FaBULOUS: Fact-checking Based on Understanding of Language Over Unstructured and Structured information

Mostafa Bouziane[*], **Hugo Perrin**[*], **Amine Sadeq**[*] and **Thanh Nguyen**[*]
Aurélien Cluzeau and **Julien Mardas**
Buster.Ai
contact@buster.ai

Abstract

As part of the FEVEROUS shared task, we developed a robust and finely tuned architecture to handle the joint retrieval and entailment on text data as well as structured data like tables. We proposed two training schemes to tackle the hurdles inherent to multi-hop multi-modal datasets. The first one allows having a robust retrieval of full evidence sets, while the second one enables entailment to take full advantage of noisy evidence inputs. In addition, our work has revealed important insights and potential avenue of research for future improvement on this kind of dataset. In preliminary evaluation on the FEVEROUS shared task test set, our system achieves 0.271 FEVEROUS score, with 0.4258 evidence recall and 0.5607 entailment accuracy.

1 Introduction

In the past year, concerns about the spread of misinformation have risen dramatically. Around 54% of people claim that they have seen fake news on COVID-19 in a week (Reuters Institute for the Study of Journalism) resulting in a global lack of trust in our political institutions and the media. Considering the amount of data there is on the Internet, the need for a trustworthy, efficient, automated fact-checking tool is undeniable.

Research has made great stride in automating fact verification, driven for instance by the successive FEVER datasets and challenges (Thorne et al., 2018a) (Thorne et al., 2018b), for which new techniques have been developed for both retrieval (Xiong et al., 2021) and entailment (Liu et al., 2019b). The retrieval part has been handled through hard sample mining and an adjusted embedding based model training. This allows very fast inference, leveraging approximate nearest neighbors searches on large dimension embeddings. Given the dataset size, this is necessary to achieve fast enough running times. On the entailment end, one of the state-of-the-art approaches is called KGAT (Liu et al., 2019b). This method leverages graph structure over the evidences to allow for both more accurate entailment as well as provide interpretability to the result. However, this method was designed for text, and is therefore not suitable for out of the box handling of structured data. This is where our interest can go towards methods like TAPAS (Eisenschlos et al., 2020), which is suitable for structured data. It was developed for Tabfact (Chen et al., 2019), on which it exhibits excellent performance.

However, while the previous FEVER challenges (Thorne et al., 2018a) (Thorne et al., 2018b), and other well-known reference datasets like HotpotQA (Yang et al., 2018) only focused on text, the FEVEROUS dataset introduced multi-modality through added structured data. This brings the dataset closer to real-world data and the task closer to fact verification in a practical setting. Many difficulties inherently arise from the diverse nature of the data, and we want to shed some light on both the data and how to reason on it. Our method achieved the top result on the FEVEROUS shared task and gave us insights on how to progress further by exploring the nature of the data and where our pipeline fails.

We also developed the Reinforced Adaptive Retrieval Embedding (RARE) and Noisy Entailment through Adapted Training (NEAT) methods to address certain hurdles inside the training itself.

In this paper, we aim to share both the insights we gained on the FEVEROUS dataset (Aly et al., 2021) during the workshop competition, as well as how we designed the top scoring pipeline of the competition. We will first present the dataset, what we learned from it, as well as the issues we encountered while designing an algorithm for it and how we devised our approach. In the second section, we will explain how our pipeline works and the tweaks

[*] Equal Contribution

Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER) at EMNLP 2021, pages 31–39

we made to the classical pipeline defined by the baseline method (Aly et al., 2021). Following, we will present the unique training methods that were used to ensure both stable predictions as well as state-of-the-art performance in both retrieval and entailment.

In short, our contributions with this paper are the following:

- Top scoring pipeline on the FEVEROUS shared task
- New training paradigms for fact verification for both retrieval (RARE) and entailment (NEAT) methods
- Insights on why previous state-of-the-art approaches fails on this kind of data, how we fixed part of the issues and the proposed paradigm shift proposal for further research.

