Combining sentence and table evidence to predict veracity of factual claims using TaPaS and RoBERTa

Martin Funkquist
martin.funkquist@gmail.com

Abstract

This paper describes a method for retrieving evidence and predicting the veracity of factual claims, on the FEVEROUS dataset. The evidence consists of both sentences and table cells. The proposed method is part of the FEVER shared task. It uses similarity scores between TF-IDF vectors to retrieve the textual evidence and similarity scores between dense vectors created by fine-tuned TaPaS models for tabular evidence retrieval. The evidence is passed through a dense neural network, that is trained on the FEVEROUS dataset, to produce a veracity label. The FEVEROUS score for the proposed model is 0.126 on the test dataset.

1 Introduction

Until recently, fact checking has been done solely by journalists and other human labor. With the large spread of information on the internet, the human labor component of fact checking has become a bottleneck. Vlachos and Riedel (Vlachos and Riedel, 2014) introduced the problem of automatic fact checking. Since then a lot of progress has been made, most notably with the release of the FEVER (Thorne et al., 2018) dataset, and succeeding systems using that dataset.

The FEVEROUS shared task (Aly et al., 2021) consists of retrieving evidence and predicting the veracity of a given claim, based on the retrieved evidence. This task can be divided into two separate tasks. The first is to retrieve the most relevant evidence from the given Wikipedia dataset. Here the evidence can be any form, such as sentences, table cells or list items. The second task is to predict the veracity for the claim, given the retrieved evidence.

In this paper a method for solving this task is proposed. The method can be divided into the following parts:

- Document retrieval
- Sentence extraction
- Table extraction
- Table cell extraction
- Claim veracity prediction

Each of the parts will be explained in detail later in this paper. The document retrieval and sentence extraction models are based on TF-IDF representation and cosine distances to get the most relevant documents and sentences. The table extraction part uses the model proposed by (Herzig et al., 2021), which uses dense representations from pre-trained TaPaS (Herzig et al., 2020) models to retrieve the most relevant tables. Similarly, the table cell extraction part uses a TaPaS model to retrieve the most relevant table cells from the previously retrieved tables. Lastly the claim veracity prediction model uses TaPaS and RoBERTa (Liu et al., 2019) representations of the retrieved evidence as input to a dense neural network that outputs the predicted veracity label of the given claim.

2 Related Work

Most previous research has been done on verification of claims based on textual evidence. However, a large amount of information on the internet is not in textual form, such as tables and databases. Chen et al introduced a dataset, TabFact (Chen et al., 2019), containing textual claims paired with tables and labelled either ENTAILED or REFUTED. Two methods for determining the veracity of the claim are also proposed, Table-BERT and Latent Program Algorithm (LPA). This paper describes a *closed setting*, where the table to use is given. In a real application an *open* setting would be encountered, where the evidence needs to be retrieved first, before veracity prediction can take place. The open setting scenario was studied by Schlichtkrull et al (Schlichtkrull et al., 2020) which introduced a model for retrieving tables before using these to predict a veracity label for a claim.

The previously mentioned research has been investigating fact verification with either text or tables as evidence, but not both. UniK-QA (Oguz et al., 2020) is a question answering system that incorporates lists, tables and knowledge graphs. This is done by *flattening* the structured data into plain text using simple heuristics.

Chen et al introduced a dataset called HybridQA (Chen et al., 2020), consisting of questions paired with both tables and text that contain the answer. Thus, to get a good score on this dataset the provided model needs to be able to represent both tabular and textual data.

Pre-trained language models have turned out to be useful on downstream NLP tasks. When BERT (Devlin et al., 2018) was introduced it showed that a model can be trained on one task then be used to perform on another with good performance. A lot of other pre-trained models have been created after the introduction of BERT, one of them being RoBERTa (Liu et al., 2019). RoBERTa has the same architecture as BERT but is trained in a different way and on more data, which yields a better performance.

