

# Joint Verification and Reranking for Open Fact Checking Over Tables

Michael Schlichtkrull<sup>1\*</sup>, Vladimir Karpukhin<sup>2</sup>, Barlas Oğuz<sup>2</sup>,  
Mike Lewis<sup>2</sup>, Wen-tau Yih<sup>2</sup>, Sebastian Riedel<sup>2,3</sup>

<sup>1</sup>University of Cambridge, <sup>2</sup>Facebook AI Research, <sup>3</sup>University College London  
michael.schlichtkrull@cst.cam.ac.uk,  
{vladk, barlaso, mikelewis, scottyih, sriedel}@fb.com

## Abstract

Structured information is an important knowledge source for automatic verification of factual claims. Nevertheless, the majority of existing research into this task has focused on textual data, and the few recent inquiries into structured data have been for the closed-domain setting where appropriate evidence for each claim is assumed to have already been retrieved. In this paper, we investigate verification over structured data in the open-domain setting, introducing a joint reranking-and-verification model which fuses evidence documents in the verification component. Our open-domain model achieves performance comparable to the closed-domain state-of-the-art on the TabFact dataset, and demonstrates performance gains from the inclusion of multiple tables as well as a significant improvement over a heuristic retrieval baseline.

## 1 Introduction

Verifying whether a given fact coheres with a trusted body of knowledge is a fundamental problem in NLP, with important applications to automated fact checking (Vlachos and Riedel, 2014) and other tasks in computational journalism (Cohen et al., 2011; Flew et al., 2012). Despite extensive investigation of the problem under different conditions including entailment and natural language inference (Dagan et al., 2005; Bowman et al., 2015) as well as claim verification (Vlachos and Riedel, 2014; Alhindi et al., 2018; Thorne and Vlachos, 2018), relatively little attention has been devoted to the setting where the trusted body of evidence is structured in nature — that is, where it consists of tabular or graph-structured data.

Recently, two datasets were introduced for claim verification over tables (Chen et al., 2020b; Gupta et al., 2020). In both datasets, claims can be verified

\*Work done while interning with Facebook AI Research.

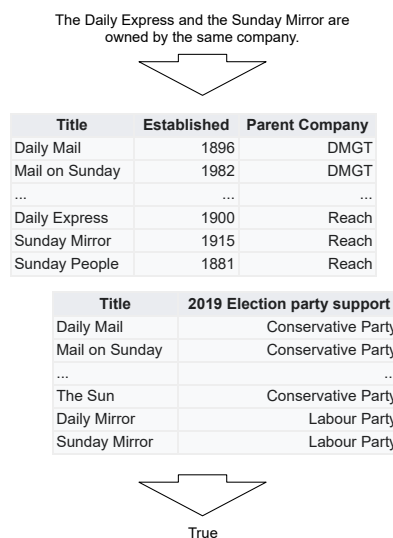


Figure 1: Example query to be evaluated against two retrieved tables. Named entities represent a strong baseline for retrieval, but ultimately a more complex model is required to distinguish highly similar tables.

given a single associated table. While highly useful for the development of models, this *closed* setting is not reflective of real-world fact checking tasks where it is usually not known which table to consult for evidence. Realistic systems must first retrieve evidence from a large data source. That is, realistic systems must operate in an *open* setting.

Here, we investigate fact verification over tables in the open setting. We take inspiration from similar work on unstructured data (Chen et al., 2017; Nie et al., 2019; Karpukhin et al., 2020; Lewis et al., 2020), proposing a two-step model which combines ad-hoc retrieval with a neural reader. Drawing on preliminary work in open question answering over tables (Sun et al., 2016), we perform retrieval based on simple heuristic modeling of individual table cells. We combine this retriever with a RoBERTa-based (Liu et al., 2019) joint reranking-and-verification model, performing

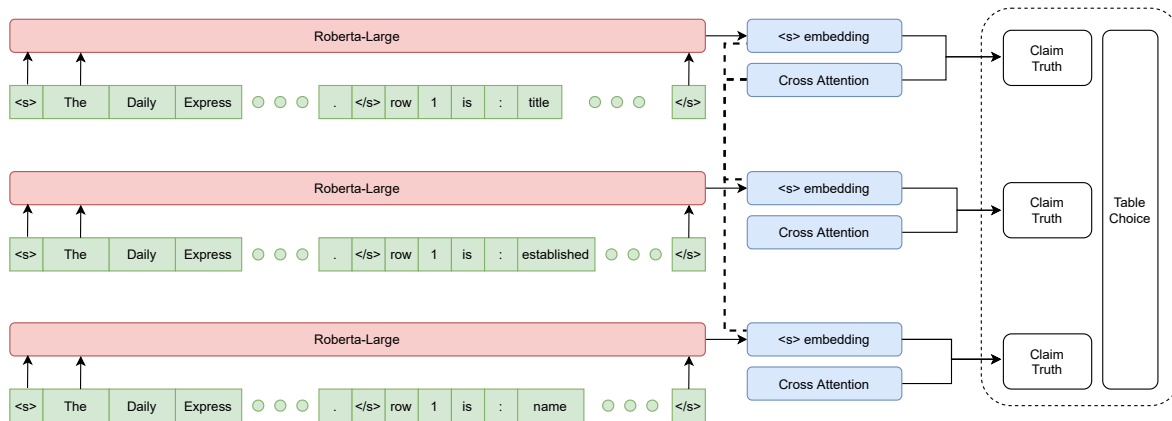


Figure 2: A diagram of our model, using the joint reranking- and verification approach described in Section 4.1. Linearised tables are encoded separately with RoBERTa. Then, cross-attention is used to contextualize each individual table with respect to the others. Finally, the model jointly predicts truth value and table selection.

fusion of evidence documents in the verification component. This corresponds to the approach suggested for question answering by e.g. Izacard and Grave (2020).