2 Dataset Insights

The dataset has been designed to introduce multi-modality (Aly et al., 2021). This has been done by adding structured data, i.e. text data which has an underlying structure to support it, like tables and lists. As we say one of these clues is not like the others, and we will explain what make cells so unlike sentences, as well the particularities involving the label annotation philosophy.

The difficulty with those is that their entries are not self-contained. Indeed, the value of a cell in a table depends on other values of the table. For instance, in Table 1, while the values of certain cells might be the same, they differ in terms of context, as one cell value of 0.5 is a number of carrots per day in one case and a number of children per year in the other case.

In order to understand the value of a given cell, you might need to access values of headers and other values of the same row/column. This makes the retrieval particularly hard as it might miss the value or the header which are needed for the full set to be retrieved. In addition, linearising the table might not be enough, as the header to retrieve might be on a different row altogether. This could also cause oversampling of a certain row or column. This issue of context can be solved more easily for entailment as you can keep the extra information present of all headers available. Indeed, we can reconstruct the full table from the retrieved cells for entailment, as long as the model for entailment allows dealing with the full table size. As it was the case with the TAPAS (Eisenschlos et al., 2020) method we used for entailment, one important part might be to find the right position embedding space to encode that structure.

Another peculiar aspect of the data that proved hard to tackle was the handling of refuting evidence sets. We have found that often refuting relies on small details which have been changed. The issue with this is that training for retrieval will most of the time converge towards trying to get as much information out of the selected evidence as possible. It will therefore not rank the small information overlap of the evidence refuting as high as needed for the evidence retrieval and as a consequence not get the full accurate result. An iterative approach to build up supporting, refuting and non relevant evidence sets might prove successful to address this. As described in Figure 1, we would build a set of supporting evidence up until we verify everything, find a refuting evidence or reach a termination heuristic, and then re-rank evidences properly based on their entailment usefulness. This paradigm would fit the way the dataset has been annotated. The main drawback to this is the need to entangle the entailment and retrieval in one model that may not scale with data as easily as the embedding-based retrieval models that rely on cosine similarity.

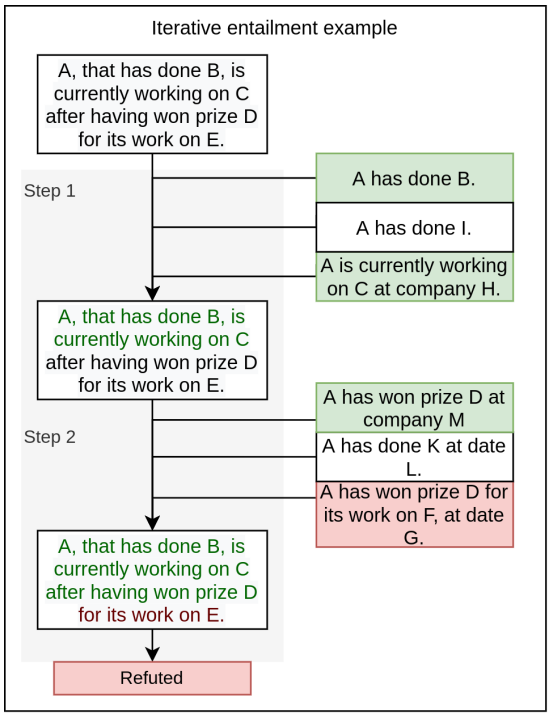
Same goes for Not Enough Information (NEI) examples, where the piece of information missing might be very small. Hence making the difference between "support" and "NEI" is hard for classical approaches. An example of this can be found in Appendix A where the claim is not completely covered by supporting evidences and therefore does not count as supported. In the setup of Figure 1, NEI labels would appear naturally from not verifying the full claim.

Finally, it is noteworthy to point out that the document retrieval is relying heavily on name entities, which makes BM25 quite good at it. An easy way to improve it would be to disambiguate the nicknames, abbreviations and the likes to make it more able to find certain pages.