Most of the language models are pre-trained on textual data, but the FEVEROUS dataset requires reasoning over tabular data for some of the samples. TaPaS (Herzig et al., 2020) is one language model that is pre-trained on tables, instead of plain text. The input to the model includes tokens that represent the structure of the tables, such as row and column number.

3 Model

In this section the different parts of the model is described in detail. A graphical overview of the model is shown in Figure 1.

3.1 Document retrieval

For document retrieval, matching of TF-IDF vectors is used to efficiently compare a claim with all the documents in the dataset. Before creating the TF-IDF matrix, the words are stemmed to reduce the size of the matrix. To reduce the matrix even further the top 10% most frequent words are removed and all words that appeared less than 2 times were also removed. This results in a matrix of shape (N_{docs}, N_{tokens}), where $N_{docs} = 5421406$ and $N_{tokens} = 2634922$.

Preprocessing data To create the corpus only the sentence data for each document are used. As

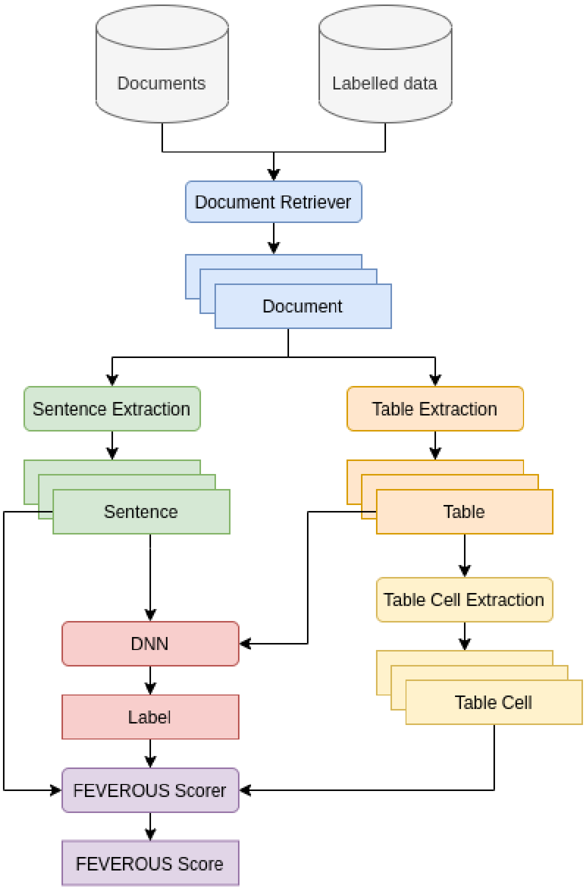


Figure 1: The design of the full model

there is more data than this, such as the tables and lists, the complete documents are not used in the retrieval step. However, the retrieval accuracy was considered sufficient, using only this reduced dataset.

Incorporating document title To improve the retrieval accuracy, a separate TF-IDF matrix is created using only the titles of the documents as the corpus. Since the title corpus is much smaller than the text corpus, it was possible to use n-grams in order to improve retrieval accuracy, with only a negligible increase in the size of the TF-IDF matrix. In this model bigrams are used for constructing the TF-IDF matrix.

3.2 Sentence extraction

For sentence extraction, a similar model as the one for document retrieval is used. Instead of dividing on distinct documents, each sentence in the top documents retrieved from the previous step is used. Then these sentences are matched with the claim using cosine similarity of the TF-IDF weighted vectors. Unigrams, bigrams and trigrams are used in order to create the TF-IDF matrix. The top k sentences are then selected as the evidence.

3.3 Table extraction

In the table extraction part, to get a more semantic representation of the tables, TaPaS (Herzig et al., 2020) is used. More specifically, the implementation of the dense table retriever described in (Herzig et al., 2021) is used[1]. Here the query and table representations are created from two separate TaPaS models. The output hidden representation of the [CLS] token is transformed through individual matrix projections into vectors, one for the query and one for the table, respectively. The matching score between a query and table is calculated by the dot product of the output vectors. Then the top k tables are returned. The dense table retriever is fine-tuned on the FEVEROUS dataset.