We evaluate our models using the recently introduced TabFact dataset (Chen et al., 2020b). While initially developed for the closed domain, the majority of claims are sufficiently context-independent that they can be understood without knowing which table they were constructed with reference to. As such, the dataset is suitable for the open domain as well. Our models represent a first step into the open domain, achieving open-domain performance exceeding the previous closed-domain state of the art—outside of Eisen-schlos et al. (2020), which includes pretraining on additional synthetic data. We demonstrate significant gains from including multiple tables, and these gains are increasing as more tables are used. We furthermore present results using a more realistic setting where tables are retrieved not just from the 16,573 TabFact tables, but from the full Wikipedia dump. Our contributions can be summarized as follows:

1. We introduce the first model for open-domain table fact verification, demonstrating strong performance exceeding the previous closed-setting state of the art.
2. We propose two strategies with corresponding loss functions for modeling table fact verification in the open setting, suitable respectively for high verification accuracy or identifying if appropriate information has been retrieved for verification.

3. In addition to our open-domain performance, our model achieves a new closed-domain state-of-the-art result.
4. We report the first results on Wikipedia-scale open-domain table fact verification, using all tables from a Wikipedia dump as the backend.

We release the source code for our experiments at <https://github.com/facebookresearch/OpenTableFactChecking>.

## 2 Open-Domain Table Fact Verification

Formally, the open table fact verification problem can be described as follows. Given a claim  $q$  and a collection of tables  $T$ , the task is to determine whether  $q$  is true or false. As such, we approach the task by modeling a binary verdict variable  $v$  as  $p(v|q, T)$ . This is in contrast to the closed setting, where a single table  $t_q \in T$  is given, and the task is to model  $p(v|q, t_q)$ . Since there are large available datasets for the closed setting (Chen et al., 2020b; Gupta et al., 2020), it is reasonable to expect to exploit  $t_q$  during training; however, at test time, this information may not be available. We follow a two-step methodology that is often adopted in open-domain setting for unstructured data (Chen et al., 2017; Nie et al., 2019; Karpukhin et al., 2020; Lewis et al., 2020) to our setting. Namely, given a claim query  $q$ , we retrieve a set of evidence tables  $D_q \subset T$  (Section 3), and subsequently model  $p(v|q, D_q)$  in place of  $p(v|q, T)$  (Section 4).

## 3 Entity-based Retrieval

We first design a strategy for retrieving an appropriate subset of evidence tables for a given query. For

question answering over tables, Sun et al. (2016) demonstrated strong performance on retrieving relevant tables using entity linking information, following the intuition that many table cells contain entities. We take inspiration from these results. In their setting, claim entities are linked to Freebase entities, and string matching on the alias list is used to map entities to cells. To avoid reliance on a knowledge graph, we instead use only the textual string from the claim to represent entities, and perform approximate matching through dot products of bi- and tri-gram TF-IDF vectors.

We pre-compute bi- and trigram TF-IDF vectors  $z(c_t^1), \dots, z(c_t^m)$  for every table  $t \in T$  with cells  $c_t^1, \dots, c_t^m$ . Then, we identify the named entities  $e_q^1, \dots, e_q^n$  within the query  $q$ . For our experiments, we use the named entity spans for TabFact, provided by Chen et al. (2020b) as part of their LPA-model.<sup>1</sup> We compute bi- and trigram TF-IDF vectors  $z(e_q^1), \dots, z(e_q^n)$  for the surface forms of those entities. To retrieve  $D_q$  given  $q$ , we then score every  $t \in T$ . Since we are approximating entity linking between claim entities and cells, we let the score between an entity and a table be the *best* match between that entity and any cell in the table. That is:

$$\text{score}(q, t) = \sum_{i=1}^n \max_{j=1}^m z(e_q^i)^\top \cdot z(c_t^j) \quad (1)$$

In other words, we compute for every entity the best match in the table, and score the table as the sum over the best matches. To construct the set of evidence tables  $D_q$ , we then retrieve the top- $k$  highest scoring tables. Our choice to use bi- and tri-gram TF-IDF as the retrieval strategy was determined empirically — see Section 5.1 and Table 1 for experimental comparisons.

## 4 Neural Verification

To model  $p(v|q, D_q)$ , we employ a RoBERTa-based (Liu et al., 2019) late fusion strategy (see Figure 2 for a diagram of our model). Given a query  $q$  with a ranked list of  $k$  retrieved tables  $D_q = (d_q^1, \dots, d_q^k)$ , we begin by linearising each table. Our linearisation scheme follows Chen et al. (2020b). We first perform sub-table selection by excluding columns not linked to entities in the query. Here, we reuse the entity linking obtained during

<sup>1</sup>In the absence of named entity tags, named entity spans would first need to be found through an off-the-shelf named entity recognizer, such as SpaCy (Honnibal et al., 2020).

the retrieval step (see Section 3), and retain only the three columns in which cells received the highest retrieval scores. We linearise each row separately, encoding entries and table headers. Suppose  $r$  is a row with cell entries  $c_1, c_2, \dots, c_m$  in a table, where the corresponding column headers are  $h_1, h_2, \dots, h_m$ . Row number  $r$  is mapped to “row  $r$  is :  $h_1$  is  $c_1$  ;  $h_2$  is  $c_2$  ; ... ;  $h_m$  is  $c_m$  .”

We construct a final linearisation  $L_{q,t}$  for each query-table pair  $q, t$  by prepending the query to the filtered table linearisation. We then encode each  $L_{q,t}$  with RoBERTa, and obtain a contextualised embedding  $f(d_q^k) \in \mathbb{R}^n$  for every table by using the final-layer embedding of the CLS-token. We construct the sequence of embeddings  $f(d_q^1), \dots, f(d_q^k)$  for all  $k$  tables.

