrates the advantage of aggregation decisions. In addition, the proposed AAE achieves the optimum performance on the overall metric among all methods. AAE exceeds MLP, Self-Attn and regular MoE about 0.57%/0.78%, 0.48%/0.27% and 0.38%/0.50% on the TABFACT/INFOTABS in terms of test dataset, respectively. One possible reason is that, unlike vanilla MoE where each "expert" employs the same MLP, AAE projects different meta-operation units into MoE. In other words, under the supervision of loss signals, the interacted statement and table learn to adaptively fuse information with fundamental operations, such as "addition","subtraction","multiplication", imitating the process of making decisions by humans.

4.7 Ablation

In order to evaluate the impact of each component of LKA, we ablate it into the following four variants: 1) **w/o Align-Inter** removes the regular and adversarial alignment networks, and the interaction network; 2) **w/o Align** deletes the regular and adversarial alignment networks; 3) **w/o Inter** removes the interaction network; 4) **w/o CL** prunes away the structure-aware contrastive learning loss.

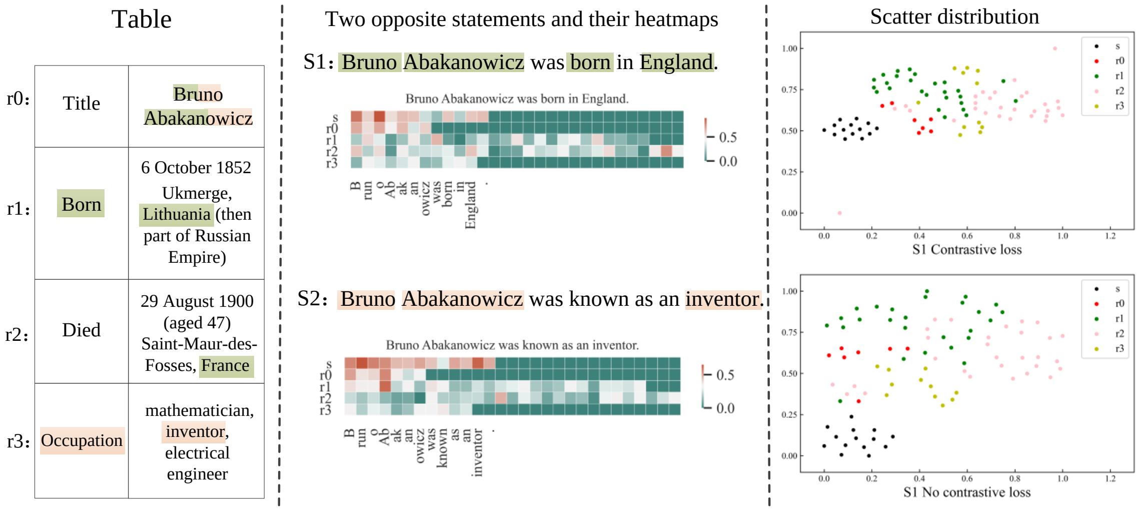
Models	TABFACT		INFOTABS	
	Val	Test	Val	Testα_1
w/o Align-Inter	83.98	83.66	81.50	81.22
w/o Align	84.35	84.39	82.05	81.44
w/o Inter	84.56	84.32	82.11	81.22
w/o CL	84.65	84.46	82.38	81.72
LKA (Ours)	**84.77**	**84.87**	**82.66**	**82.05**

Table 5: Ablation analysis of LKA (%)

As shown in Table 5, we can conclude that removing each component would decrease from 0.33% to 1.21% in Accuracy on the test dataset, which verifies the effectiveness of each component and the reasonable integrity of the LKA. 1) **w/o Align-Inter**: w/o Align-Inter reflects the lowest performance in all simplified variants, decreasing 1.21% and 0.83% on Test, respectively. The experiment results elaborate the validity of our LKA capturing the interactive information and the dual-view alignment. 2) **w/o Align**: w/o Align underperforms LKA, showing 0.48% and 0.61% degradation on Test, respectively. It elaborates the necessity of the LKA capturing the alignment information from the statement and the table views. 3) **w/o Inter**: removing the interaction network decreases 0.55% and 0.83% on Test compared to LKA. The reduction conveys the effectiveness of integrating the attentive representations from the statement and table. 4) **w/o CL**: When eliminating the structure-aware contrastive loss, there are 0.41% and 0.33% accuracy decrease on Test. It reveals that the introduced contrastive loss can improve performance by capturing structure information.