3.4 Table cell extraction

The model for extracting table cells is a fine-tuned TaPaS model. More specifically the model used is the TapasForQuestionAnswering from the Huggingface library[2], which is based on the model described in (Herzig et al., 2020). This model is first fine-tuned on the FEVEROUS dataset, then the fine-tuned model is used to extract the most relevant cells. Since the model only takes one table as input, each table is handled separately. A cell classification threshold is set to 0.1, to only retrieve the most relevant cells from each table. This means that there could be a varying amount of cells retrieved from each table. Because of this only the top m cells are kept, if the retriever would retrieve more than m cells. That leads to a maximum of $k \times m$ table cells for each claim, where k is the number of extracted tables.

3.5 Claim verification

In the last part of the model, a dense neural network is trained on the FEVEROUS dataset. This DNN consists of three dense layers, the first two with ReLU activations and the last with a Softmax activation for predicting the veracity of the given claim. The claim and top table are embedded using a pretrained TaPaS model and the sentence evidence is concatenated then embedded using a pretrained RoBERTa model. The TaPaS output are flattened before being concatenated with the RoBERTa output. The concatenated vector is then used as the input for the DNN. Note that the input to the TaPaS model is the *whole* highest scored table from the table extraction phase described in 3.3, and not the individual table cells extracted in section 3.4

One drawback with this model is that it only uses one table. The retrieved evidence may consists of more than one table, and it is not certain that this one table contains all the required information to give a valid prediction. Improvements to this model is discussed in Section 5.4.

The DNN is shown in Figure 2.

4 Results

4.1 Document Retrieval

The incorporation of titles using TF-IDF improved the accuracy of the document retrieval from 56.16% to 81.44%. However, note that when both title and text TF-IDF were used, the model returned a maximum of 10 documents (documents that is retrieved by both the text and title TF-IDF is only

[1] https://github.com/google-research/tapas/blob/master/DENSE_TABLE_RETRIEVER.md

[2] https://huggingface.co/transformers/model_doc/tapas.html#tapasforquestionanswering

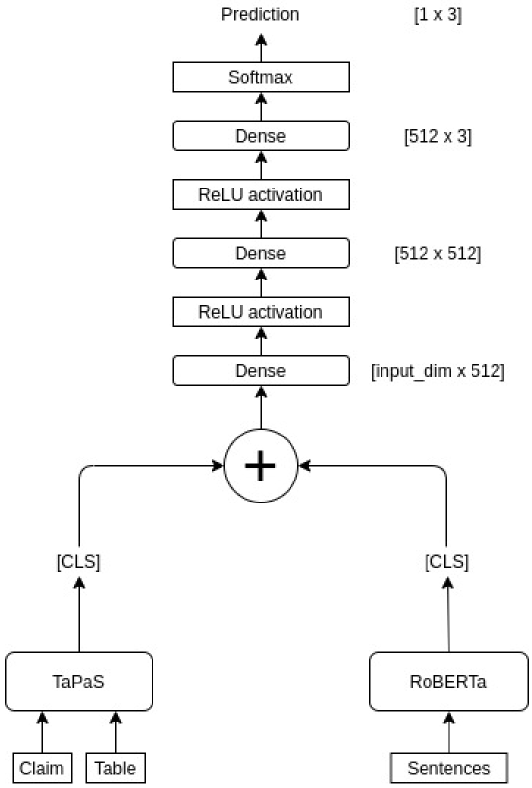


Figure 2: The design of the DNN for predicting the veracity of a claim

counted once), but in the case of only the text TF-IDF, a set of 5 documents were retrieved.

Comparing these results to the baseline document retrieval component of the FEVER system (Thorne et al., 2018) which is 77.24% for the top 10 retrieved documents, this seems like a sufficient score for document retrieval part of the system.

4.2 Sentence extraction

When retrieving the top k sentences, the result of the sentence extraction is as follows, for $k = 5$:

- Precision: 24.17
- Recall: 64.59
- F1: 35.18

Precision and recall is only calculated on the claims that have associated sentence evidence. If the examples that only contain table evidence were included, the precision would be lower.