When the model attempts to judge whether to rely on a given table for verification, other highly-scored tables represent useful contextual information (e.g., in the example in Figure 1, newspapers belonging to the same owner may be likely to also share political leanings). Nevertheless, each table embedding  $f(d_q^k)$  is functionally independent from the embeddings of the other tables. As such, contextual clues from other tables cannot be taken into account. To remedy this, we introduce a cross-attention layer between all tables corresponding to the same query. We collect the embeddings  $f(d_q^k)$  of each table into a tensor  $F(D_q)$ . We then apply a single multi-head self-attention transformation as defined by Vaswani et al. (2017) to this tensor, and concatenate the result. That is, we compute an attention score for head  $h$  from table  $i$  to table  $j$  with query  $q$  as:

$$\alpha_{ij}^h = \sigma \left( \frac{W_Q^h f(d_q^i) (W_K^h f(d_q^j))^T}{\sqrt{\dim(K)}} \right) \quad (2)$$

where  $\sigma$  is the softmax function, and  $W_Q$  and  $W_K$  represent linear transformations to queries and keys, respectively. We then compute an attention vector for that head as:

$$A_i^h = \sum_{j \in D_q} \alpha_{ij} W_V^h f(d_q^j) \quad (3)$$

and finally construct contextualized table representations through concatenation as:

$$f^*(d_q^k) = [f(d_q^k), A_i^1, \dots, A_i^h] \quad (4)$$

We subsequently use  $F^*(D_q)$ , i.e. the tensor containing  $f^*(d_q^1), \dots, f^*(d_q^k)$ , for downstream predictions. We note that our approach can be viewed

as an extension of the Table-BERT algorithm introduced by Chen et al. (2020b) to the multi-table setting, using an attention function to fuse together the information from different tables.

#### 4.1 Training & Testing

Relying on a closed-domain dataset provides a table with appropriate information for answering each query; namely, the table against which the claim is to be checked in the closed setting. Although this information is not available at test time, we can construct a training regime that allows us to exploit it to improve model performance. We experiment with two different strategies: jointly modeling reranking of tables along with verification of the claim, and modeling for each table a ternary choice between indicating truth, falsehood, or giving no relevant information. Later, we demonstrate how the former leads to increased performance on verification, while the latter gives access to a strong predictor for cases where no appropriate table has been retrieved.

**Joint reranking and verification** For the joint reranking and verification approach, we assume that a *best* table for answering each query is given and can be used to learn a ranking function. We model this as selecting the right table from  $D_q$ , e.g., through a categorical variable  $s$  that indicates which table should be selected. We then learn a joint probability of  $s$  and the truth value of the claim  $v$  over the tables for a given query. Assuming that  $s$  and  $v$  are independent,  $p(s, v|q, D_q)$  is also a categorical distribution with one correct outcome that can be optimized for (that is, one correct pair of table and truth value). As such, we let:

$$p(s, v|q, D_q) = \sigma(W(F^*(D_q)_s)_v) \quad (5)$$

Where  $W : \mathbb{R}^{2n} \rightarrow \mathbb{R}^2$  is an MLP and  $\sigma$  is the softmax function. At train time, we obtain one cross-entropy term corresponding to  $p(s, v|q, D_q)$  per query. At test time, we marginalize over  $s$  to obtain a final truth value:

$$p_v(v|q, D_q) = \sum_{t \in D_q} p(v, s = t|q, D_q) \quad (6)$$

This formulation has the additional benefit of also allowing us to make a prediction on which table matches the query. We can do so by marginalizing

over  $v$ :

$$p_s(s|q, D_q) = \sum_{v_q \in \{true, false\}} p(s, v = v_q|q, D_q) \quad (7)$$

With this loss, we train the model by substituting for  $D_q$  a set  $D_q^*$  containing wherein the gold table is guaranteed to appear. We ensure this by replacing the lowest-scored retrieved table in  $D_q$  with the gold table whenever it has not been retrieved.

**Ternary verification** At test time, there may be cases where a table refuting or verifying the fact is not contained in  $D_q$ . For some applications, it could be useful to identify these cases. We therefore design an alternative variant of our system better suited for this scenario. Intuitively, each table can represent three outcomes – the query is true, the query is false, or the table is irrelevant. We can model this through a ternary variable  $i$  such that for table  $t$ :

$$p(i|q, t, D_q) = \sigma(W'(F^*(D_q)_t)_i) \quad (8)$$

Where  $W' : \mathbb{R}^{2n} \rightarrow \mathbb{R}^3$  is an MLP and  $\sigma$  is the softmax function. During training, we assign *true* or *false* to the gold table depending on the truth of the query, and *irrelevant* to every other table. We then use the mean cross-entropy over the tables associated with each query as the loss for each example. At test time, we compute the truth value  $v$  of each query as:

$$\sum_{t \in D_q} p(i = true|q, t) > \sum_{t \in D_q} p(i = false|q, t) \quad (9)$$

## 5 Experiments

We apply our model to the TabFact dataset (Chen et al., 2020b), which consists of 92,283 training, 12,792 validation and 12,792 test queries over 16,573 tables. The task is binary classification of claims as true or false, with an even proportion of the two classes in each split. To benchmark our open-domain models and construct performance bounds, we begin by evaluating in the closed domain. As an upper bound, we can then compare against the performance of the closed-domain system scored using a single table retrieved through an oracle. As a lower bound, we can again use the closed-domain system, but using the highest-ranked table according to our TF-IDF retriever. The evaluation metric is simply prediction accuracy.



Dataset	H@1	H@3	H@5	H@10
Query-matching word-level TF-IDF	41.7	54.2	59.0	65.3
Query-matching character-level (2,3)-gram TF-IDF	34.7	45.5	50.2	56.8
Entity-matching word-level exact match	48.2	57.9	64.2	67.3
Entity-matching word-level TF-IDF	56.0	65.6	74.1	81.2
Entity-matching character-level (2,3)-gram TF-IDF	<b>69.6</b>	<b>78.8</b>	<b>82.3</b>	<b>86.6</b>
Entity-matching character-level (1,2,3)-gram TF-IDF	62.3	75.2	80.1	86.1

Table 1: Retrieval accuracy for our entity-based TF-IDF retrieval along with several baselines for the TabFact validation set, computed using all 16,573 TabFact tables. We experiment with matching the entire query against table cells (above), and matching individual entities in the query against table cells using Equation 1 (below). For all subsequent experiments we rely on character-level (2,3)-grams with entity-matching for retrieval.