3 Our Pipeline

As depicted in Figure 2, for the full pipeline, we first perform document retrieval, then passage retrieval and finally entailment. On a single GPU machine described in appendix B, inference takes

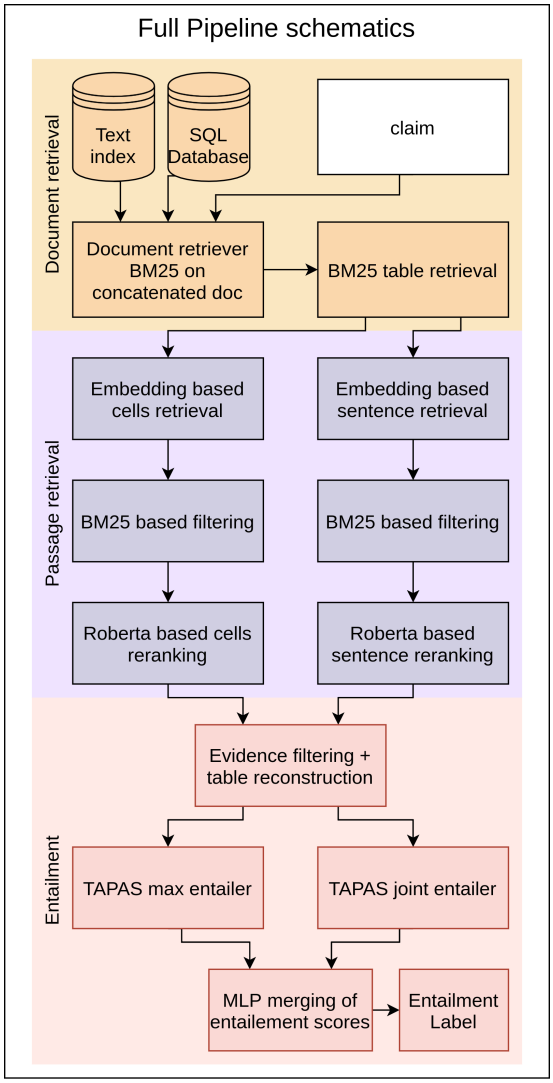
Figure 1: Example of an iterative retrieval and entailment process. Ideally we would take the candidates from retrieval, eliminate the non-supporting passages, and retrieve again, à la Multi-hop Dense Retriever (Xiong et al., 2021).



Organisation	Carrots/day	Children/year
Hogwarts	0.5	0.98
Phoenix Order	0.98	0.5

Table 1: Table example. It displays an example of table that might be hard to make sense of without columns and rows contexts.

Figure 2: Full pipeline diagram of our solution to the FEVEROUS Shared Task. The mentioned SQL database is the one available on the FEVEROUS dataset page



less than 10 seconds per sample, which translates to under 20 hours for the full dev and test sets.

For each claim to verify, we first retrieve documents using BM25 and filter the tables of those documents to only keep the most relevant ones according to BM25. From these documents we can extract sentence and cell passages, on which, in turn, we run our passage retriever to get the most relevant passages. Using input normalisation techniques, we feed this to the ensemble entailer model which predicts the entailment class. This full pipeline achieves a significant improvement of 53% improvement over the FEVEROUS baseline (Aly et al., 2021). We will detail in the next subsections each sub-part of the pipeline with its respective performance.

3.1 Document retrieval

For document retrieval, we use BM25 by indexing the documents. We tested indexing with (i) the page title only, (ii) the page and the beginning of the document and (iii) indexing the page title concatenated with all the passages of the document, and the latter performed best with Mean Average Precision (MAP) at depth (here the number of top retrieved documents) 3 of 81.8% (Table 2). Here the MAP is computed for full sets of evidence present rather than just one piece of evidence. For just one evidence, the MAP@3 is around 92%. We intentionally decided to stop at three documents retrieved, as taking more documents was counterproductive for the passage selection for which dropped significantly when using five or more pages instead of

three.

Additionally, we also filter the tables by taking the seven best tables according to BM25 as well. Here we take all tables present in the retrieved documents, and we index the linearised and concatenated tables as strings for BM25. Thereby, eliminate a lot of potential cell candidates and improved cell MAP of cells by 10% afterwards. Grid search showed us that seven tables proved to be the ideal number in our pipeline, as we achieved around 92% MAP@7 for the retrieved tables (see Table 2).