4.3 Table extraction

The accuracy of the table extraction is measured with precision@k, and is shown in Table 1. As the results for the top 5 tables are good, this value is used in the end to end system.

Train		Dev	
k	**precision**	**k**	**precision**
1	0.59	1	0.75
5	0.81	5	0.91
10	0.87	10	0.95
15	0.90	15	0.96
50	0.96	50	0.99
100	0.97	100	0.995

Table 1: Results for the table retrieval using TaPaS on the train and dev set

4.4 Table cell extraction

The result for the table cell extraction is shown in Table 2. The accuracy of the cell extraction is limited by the table retrieval accuracy. Therefore, having higher accuracy than the retrieval part of the model is not possible.

The precision is low, which may be explained by the model extracting 5 cells from each of the 5 extracted tables, resulting in a total of 25 cells for each claim, and the evidence for most claims usually consists of just a few table cells.

	Tables only	All
Precision	04.95	10.09
Recall	29.89	61.95
F1	08.50	17.35

Table 2: Scores on the dev dataset for the table cell extraction, when considering only examples with table evidence and all examples

4.5 Claim verification

The accuracy of the claim verification part is calculated as the total amount of correct predictions divided by the total number of examples. The oracle results (e.g. the model is given the correct evidence) on the dev set are 0.68.

As shown in Table 3, the model only predicted SUPPORTS and REFUTES labels. Some reasons for this is discussed later in this paper.

The confusion matrix in Figure 3 shows that the model has some sense of which claims are supported and which ones are refuted.

4.6 FEVEROUS score

The final result is the FEVEROUS score, which measures the overall performance of the model. This is the only score available for the unlabelled test set. The score for this test set is presented in

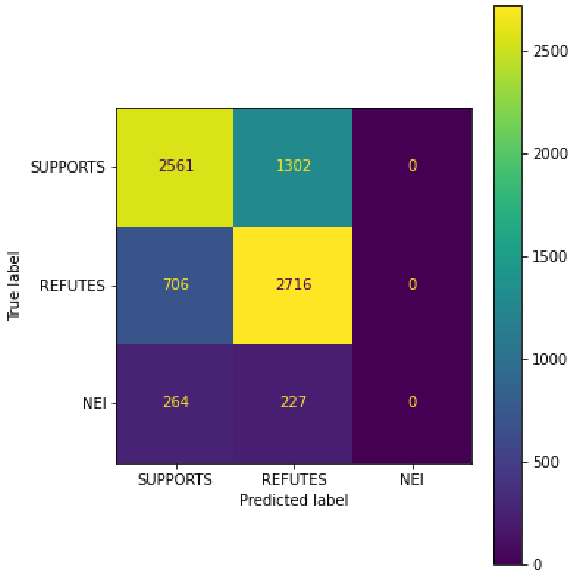


Figure 3: Confusion matrix of the predicted labels

Label	Predicted	Actual
SUPPORTS	3533	3869
REFUTES	4243	3417
NOT ENOUGH INFO	0	490

Table 3: The true labels compared to what the model predicted

Table 4, along with the score for the baseline model (Aly et al., 2021). As seen in the table, the model presented in this paper did not beat the baseline model.

	Baseline	Test
Feverous score	0.19	0.13
Label accuracy	0.53	0.43
Evidence precision	0.12	0.06
Evidence recall	0.29	0.28
Evidence F1	0.17	0.10

Table 4: Feverous score result on the unlabelled test set and baseline model.

The results presented previously has been on the train and dev datasets, and has been presented for each individual component of the model. The FEVEROUS score is the first score that measures the accuracy of the whole model, and might be the reason for the large drop in accuracy, since the errors propagate through the model.

5 Discussion

One improvement to the algorithm could be to incorporate the table caption into the table representation. Currently that information is discarded, but for some tables it could be crucial to know that to make a valid interpretation of the table. By studying the dataset, one could find that some required evidence are the table captions.

Another extension would be to add the page title or surrounding context into the representation of the table. Tables are rarely presented without some context around them, and for the algorithm to give a valid prediction, it might need to know the surrounding context, the same way as human beings would.