Model	Dev	Test	Simple Test	Complex Test	Small Test
Table-BERT (Chen et al., 2020b)	66.1	65.1	79.1	58.2	68.1
LogicalFactChecker (Zhong et al., 2020)	71.8	71.7	85.4	65.1	74.3
ProgVGAT (Yang et al., 2020)	74.9	74.4	88.3	67.6	76.2
TAPAS (Eisenschlos et al., 2020)*	81.0	81.0	92.3	75.6	83.9
Ours (Oracle retrieval)	78.2	77.6	88.9	72.1	79.4
Ours (1 retrieved table)	74.1	73.2	86.7	67.8	76.6
Ours (Ternary loss, 3 tables)	73.8	73.5	86.9	68.1	76.9
Ours (Ternary loss, 5 tables)	74.1	73.7	87.1	67.9	76.5
Ours (Ternary loss, 10 tables)	73.9	73.1	86.5	67.9	77.3
Ours (Joint loss, 3 tables)	74.6	73.8	87.0	68.3	78.1
Ours (Joint loss, 5 tables)	<b>75.9</b>	<b>75.1</b>	<b>87.8</b>	<b>69.5</b>	<b>77.8</b>
Ours (Joint loss, 10 tables)	73.9	73.8	86.9	68.1	76.9

Table 2: Prediction accuracy of our RoBERTa-based model on the official splits from the TabFact dataset. We include closed-domain performance of several models from the literature, as well as the performance of our model in both the closed and the open domain, using both proposed loss functions. The first section of the table contains closed-domain results, the second open-domain. \* employs intermediary pretraining on additional synthetic data.

## 5.1 Retrieval

We choose bi- and tri-gram TF-IDF as the retrieval strategy empirically. To address the comparative performance of this choice, we compute and rank in Table 1 the retrieval scores obtained through our strategy on the TabFact test set. We compare against several alternative strategies: bi- and tri-gram TF-IDF vectors for all words in the query (rather than just the entities), word-level TF-IDF vectors for entities, and entity-level exact matching. Our bi- and tri-gram TF-IDF strategy yields by far the strongest performance. We furthermore demonstrate how the exclusion of unigrams from the TF-IDF vectors slightly increases performance.

## 5.2 Verification

In Table 2, we compare our best-performing models to the closed-setting system from Chen et al.

(2020b), as well as to several recent models from the literature (Zhong et al., 2020; Yang et al., 2020; Eisenschlos et al., 2020). We include results with both losses as discussed in Section 4, using varying numbers of tables.

With an accuracy of 75.1%, we obtain the best open-domain results with our model using the joint reranking-and-verification loss and five tables. We see performance improvements when increasing the number of tables, both from 1 to 3 and from 3 to 5. In the closed domain, the 77.6% accuracy our model achieves is a significant improvement over the 74.4% the strongest comparable baseline reached. This may be due to our use of RoBERTa, which has previously been found to perform well for linearised tables (Gupta et al., 2020).

Relying purely on TF-IDF for retrieval — that is, using our system with only one retrieved table

Model	R@1	R@2-3	R@4-5
Oracle retrieval	80.6	74.1	75.0
1 table	80.6	55.6	53.9
3 tables	78.8	66.7	58.2
5 tables	79.4	73.1	71.7

Table 3: Performance of our RoBERTa-based model on the parts of the TabFact test set where our TF-IDF retriever assigns the gold table rank respectively 1, 2-3, or 4-5.

— yields a performance of 73.2%. This is a surprisingly small decrease compared to the closed domain, given that an incorrect table is provided in approximately a third of all cases (see Table 1). We suspect that many cases for which the retriever fails are also cases for which the closed-domain model fails. To make sure we are not seeing the effect of false negatives (e.g., tables which are not the gold table, but which nevertheless have the information to verify the claim), we run the model in a setting where one retrieved table is used, but the gold table is removed from the retrieval results; here, the model achieves an accuracy of only 56.2%. We furthermore test a system relying on a random table rather than a retrieved table; with a performance drop to 53.1, we find that the information in the retrieved table is indeed crucial to obtain high performance (rather than the performance being purely a consequence of, say, RoBERTa weights).

To understand how our model derives improvement from the addition of more tables, we compute in Table 3 the performance of our reranking-and-verification model when TF-IDF returns the correct table at rank 1, rank 2-3, or rank 4-5. Immediately, we notice a much stronger improvement from using multiple tables when TF-IDF fails to correctly identify the gold table. This is natural, as those are exactly the cases where our model (as opposed to the baseline) has access to the appropriate information to verify or refute the claim.

Interestingly, using three tables improves on using one table even when the gold table is not included among the top three (from 53.9% to 58.2%), and using five tables improves on using three tables also when the gold table *is* included among the top three (from 66.7% to 73.1%). Manual inspection reveals that our model in some cases relies on correlations between tables — if a sports team loses games in three tables, then that may give a

Model	Accuracy
Full model	75.1
- Attention	73.6
- Joint objective	72.9
- Both	71.2

Table 4: Ablation study for our model, performing verification with the five-table version on the TabFact test set. We remove respectively our cross-attention function, the reranking component in the loss, and both.

higher probability of that team also losing in an unretrieved, hypothetical fourth table. To test this, we apply the model in a setting where we retrieve the top five tables *excluding* the gold table, and a setting where we use five random tables. Using highly scored (but wrong) tables, we achieve a performance of 59.4%, a significant improvement on the 53.1% we achieve using random tables. This supports our hypothesis that other *good* tables can provide useful background context for verification.

It should be noted that such inferences, while increasing model performance, may also increase the degree to which the model exhibits biases. Depending on the application, this may as such not be a desirable basis for verification. Returning to the example in Figure 1, inferring ownership on the basis of political affiliation when no other information is available may increase accuracy *on average*, but it can also lead to erroneous or biased decisions (indeed, for the claim in the example, the prediction would be wrong).

### 5.3 Ablation Tests

Our best-performing model from Table 2 relies on two innovations: The cross-attention function which contextualizes retrieved tables in relation to each other, and the joint reranking-and-verification loss. In Table 4, we evaluate the model without either of these. Leaving the attention function out is simple — we use  $f(d_q^k)$  for each table directly for predictions. We model performance without the reranking component of our loss function by assuming a uniform distribution over the tables.