3.2 Passage retrieval

After the documents have been retrieved, we derive relevant passages therein using the following pipeline: the embedding-based retrieval, the BM25 re-filtering, and finally a re-ranking step. We apply the following processes on cells and sentences independently.

Embedding-based retrieval

One main challenge we face when we retrieve the top k documents is scaling to large documents. A simple approach to get the top passages consists of using BM25 here also. However, the drawback of this approach would be that we lose the multi-hop aspect of the dataset. Also, we want to catch the passages with semantics similar to the query, while BM25 would only find term matches. We propose training a Multi-hop Dense Retriever (MDR) (Xiong et al., 2021).

The page passages contain in some cases, words in different languages. To tackle this, we decided to use a language-agnostic embedding model LaBSE (Feng et al., 2020) that has proven to be efficient in this kind of tasks. After training the model, the passages from the documents retrieved at last step are scored with cosine similarity in an iterative manner as suggested in the MDR paper and then ranked by their relevance. We then keep only the best passages with top k value adjusted based on the passage type.

BM25 re-filtering

While analysing the results of our embedding-based retrieval model, we found out that it is not robust to the presence of named entities in passages. In this case, it struggles to put the right pieces of evidence on top, as it tends to choose the ones that are close in terms of semantics and context. To tackle this problem, we run BM25 on top of the retrieved passages. That way, we keep the relevant semantics, and only push to the top those that actually contain the right named entities. Using this strategy, we succeeded to achieve improvements of around 11% on MAP@50 on sentences, and 22% on MAP@200 on cells. This effectively re-ranks the filtered passages from the embedding-based retrieval. We provide an example where it helped in Appendix C.

Re-ranking

Finally, we re-rank the remaining passages and only keep the top five sentences and top 25 cells. In order to do this, we trained a RoBERTa (Liu et al., 2019a) model with the same training data as the passage retrieval embedding model. We will give more details in Section 4. The model is trained to output a score between 0 and 1, given a pair input consisting of the query and the passage claim we want to re-rank, with 0 being non-matching and 1 being matching. This allowed a significantly more robust MAP on the train and dev set, even though it only gives a minor boost in performance for the MAP after re-ranking evaluation. Indeed, we barely gain anything in terms of cell ranking, and around 1% for sentence ranking but that ends up giving an almost absolute 3% boost to the combined MAP (for which we evaluate if the full set containing the cells and sentences is retrieved), which is around 7.5% relative improvement. The re-ranking is, however, not being trained in a multi-hop manner, which might explain why it does not show a more significant performance increase. We decided against using a multi-hop approach for computational budgeting reasons, as it would have required us to evaluate the network on $\frac{n!}{(n-k)!}$ pairs instead of just n, with n the number of pieces of evidence retrieved at the last step.

Step	MAP @ depth	depth
Document retrieval	0.818	3
Table retrieval	0.92	7
Sentence retrieval	0.715	50
Sentence retrieval	0.543	5
Cell retrieval	0.555	200
Cell retrieval	0.341	25
Total retrieved	0.376	30
Sentence re-ranked	0.566	5
Cell re-ranked	0.342	25
Total re-ranked	0.404	30

Table 2: Table of retrieval results on the dev set.

3.3 Entailment

For entailment on retrieved pieces of evidence, we propose using an Multi-Layer Perceptron (MLP) ensemble to combine two models: TAPAS Max and TAPAS Joint. The name TAPAS Max is derived from the use of the max pooling layer described in Section 3.3.1, whereas TAPAS Joint is derived from the way we join the two modalities, as explained in Section 3.3.2. Both models use a TAPAS (Herzig et al., 2020; Eisenschlos et al., 2020) backbone pre-trained on TabFact (Chen et al., 2019). While the two models share the same backbone architecture, the ways they process sentences and cells are significantly different. The following subsections explain how each model encodes the pieces of evidence and then aggregates the information.