5.1 Table size restrictions

The tables that are too large to fit the TaPaS model are filtered out. The cap is set at $n_{rows} = 64$, $n_{cols} = 32$ and $n_{cells} = 512$. This means that evidence inside larger tables will never be retrieved with this model. If the size limitation problem would be solved, the accuracy of the model might increase.

However, with the given size limits only 424 tables that are needed for verification is filtered out. This results in 0.77% of the total amount of required tables, on the train and dev set. Therefore, increasing the maximum possible size of the tables will not likely lead to a significantly better score, unless the test set has a much larger portion of large tables.

5.2 Limit on the amount of retrieved cells

For the table retriever component, each table was limited to retrieve the top 5 cells. This means that if more than 5 cells in a single table were part of the evidence, these cells would never be retrieved. Here an improvement could be to analyze the probabilities for all table cells simultaneously, in order to select the most relevant cells from all tables, instead of only the most relevant cells from each table separately, as done in this model.

There are 12904 claims in the train and dev datasets that need more than 5 table cells as evidence from at least one table. This is 16.3% of the total amount of claims, which is a significant amount. Thus, removing the limit of 5 cells per table might contribute to an increased score.

5.3 Improvements on the veracity prediction model

As shown in the results for the veracity prediction model, the model did not predict any NOT ENOUGH INFO (NEI) labels. One reason for this could be that the dataset is not balanced and contains significantly more data samples for the other labels. This could be solved by sampling NEI data points as done in the FEVEROUS baseline (Aly et al., 2021). Without the sampling of NEI samples the baseline model does not predict any NEI labels correctly.

The test set consists of 19% NEI samples, while the train set has 3% and the dev set has 6%. Since the veracity prediction model did not predict any NEI labels for any of the datasets, this hurts the accuracy on the test set more than the accuracy on the other datasets.

Another reason why the model does not predict any NEI labels could be because the evidence retrieved for the NEI samples may not be as relevant as the evidence for the other labels. Therefore, the model did not find a good correlation between the NEI evidences.

The confusion matrix in Figure 3 shows that the model predicts the SUPPORTS and REFUTES labels for the NEI samples about an equal amount of times. There is no clear sign that it has a bias towards predicting one or the other. This might mean that the model has no idea of how to treat NEI samples.

5.4 Extending the veracity prediction network input

As previously mentioned, one drawback with the veracity prediction network is that it only takes one table as input. This way it discards the other tables that have been retrieved, because all tables except the top one, are not used for predicting the veracity. Thus, a possible improvement to the method would be to take all the retrieved tables into consideration.

The train and dev datasets have a total of 6624 samples which have multiple tables as evidence. With a total of 79181 samples this means that 8.37% of the samples needs multiple tables to determine the correct label of the claim. This is a significant portion of the samples and therefore incorporating multiple tables into the label prediction model could potentially yield an increased score.

The current model has a large input representation, of the size $l_R \times d_R + l_T \times d_T$, where $l_R = 256$ is the RoBERTa sequence length, $d_R = 768$ is the RoBERTa hidden dimension, $l_T = 512$ is the TaPaS sequence length and $d_T = 128$ is the TaPaS hidden dimension. This adds up to a total input dimension of 262144, which is fairly large. The main reason that only one table was used in the claim verification model is because of this large input dimension. Using all of the k extracted tables by concatenating the vector representations would yield a larger input dimension. For $k = 5$ the input dimensions of the DNN would increase to 524288 as the TaPaS sequence length would be $l_T = 2560$.

In the proposed input representation each word in the sentences and each cell in the table have their own representation. One reduction in the dimensionality could be to represent each sentence as a vector, instead of each individual word. A similar strategy could be used for the tables. A set of cells could have a common vector representation instead of each individual cell having its own representation, as in the current state of the model.

5.5 Future work

One obvious task to investigate would be to include several tables in the prediction part, as described in the previous section. This could be done by creating a model to represent multiple tables, or extending the input layer of the proposed prediction model.