As can be seen, the combination of both is strictly necessary to obtain strong performance — indeed, without our joint objective, the model performs worse than simply applying the baseline model to the top table returned by TF-IDF as in Table 2. The ability for the model to express the relative relatedness of tables to the query is crucial.

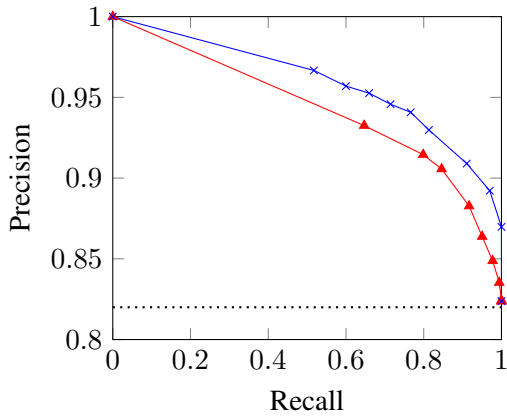


Figure 3: Precision-recall curve for determining whether a set of five retrieved tables in the TabFact validation set contains the gold table, using respectively entropy of the reranking scores with our joint loss ( $\text{---}\blacktriangle\text{---}$ ) or the maximum probability of some table being the gold table with our ternary loss, ( $\text{---}\blacktimes\text{---}$ ). We also include a most frequent class baseline ( $\text{---}\cdot\text{---}$ ).

We include further investigation of the role our cross-attention mechanism plays in Appendix E.

#### 5.4 Predicting Insufficient Information

In realistic settings, some claims will not be directly answerable from any retrieved table. In such cases, it can be valuable to explicitly inform the user — giving false verifications or refutations when sufficient information is not available is misleading, and can decrease user trust. To model a scenario where the lack of relevant information must be detected, we create a classification task wherein the model must predict for all examples, whether the gold table is among the  $k$  documents in  $D_q$ .

Using the ternary loss, our model directly gives the probability of each table containing appropriate information as  $(1 - p(I_t = \textit{irrelevant}|q, t))$ . We can estimate the suitability of the best retrieved table for verifying the claim as  $\max_t(1 - p(I_t = \textit{irrelevant}|q, t))$ , and apply a threshold  $\tau_1$  to classify  $D_q$  as suitable or unsuitable. For the joint loss, a more indirect approach is necessary. Intuitively, if our model is too uncertain about which table answers the query, there is a high likelihood that no suitable table has been retrieved. This corresponds to the entropy of the reranking component  $H_s(s|q, D_q)$  after marginalizing over the truth value of the claim exceeding some threshold  $\tau_2$ .

We compare these strategies in Figure 3, obtaining Precision-Recall curves by measuring at varying  $\tau_1$  and  $\tau_2$ . We find that while both approaches

outperform a most frequent class baseline by a significant margin, the ternary loss performs better than the joint loss. As such, the choice between the two losses represents a tradeoff between raw performance (see Tables 2 and 5) and the ability to identify missing or incomplete information.

#### 5.5 Wikipedia-scale Table Verification

In our experiments so far, we have relied on the 16,573 TabFact tables as the knowledge source. The tables selected for TabFact were taken from WikiTables (Bhagavatula et al., 2013), and filtered so as to exclude “overly complicated and huge tables” (Chen et al., 2020b). Moving beyond the scope of that dataset, a fully open fact verification system should be able to verify claims over even larger collections of tables — for example, the full set of tables available on Wikipedia. To make a preliminary exploration of that larger-scale setting, we include in Table 5 the performance of our approach evaluated using roughly 3 million tables automatically extracted from Wikipedia.

Model	Accuracy
RoBERTa only	52.1
Ours (1 table)	53.6
Ternary loss, 3 tables	55.8
Ternary loss, 5 tables	57.5
Joint loss, 3 tables	56.1
Joint loss, 5 tables	<b>58.1</b>

Table 5: Performance of our RoBERTa-based model on the TabFact test set, using all Wikipedia tables rather than just the TabFact tables as a backend.

As can be seen, our approach improves on the naive strategy of using a single table and a closed-domain verification component also in this more complex setting. To verify that the inference happens on the basis of the retrieved tables and not simply the RoBERTa-weights, we include also the performance of a model which simply uses classification on top of a RoBERTa-encoding of the claim. Similar to our previous experiments, the joint-loss model with five retrieved tables performs the strongest. We note that it is unclear whether the performance we observe here originates from correlations obtained through background information (as we see in Section 5.2 when the retriever fails to find the appropriate table), or due to verification against a single entirely appropriate table happening at a lower rate than when using TabFact.

## 6 Related Work

Semantic querying against large collections of tables has previously been studied for question answering. Sun et al. (2016) used string matching between aliases of linked entities to search millions of tables crawled from the Web, with retrieved table cells providing evidence for a question answering task. Jauhar et al. (2016) demonstrated strong results with a Lucene index and a Markov Logic Network-based model for answering scientific questions. Recently, Chakrabarti et al. (2020a,b) developed an improved model for table retrieval combining neural representations of the table and the query with a BM25 index.

Cafarella et al. (2008, 2009) employed keyphrase-based table retrieval by reranking a list of tables returned by a search engine. Pimplikar and Sarawagi (2012) used a graphical model to perform retrieval on the basis of co-occurrence statistics, table metadata, and column headers. In (Ghasemi-Gol and Szekely, 2018), non-parametric clustering was employed as a strong heuristic for table retrieval. Zhang and Balog (2018) introduced a ranking method based on mapping available features into several semantic spaces. Recently, Zhang et al. (2019) introduced a neural method for table retrieval and completion using word- and entity-embeddings of table elements.

Neural modeling of tables has been the subject of several recent papers. Aside from the original BERT-based model in (Chen et al., 2020b), the closest to our work is (Yin et al., 2020). In these paper, a pretrained BERT-based encoder for tables is introduced and demonstrated to yield strong improvements on several semantic parsing tasks. Chen et al. (2019) introduced a model to automatically predict and compare column headers for tables in order to find semantically synonymous schema attributes. Similarly, Zhang and Balog (2019) introduced an autoencoder for predicting table relatedness.