4.8 Case Study

To promote the understanding of LKA, we illustrate two random examples in Figure 5. The dual-view alignment module captures the highlighted words to interpret the evidence fragment. From Figure 5, it can be seen that the proposed alignment module is able to capture essential words with more informative semantics (i.e.,"Bruno Abakanowicz", "born", "Born", "France", "England" for S1, "Bruno Abakanowicz", "inventor" for S2). Although some underlying keypoints are ignored, LKA can gain available evidence fragments such as "France", "Lithuania" for S1 with the MoE-level interaction module. Furthermore, we project the output of PLM into a 2-D dimension vector with TSNE (Van der Maaten and Hinton, 2008)



r0:	Title	Bruno Abakanowicz
r1:	Born	6 October 1852 Ukmerge, Lithuania (then part of Russian Empire)
r2:	Died	29 August 1900 (aged 47) Saint-Maur-des-Fosses, France
r3:	Occupation	mathematician, inventor, electrical engineer

Figure 5: Case analysis via one contradiction and one entailment example on INFOTABS. Due to limited space, we only report the alignment weight heatmap from the statement view. Deeper red color means larger weight in alignment. The padding grids and tokens weighting close to zero are highlighted in green.

for the S1. We conclude that distributions of local sub-structure (e.g., statement, 0-th row, 1-th row) are more condensed than that of the setting with no contrastive loss. The condensed distribution verifies that the proposed LKA can perceive the structure feature of the statement-table pair.

5 Conclusion

This paper takes full advantage of alignment signals to facilitate a self-labeled learning procedure from the statement and table views. More importantly, an MoE-level aggregation module is designed to explore the valuable information. Besides, a contrastive learning loss is introduced to promote the awareness of table structure. Future research could be extended as follows: 1) Exploring alignment mechanism in the table-based question-answer tasks; 2) Developing fact verification approaches in the multi-evidence table setting.

Acknowledgements

The authors thank the anonymous reviewers for their valuable comments. We also thank Yu Yao, Xianyu Zha for their instructive suggestions and invaluable help. This work is supported in part by the National Natural Science Foundation of China under Grant 62272100, and in part by the Consulting Project of Chinese Academy of Engineering under Grant 2020-XY-5, and in part by the Fundamental Research Funds for the Central Universities and the Academy-Locality Cooperation Project of Chinese Academy of Engineering under Grant JS2021ZT05.

References

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020a. A simple framework for contrastive learning of visual representations. In *ICML'20*, volume 119, pages 1597–1607.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020b. Tabfact: A large-scale dataset for table-based fact verification. In *ICLR'20, Addis Ababa, Ethiopia*.

Rui Dong and David Smith. 2021. Structural encoding and pre-training matter: Adapting BERT for table-based fact verification. In *EACL'21*, pages 2366–2375, Online.

Julian Eisenschlos, Syrine Krichene, and Thomas Müller. 2020. Understanding tables with intermediate pre-training. In *Findings of EMNLP'20*, pages 281–296, Online.

William Fedus, Barret Zoph, and Noam Shazeer. 2021. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity.

Devansh Gautam, Kshitij Gupta, and Manish Shrivastava. 2021. Volta at semeval-2021 task 9: Statement verification and evidence finding with tables using tapas and transfer learning. In *SemEval'21*.

Vivek Gupta, Maitrey Mehta, Pegah Nokhiz, and Vivek Srikumar. 2020. INFOTABS: Inference on tables as semi-structured data. In *ACL'20*, pages 2309–2324.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly supervised table parsing via pre-training. In *ACL'20*, pages 4320–4333.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. 1991. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87.

Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Qian Liu, Bei Chen, Jiaqi Guo, Zeqi Lin, and Jian-Guang Lou. 2021. TAPEX: table pre-training via learning a neural SQL executor.

Thomas Müller, Julian Martin Eisenschlos, and Syrine Krichene. 2021. Tapas at semeval-2021 task 9: Reasoning over tables with intermediate pre-training. In *SemEval'21*.