Another interesting topic would be to investigate the possibility of training a BERT based Transformer model that has the ability to take both tables and sentences as inputs, and output the veracity as the hidden representation of the [CLS] token, and at the same time provide the most relevant table cells. This could even be extended to input a set of say 5 documents containing both text and tables, and the model would predict the aforementioned elements. But it seems infeasible due to the large input representations that would be needed to represent the full documents.

6 Conclusion

This article describes a method for solving the FEVEROUS shared task introduced by (Aly et al., 2021). The system consists of two main parts, the retrieval part and the verification part. In the retrieval part, documents are first retrieved, then sentences and table cells are extracted from these documents. For the verification part, the previously retrieved evidence, in form of sentences and table cells, are used to make a prediction about the veracity of a given claim.

While the results show that this method may not be feasible in a real world application, there are parts that show promising results. The problem of automatic fact checking is difficult and includes several NLP topics, such as information extraction and natural language inference. Certainly there will be further advances in this topic, which may result in a system that is usable in real world applications.

Acknowledgements

Thanks to Rami Aly, Christos Christodoulopoulos, Oana Cocarascu, Zhijiang Guo, Arpit Mittal, Michael Schlichtkrull, James Thorne and Andreas Vlachos for organizing the FEVER 2021 workshop.

References

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. FEVEROUS: fact extraction and verification over unstructured and structured information. *CoRR*, abs/2106.05707.

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

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020. HybridQA: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1026–1036, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Jonathan Herzig, Thomas Müller, Syrine Krichene, and Julian Martin Eisenschlos. 2021. Open domain question answering over tables via dense retrieval. *CoRR*, abs/2103.12011.

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.

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. *CoRR*, abs/1907.11692.

Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Sejr Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Scott Yih. 2020. Unified open-domain question answering with structured and unstructured knowledge. *CoRR*, abs/2012.14610.

Michael Sejr Schlichtkrull, Vladimir Karpukhin, Barlas Oguz, Mike Lewis, Wen-tau Yih, and Sebastian Riedel. 2020. Joint verification and reranking for open fact checking over tables. *CoRR*, abs/2012.15115.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and verification. *CoRR*, abs/1803.05355.

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.