Closed-domain semantic parsing over tables has been studied extensively in the context of question answering (e.g., Pasupat and Liang (2015); Khashabi et al. (2016); Yu et al. (2018)). In Zhong et al. (2020), a logic-based fact verification system was introduced to improve on the model presented in the initial TabFact paper (Chen et al., 2020b). Yang et al. (2020) builds on the program induction model also introduced in Chen et al. (2020b), using a graph neural network to verify generated programs. Orthogonally, a similar dataset for table-

based natural language inference was introduced by Gupta et al. (2020) — interestingly, like in our experiments, they found RoBERTa-large to work extremely well for linearised tables. Finally, Herzig et al. (2020); Eisenschlos et al. (2020) introduced BERT-based models for various table semantic tasks, extending BERT with additional position embeddings denoting columns and rows.

Open-domain fact verification and question answering over unstructured, textual data has been studied in a series of recent papers. Early work resulted in several highly sophisticated full pipeline systems (Brill et al., 2002; Ferrucci et al., 2010; Sun et al., 2015). These provided inspiration for the influential DrQA model (Chen et al., 2017), which like ours relies on a TF-IDF-based heuristic retrieval model, and a complex reading model. Recent work (Karpukhin et al., 2020; Lewis et al., 2020) has built on this approach, developing learned dense retrieval models with dot-product indexing (Johnson et al., 2017), and increasingly advanced pretrained transformer-models for reading. The development of similarly fast, reliable and learnable indexing techniques for tables as well as text is an important direction for future work.

Concurrently with our work, Chen et al. (2020a) have introduced a BERT-based model to perform question answering over open collections of data including tables. Like ours, their model consists of separate retriever- and reader-steps. Their best-performing reader employs a long-range sparse attention transformer (Ainslie et al., 2020) to jointly summarize all retrieved data. As in our case, their model demonstrates significant improvements from using multiple retrieved tables.

## 7 Conclusion

We have introduced a novel model for fact verification over large collections of tables, along with two strategies for exploiting closed-domain datasets to increase performance. Our approach performs on par with the current closed-domain state of the art, with larger gains the more tables we include. When using an oracle to retrieve a reference table, our approach also represents a new closed-domain state of the art. Finally, we have made an initial foray into Wikipedia-scale open-domain table fact verification, demonstrating improvements from multiple tables also when using a full set of Wikipedia tables as the knowledge source. Our results indicate that the use of multiple tables can provide contex-



tual clues to the model even when those tables do not explicitly verify or refute the claim, because they can provide evidence for the *probability* of the claim. This is a double-edged sword, as reliance on such clues can increase performance while also inducing biased claims of truthfulness. Care will be needed in future work to disentangle the positive and negative aspects of this phenomenon.

## Acknowledgments

We would like to thank Fabio Petroni and Nicola De Cao for helpful discussions and comments.

## References

- Joshua Ainslie, Santiago Ontanon, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. 2020. [ETC: Encoding long and structured inputs in transformers](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 268–284, Online. Association for Computational Linguistics.
- Tariq Alhindi, Savvas Petridis, and Smaranda Muresan. 2018. [Where is your evidence: Improving fact-checking by justification modeling](#). In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 85–90, Brussels, Belgium. Association for Computational Linguistics.
- Chandra Sekhar Bhagavatula, Thanapon Noraset, and Doug Downey. 2013. Methods for exploring and mining tables on wikipedia. In *Proceedings of the ACM SIGKDD Workshop on Interactive Data Exploration and Analytics*, pages 18–26.
- 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.
- Eric Brill, Susan Dumais, and Michele Banko. 2002. [An analysis of the AskMSR question-answering system](#). In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 257–264. Association for Computational Linguistics.
- Michael J Cafarella, Alon Halevy, and Nodira Khoussainova. 2009. Data integration for the relational web. *Proceedings of the VLDB Endowment*, 2(1):1090–1101.
- Michael J Cafarella, Alon Halevy, Daisy Zhe Wang, Eugene Wu, and Yang Zhang. 2008. Webtables: exploring the power of tables on the web. *Proceedings of the VLDB Endowment*, 1(1):538–549.
- Kaushik Chakrabarti, Zhimin Chen, Siamak Shakeri, and Guihong Cao. 2020a. Open domain question answering using web tables. *arXiv preprint arXiv:2001.03272*.
- Kaushik Chakrabarti, Zhimin Chen, Siamak Shakeri, Guihong Cao, and Surajit Chaudhuri. 2020b. [Tableqna: Answering list intent queries with web tables](#). *arXiv preprint arXiv:2001.04828*.
- Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. [Reading Wikipedia to answer open-domain questions](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.
- Jiaoyan Chen, Ernesto Jiménez-Ruiz, Ian Horrocks, and Charles Sutton. 2019. [Colnet: Embedding the semantics of web tables for column type prediction](#). In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 29–36. AAAI Press.
- Wenhu Chen, Ming-Wei Chang, Eva Schlinger, William Wang, and William W Cohen. 2020a. Open question answering over tables and text. *arXiv preprint arXiv:2010.10439*.
- 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 *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.
- Sarah Cohen, Chengkai Li, Jun Yang, and Cong Yu. 2011. [Computational journalism: A call to arms to database researchers](#). In *CIDR 2011, Fifth Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 9-12, 2011, Online Proceedings*, pages 148–151. www.cidrdb.org.
- Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer.
- 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.
- David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J William Murdock, Eric Nyberg, John Prager, et al. 2010. Building watson: An overview of the deepqa project. *AI magazine*, 31(3):59–79.
- Terry Flew, Christina Spurgeon, Anna Daniel, and Adam Swift. 2012. The promise of computational journalism. *Journalism Practice*, 6(2):157–171.
- Majid Ghasemi-Gol and Pedro Szekely. 2018. Tabvec: Table vectors for classification of web tables. *arXiv preprint arXiv:1802.06290*.
- Vivek Gupta, Maitrey Mehta, Pegah Nokhiz, and Vivek Srikumar. 2020. [INFOTABS: Inference on tables as semi-structured data](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2309–2324, 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.
- Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. [spaCy: Industrial-strength Natural Language Processing in Python](#).
- Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*.
- Sujay Kumar Jauhar, Peter Turney, and Eduard Hovy. 2016. [Tables as semi-structured knowledge for question answering](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 474–483, Berlin, Germany. Association for Computational Linguistics.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with GPUs. *arXiv preprint arXiv:1702.08734*.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. [Dense passage retrieval for open-domain question answering](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.
- Daniel Khashabi, Tushar Khot, Ashish Sabharwal, Peter Clark, Oren Etzioni, and Dan Roth. 2016. [Question answering via integer programming over semi-structured knowledge](#). In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 1145–1152. IJCAI/AAAI Press.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. [Retrieval-augmented generation for knowledge-intensive NLP tasks](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. [Combining fact extraction and verification with neural semantic matching networks](#). In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6859–6866. AAAI Press.
- 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.
- Rakesh Pimplikar and Sunita Sarawagi. 2012. Answering table queries on the web using column keywords. *Proceedings of the VLDB Endowment*, 5(10):908–919.
- Huan Sun, Hao Ma, Xiaodong He, Wen-tau Yih, Yu Su, and Xifeng Yan. 2016. [Table cell search for question answering](#). In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 771–782. ACM.
- Huan Sun, Hao Ma, Wen-tau Yih, Chen-Tse Tsai, Jingjing Liu, and Ming-Wei Chang. 2015. [Open domain question answering via semantic enrichment](#).