J. Neeraja, Vivek Gupta, and Vivek Srikumar. 2021. Incorporating external knowledge to enhance tabular reasoning. In *NAACL-HLT'21*, pages 2799–2809.

Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. Contrastive learning for many-to-many multilingual neural machine translation. In *ACL'21*, pages 244–258.

Michael Schlichtkrull James Thorne Andreas Vlachos Christos Christodoulopoulos Oana Cocarascu Rami Aly, Zhijiang Guo and Arpit Mittal. 2021. Feverous: Fact extraction and verification over unstructured and structured information.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *ICLR'17*, Toulon, France.

Qi Shi, Yu Zhang, Qingyu Yin, and Ting Liu. 2020. Learn to combine linguistic and symbolic information for table-based fact verification. In *COLING'20*, pages 5335–5346, Online.

Qi Shi, Yu Zhang, Qingyu Yin, and Ting Liu. 2021. Logic-level evidence retrieval and graph-based verification network for table-based fact verification. In *EMNLP'20*, pages 175–184, Punta Cana, Dominican Republic.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *NAACL-HLT'18*, pages 809–819, New Orleans, Louisiana.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS'17*, pages 5998–6008, Long Beach, California, USA.

Fei Wang, Kexuan Sun, Jay Pujara, Pedro Szekely, and Muhao Chen. 2021a. Table-based fact verification with salience-aware learning. In *EMNLP'21*, pages 4025–4036, Punta Cana, Dominican Republic.

Nancy XR Wang, Diwakar Mahajan, Marina Danilevsky, and Sara Rosenthal. 2021b. Semeval-2021 task 9: Fact verification and evidence finding for tabular data in scientific documents (sem-tab-facts).

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 *EMNLP'20*, pages 7810–7825, Online.

Xiaoyu Yang and Xiaodan Zhu. 2021. Exploring decomposition for table-based fact verification. In *EMNLP'21*, pages 1045–1052, Punta Cana, Dominican Republic.

Pengcheng Yin, Graham Neubig, Wentau Yih, and Sebastian Riedel. 2020. TaBERT: Pretraining for joint understanding of textual and tabular data. In *ACL'20*, pages 8413–8426, Online.

Hongzhi Zhang, Yingyao Wang, Sirui Wang, Xuezhi Cao, Fuzheng Zhang, and Zhongyuan Wang. 2020. Table fact verification with structure-aware transformer. In *EMNLP'20*, pages 1624–1629, Online.

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 *ACL'20*, pages 6053–6065, Online.

A Dataset Details

In this section, we describe more detailed settings about the experiments to aid in reproducibility. We also anonymously submit the source code and predicted results on the three datasets to the submission system.

Relational table dataset **TABFACT**[6] contains about 118K natural language statements accompanied by human-annotated 16K regular Wikipedia tables (similar to database tables) of evidence. In addition to the regular validation and test sets, TABFACT extracts subsets of Test_simple and

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

Datasets	Splits				
TABFACT	Train	Val	Test	Simple	Complex
Statement	92,283	12,792	12,779	50,244	68,031
Table	13,182	1,696	1,695	9,189	7,392
INFOTABS	Train	Val	Testα_1	Testα_2	Testα_3
Statement	16,538	1,800	1,800	1,800	1,800
Table	1,740	200	200	200	200
SEM-TAB-FACTS	Train$_a$	Train$_m$	Val	Test	–
Statement	179,345	4,506	463	522	–
Table	1980	981	52	52	–

Table 6: Numbers of examples for all datasets.

Test_complex from the *Simple* and *Complex* channels, as shown in Table 6. **INFOTABS**[7] consists of almost 23K statements and 2.5K unique entity web table drawn from Wikipedia articles in various domains. The entity table could be viewed as a special table since it contains multiple rows and only two columns, of which one denotes the title of a record and the other is the corresponding content. **SEM-TAB-FACTS**[8] is proposed at SemEval-2021 task 9 and focus on matrix tables from scientific articles. The dataset contains an auto-generated train set *Train*$_a$ and a human-annotated train set *Train*$_m$. Note that we only use the *Train*$_m$ to train our LKA model since the *Train*$_a$ is more noisy.