A Dataset statistics

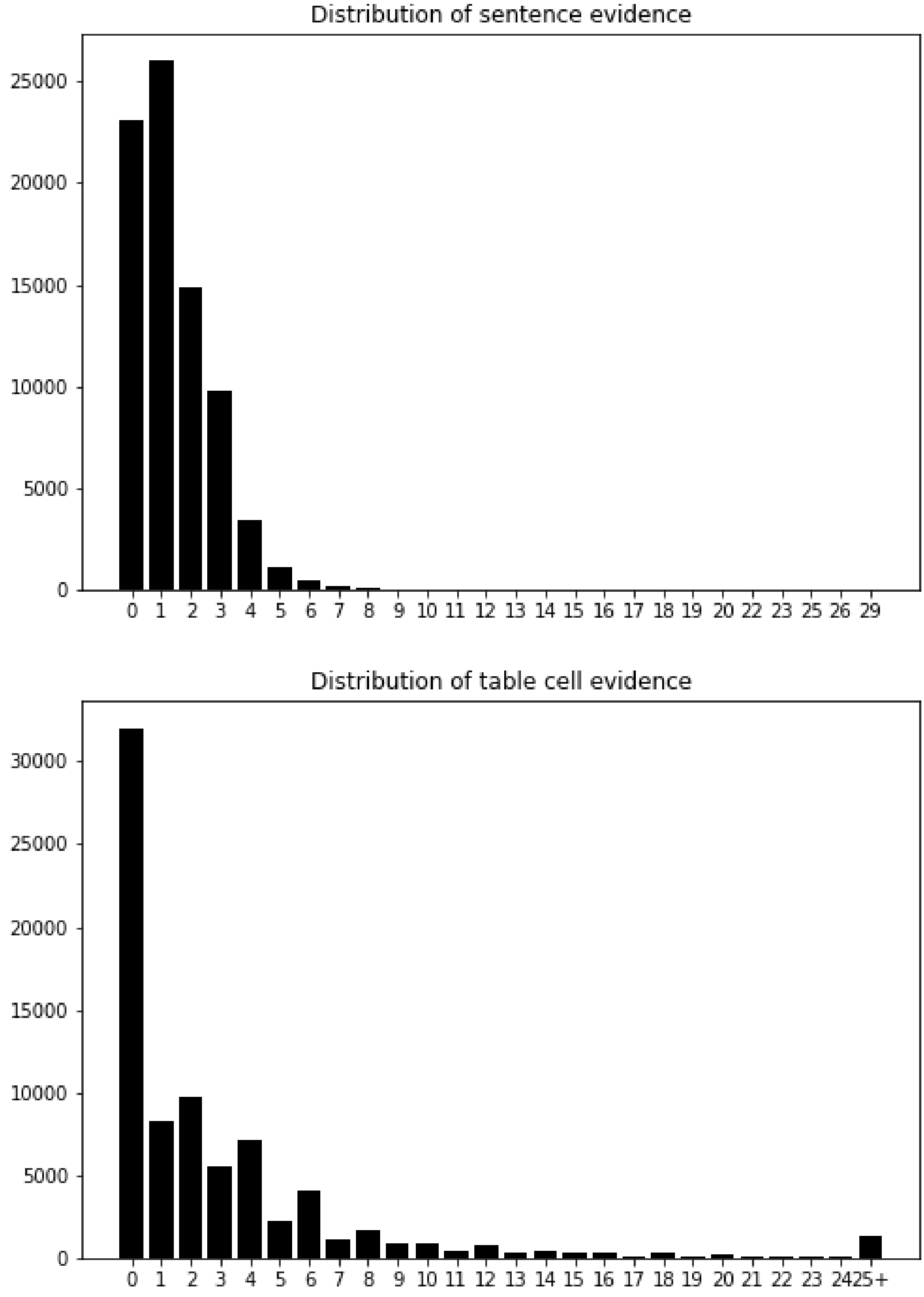
It can be valuable to take a look at the statistics of the dataset in order to examine where the model could do wrong. Table 5 shows some statistics for the FEVEROUS dataset. As can be seen in the label statistic, there are many more samples with labels SUPPORTS and REFUTES than NOT ENOUGH INFO. This may be the reason for the model not predicting any NOT ENOUGH INFO labels. It may not have learned what separates these labels from the others, due to the lack of examples, or it could simply assume that the distribution of labels will be the same always and be satisfied with only optimizing on predicting the other labels. Thus, one improvement in the training phase could be to sample NOT ENOUGH INFO examples for the Wikipedia data, as done in the FEVEROUS baseline system.

Worth noting in Table 5 is that a total of 4921 (about 6%) of the claims have evidence that is neither a sentence nor a table cell. The model proposed in this paper will never return any of these evidences. An extension would then be to include this type of evidence in a future iteration of the model.

The distribution of the different types of evidence is shown in Figure 4. There are very few samples that have more than 5 sentences as evidence, which means that not many examples will be missed due to a known limitation in the model. The model currently only returns a maximum of 25 table cells, and there are a few examples that have more cells than this as the evidence. However, this is comparatively small compared to the total number of examples, which means that the majority of errors are not due to this limitation of the system, given that the distribution of data samples is similar for the test set.

	Total	**Train**	**Dev**
Samples	79181	71291	7890
w. single evidence	72945	65764	7181
w. multiple evidence	6236	5527	709
w. sentence evidence	56139	50484	5655
w. table evidence	47174	42611	4563
w. other evidence	4921	4423	498
Labels			
SUPPORTS	45743	41835	3908
REFUTES	30696	27215	3481
NOT ENOUGH INFO	2742	2241	501

Table 5: Feverous score result on the unlabelled test set.



Distribution of sentence evidence

Distribution of table cell evidence

Figure 4: The distribution of claims with the given sentence evidence and table cell evidence respectively