- In *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, pages 1045–1055. ACM.
- James Thorne and Andreas Vlachos. 2018. [Automated fact checking: Task formulations, methods and future directions](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3346–3359, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- 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 *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.
- Andreas Vlachos and Sebastian Riedel. 2014. [Fact checking: Task definition and dataset construction](#). In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pages 18–22, Baltimore, MD, USA. 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.
- Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. [TaBERT: Pretraining for joint understanding of textual and tabular data](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8413–8426, Online. Association for Computational Linguistics.
- Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2018. [Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921, Brussels, Belgium. Association for Computational Linguistics.
- Li Zhang, Shuo Zhang, and Krisztian Balog. 2019. [Table2vec: Neural word and entity embeddings for table population and retrieval](#). In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 1029–1032. ACM.
- Shuo Zhang and Krisztian Balog. 2018. [Ad hoc table retrieval using semantic similarity](#). In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, pages 1553–1562. ACM.
- Shuo Zhang and Krisztian Balog. 2019. [Auto-completion for data cells in relational tables](#). In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pages 761–770. ACM.
- Wanjun Zhong, Duyu Tang, Zhangyin Feng, Nan Duan, Ming Zhou, Ming Gong, Linjun Shou, Daxin Jiang, Jiahai Wang, and Jian Yin. 2020. [Logical-FactChecker: 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.

## A Performance using RoBERTa-base

In our paper, we have reported results using the larger version of RoBERTa with additional hidden layers and greater dimensionality. Liu et al. (2019) also include a smaller version, RoBERTa-base, corresponding to the BERT-base model of Devlin et al. (2019). In Table 6, we report results corresponding to those of Table 2 for our joint model using RoBERTa-base instead of RoBERTa-large.

Interestingly, the performance gain from using multiple tables is even larger for RoBERTa-base (an increase of 4.1 rather than 1.9 point accuracy from using five tables, for example). One explanation could be that some information necessary to verify certain facts may be encoded in the weights of the larger RoBERTa. We attempt to investigate this using random tables; however, with five random tables, RoBERTa-large and RoBERTa-base reach an almost equivalent respective performance of 52.1 and 52.0. As such, we believe that RoBERTa-large exploits the correlations between tables which we discuss in Section 5.2 better than RoBERTa-base.

## B Hyperparameters

Our model uses RoBERTa (Liu et al., 2019) to encode each table into vectors. On top of RoBERTa we employ key-value self-attention (Vaswani et al., 2017) with two attention heads. We then use an MLP consisting of a linear transformation to  $h = 3072$  hidden units, followed by  $\tanh$ -activation and linear projection to the output space. During training, we employ dropouts with probability 0.1 before each linear transformation in the MLP. The hyperparameters for all experiments were selected using the TabFact development set (with TabFact tables as the backend).



Model	Dev	Test	Simple		
Table-BERT (Chen et al., 2020b)	66.1	65.1	79.1	58.2	68.1
LogicalFactChecker (Zhong et al., 2020)	71.8	71.7	85.4	65.1	74.3
ProgVGAT (Yang et al., 2020)	74.9	74.4	88.3	67.6	76.2
TAPAS (Eisenschlos et al., 2020)*	81.0	81.0	92.3	75.6	83.9
Ours (Roberta-base, oracle retrieval)	72.3	69.9	85.7	62.1	71.4
Ours (Roberta-large, oracle retrieval)	78.2	77.6	88.9	72.1	79.4
Ours (Roberta-base, 1 table)	64.7	64.2	78.4	57.8	67.3
Ours (Roberta-base, 3 tables)	67.8	67.0	81.5	60.5	69.4
Ours (Roberta-base, 5 tables)	68.1	68.3	83.4	61.2	70.2
Ours (Roberta-base, 10 tables)	67.5	67.1	82.3	59.9	70.1
Ours (Roberta-large, 1 tables)	74.1	73.2	86.7	67.8	76.6
Ours (Roberta-large, 3 table)	74.6	73.8	87.0	68.3	78.1
Ours (Roberta-large, 5 tables)	75.9	75.1	87.8	69.5	77.8
Ours (Roberta-large, 10 tables)	73.9	73.8	86.9	68.1	76.9

Table 6: Prediction accuracy of our RoBERTa-based model on the official splits from the TabFact dataset, using RoBERTa-base in addition to RoBERTa-large. The first section of the table contains closed-domain results, the second and third open-domain. All results use the joint objective described in Section 4.1. \* employs intermediary pretraining on additional synthetic data.

We train the model using Adam (Kingma and Ba, 2015) with a learning rate of  $5e - 6$ . We use a linear learning rate schedule, warming up over the first 30000 batches. We use a batch size of 32. Training was done on 8 NVIDIA Tesla V100 Volta GPUs (with 32GB of memory), and completed in approximately 36 hours.