The experimental data we used is taken from their links, and no additional processing is performed on the TABFACT and INFOTABS datasets beyond the steps described in Section 3.3. Considering that the matrix tables in SEM-TAB-FACTS contain multi-row or multi-column header, we follow the paper[9] to standardize the table header by dividing multi-row or multi-column header into multiple headers with the same content. In addition, conducting experiments with LKA, TaPEx[10] and DeBERTaV1 on the SEM-TAB-FACTS, the trained model on the TABFACT is utilized to initialize the training of SEM-TAB-FACTS.

B Algorithm Description

The algorithm description is given for further understanding and facilitating reproducibility of the proposed LKA model.

Algorithm 1 Table-based fact verification with self-labeled keypoint alignment

```
Require:
    Source table, statement and ground-truth label $(S, T, Y)$; model parameters $\boldsymbol{\theta}$; the alignment and the adversarial label $A$, $A^{'}$
Initialize model parameters $\boldsymbol{\theta}$
while not converged do
    Sample a training example $(S, T, Y)$
    Flatten $(S, T)$ to $e = \{[\mathrm{CLS}], state, [\mathrm{SEP}], head, ..., r_w, [\mathrm{SEP}]\}$
    $\boldsymbol{H} = \mathrm{PLM}(e)$, $\boldsymbol{H}_s = \boldsymbol{H} * mask_s$, $\boldsymbol{H}_t = \boldsymbol{H} * mask_t$, let $\boldsymbol{H}_s^{\tau} = \boldsymbol{H}_s$
    for step $\tau = 0 \rightarrow \mathcal{T} - 1$ do
        $\boldsymbol{q}_s^{\tau} \leftarrow \underset{j=1,\ldots,s}{\mathrm{softmax}}(\boldsymbol{H}_s^{\tau}\mathbf{W}_q^s \cdot (\boldsymbol{H}_{s_j}\mathbf{W}_k^s + b_j)^T)\boldsymbol{H}_s\mathbf{W}_v^s$
        $\boldsymbol{q}_t^{\tau} \leftarrow \underset{j=1,\ldots,t}{\mathrm{softmax}}(\boldsymbol{q}_s^{\tau}\mathbf{W}_q^t \cdot (\boldsymbol{H}_{t_j}\mathbf{W}_k^t + b_j)^T)\boldsymbol{H}_t\mathbf{W}_v^t$
        $E(\boldsymbol{x}_1, \boldsymbol{x}_2) = \{\boldsymbol{x}_1 \circ \boldsymbol{x}_2, \boldsymbol{x}_1 \otimes \boldsymbol{x}_2, \boldsymbol{x}_1 \oplus \boldsymbol{x}_2, \boldsymbol{x}_1 \ominus \boldsymbol{x}_2\}$
        $\boldsymbol{H}_s^{\tau+1} = \sum_{i \in N} p_i(\boldsymbol{q}_s^{\tau}, \boldsymbol{q}_t^{\tau})E_i(\boldsymbol{q}_s^{\tau}, \boldsymbol{q}_t^{\tau})$,
    end for
    Obtain the alignment $\mathcal{A}(t)$ and the adversarial alignment $\mathcal{A}^{'}(t)$ with $\boldsymbol{H}_s^{\tau}$ and $\boldsymbol{H}_t$
    $L_{align} = L(A(t), \mathcal{A}(t)) + L_{adv}(A^{'}(t), (\mathcal{A}^{'}(t))$
    $L_{cl} = \frac{1}{m}\sum_{i=1}^{m}\left[d\left(\boldsymbol{t}_i^a, \boldsymbol{t}_i^p\right) - d\left(\boldsymbol{t}_i^a, \boldsymbol{t}_i^n\right) + \xi\right]_+$
    $L_{ce} = \sum_{x,y \in D} -logP_{\theta}(y|x)$
    $L(\boldsymbol{\theta}) = L_{ce} + \lambda_1 L_{align} + \lambda_2 L_{cl} + L_{moe}$
    $\boldsymbol{\theta} \leftarrow \mathrm{AdamW}\left(\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}), \boldsymbol{\theta}\right)$
end while
return $\boldsymbol{\theta}$
```

[7] https://github.com/infotabs/infotabs

[8] https://sites.google.com/view/sem-tab-facts

[9] https://github.com/devanshg27/sem-tab-fact

[10] https://github.com/microsoft/Table-Pretraining