### C Retrieval accuracy for TabFact splits

The TabFact dataset comes with several different data splits. We include here the performance of our retrieval component for each split:

Dataset	H@1	H@3	H@5	H@10
Train	59.5	71.2	74.8	79.2
Dev	69.6	78.8	82.3	86.6
Test	69.7	78.7	81.9	86.3
Simple Test	92.7	97.1	98.1	99.0
Complex Test	64.7	75.2	79.5	84.8
Small Test	82.1	89.6	91.4	94.7

Table 7: Retrieval accuracy with our entity-based TF-IDF heuristic on the different TabFact splits.

### D Reranking Performance

In Section 4, we introduced our model as a joint system for fact verification and evidence reranking. A benefit of our formulation is the ability to

reason about the ability of our model to rerank by marginalizing over the truth value of the claim, following Equation 7. In Table 8, we compare the table retrieval ranking performance of our joint model to a model only trained for reranking, as well as to the TF-IDF baseline.

Model	H@1	H@3	H@5
TF-IDF	69.6	78.8	82.3
Reranking only	69.9	78.9	82.3
Ours (no attention)	67.4	78.3	82.3
Ours (attention)	70.9	79.4	82.3

Table 8: Ranking performance on the TabFact validation set, using either our TF-IDF retriever alone or reranking with our model. We test a version of our model using only a reranking loss, as well as joint-loss model with and without attention.

As can be seen, our joint loss provides a slight performance improvement when the attention component is included. Interestingly, the joint-loss model performs better than a system trained purely for reranking — this highlights the complementary nature of the reranking and verification tasks.

### E The Role of Attention

An interesting question is the role attention plays in our model. As can be seen from Tables 2 and 8, our



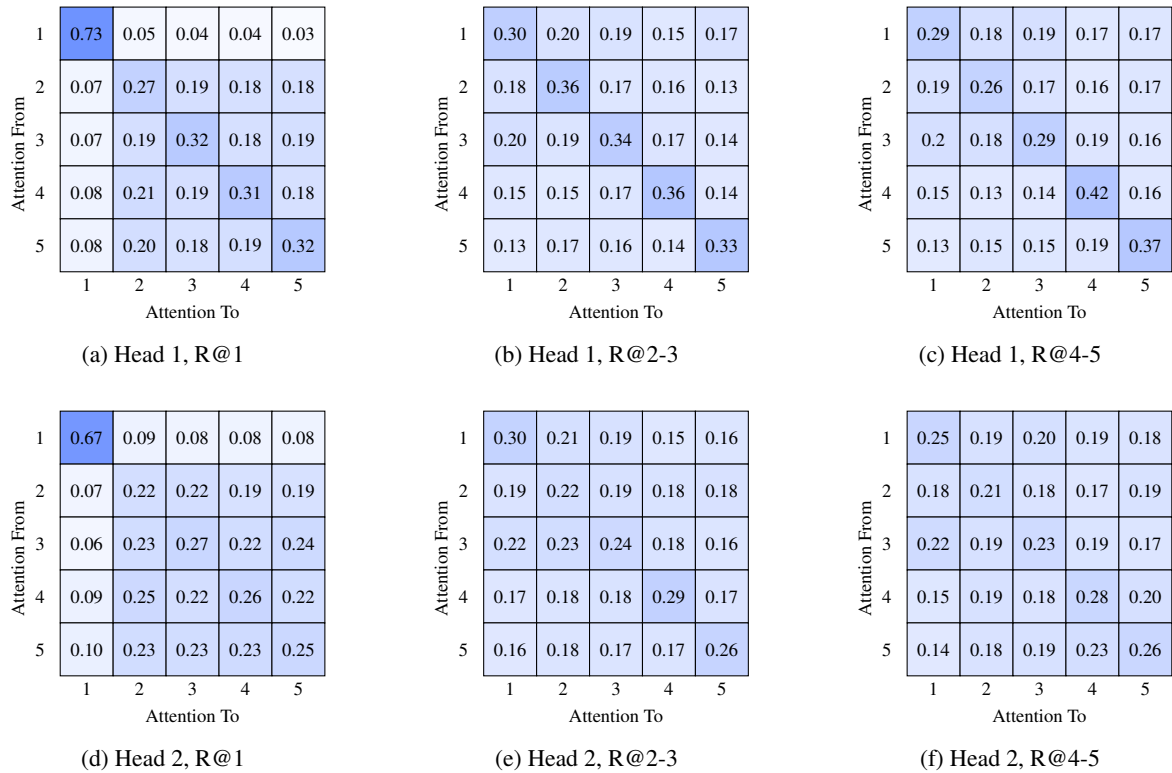


Figure 4: Confusion matrices for the cross-attention between each pair of tables for the five-table version of our model. Each head is represented separately, and individual figures are included for the parts of the dataset where our TF-IDF retriever assigns the gold table rank respectively 1, 2-3, or 4-5.

cross-attention module is necessary to achieve high performance – without it, the model struggles to identify which table should be used for verification. To investigate the function of attention, we plot in Figure 4 the strength of the cross-attention between each table for our five-table model. We produce separate plot for the two attention heads, as well as for each of the splits used in Table 3 representing the parts of the dataset where our TF-IDF retriever assigns the gold table rank respectively 1, 2-3, or 4-5.

For both attention heads, the attention function has clearly distinct behaviour when the gold table is retrieved as top 1; the degree to which that table attends to itself is much greater. We suspect that this is because of “easy” cases, where the attention function is used to separate a clearly identifiable “appropriate” table from the other tables. In harder cases, the model uses the attention focus to compare information across tables. To test this, we run the model in a setting where four random tables are used along with the gold table. In that setting, the division is even clearer. For the gold table, respectively 86 and 82 percent of the attention for the two heads is on average focused on itself; for the four

random tables, the attention is evenly distributed over all tables except the gold table.

To distinguish the two heads, we in general see the first head exhibit a pattern of behaviour where each table assigns the majority of attention to itself — especially when that table is the gold table. The second head seemingly encodes a more even spread over the retrieved tables, perhaps representing general context more than an attempt to identify the gold table.