3.3.1 TAPAS Max

Given a claim, retrieved evidences include sentences, cells and headers. TAPAS Max preprocesses the sentences as tables with the document title and the sentence. For cells and headers, TAPAS Max recreates the table from the Wiki database and crops them to only the retrieved rows and columns. Table evidences are afterwards encoded in parallel via the TAPAS backbone. Since the number of pieces of table evidence is not known a priori for each claim, TAPAS Max aggregates the encoded tables in latent space as a fixed-size feature vector using a Max Pooling Layer. Finally, a classification head assigns the claim label "SUPPORTS", "REFUTES" or "NEI" based on the aggregated feature vector.

3.3.2 TAPAS Joint

The idea of TAPAS Joint is to aggregate all pieces of evidence in a single joint table in advance. For sentences, TAPAS Joint also considers each sentence as a table of document title and the sentence itself. For cells and headers, TAPAS Joint groups pieces of evidence by the table and truncates them to the right rows and columns. Afterwards, table evidences are outer-joined as a single joint table. Experiments have shown that table joining is order-sensitive, i.e., sentence, tables... are preferred to stand before other tables. Once having a single joint table, for each claim, encoding and classification parts are similar to the TAPAS Max without the Max Pooling aggregation.

3.4 Entailment Ensemble Model

As stated above, after training, we ensemble the scores of the TAPAS Max and the TAPAS Joint. We experimented with two strategies:

- Max confidence: we choose the label corresponding to the maximum score.
- MLP ensemble: we train a MLP model on the dev set to aggregate the scores.

The tables 3, 4 below show the performance of each model alone on dev 3 and test 4 sets, and the performance of the ensembled ones on the same datasets. We reach the best performance on label accuracy and FEVEROUS score using the MLP (hidden layer size of 32, and a ReLU activation function) ensemble.

Model	Accuracy	FEVEROUS score
TAPAS Max	56.23 %	0.2461
TAPAS Joint	58.02 %	0.2560
Max confidence	58.82 %	0.2574
MLP ensemble	65%	0.3046

Table 3: Entailment models performance on dev set: label accuracy & FEVEROUS score (Accuracy is computed only on the claims for which we find the right evidences with the same retrieval output).

Model	Accuracy	FEVEROUS score
TAPAS Joint	53.24 %	0.2267
Max confidence	55.04 %	0.2344
MLP ensemble	63.64 %	0.2710

Table 4: Entailment models performance on test set: label accuracy & FEVEROUS score (Accuracy is computed only on the claims for which we find the right evidences with the same retrieval output)

4 Training methodology

In this section, we will give additional detail on our training methodology and how it helped us achieve state-of-the-art performance on the FEVEROUS dataset.

4.1 Passage retrieval

For passage retrieval, we propose a novel approach we call Reinforced Adaptive Retrieval Embedding paradigm (RARE). Building upon the training scheme of MDR (Xiong et al., 2021), we not only use BM25 to retrieve $m_i \in \mathbb{N}$ passages, as we did

in (Bouziane et al., 2020), we then compute embeddings based on an earlier copy of the model and re-rank the negative samples based on their embedding cosine similarities to the embedding of the query, and we keep the top $m_f \leq m_i, m_f \in \mathbb{N}$ for negative sampling (Figure 3). Here m_f stands for m filtered and m_i for m initial. It has to be noted that this re-ranking is not done during the first epoch, but is rather only done using the frozen weights of the latest epoch, starting from the second epoch. This is a direct analogue of what is called re-targeting in the reinforcement learning setting that has been used numerous times (Mnih et al., 2013).

Using this training scheme allows getting better hard negatives for the learning process as it helps the algorithm to naturally correct itself where it is wrong, and doing so in a manner that will not overfit on handcrafted hard samples. Ideally, we would want to re-index the complete database and directly perform retrieval on it using cosine similarity search. Unfortunately in the case of the FEVEROUS dataset, the amount of data to re-index with embeddings was too large to be tractable, hence why we had to do it the proposed way. Using a large threshold for BM25 increases the probability of having high recall of both the samples it should retrieve, as well as the samples on which it would make mistakes on.

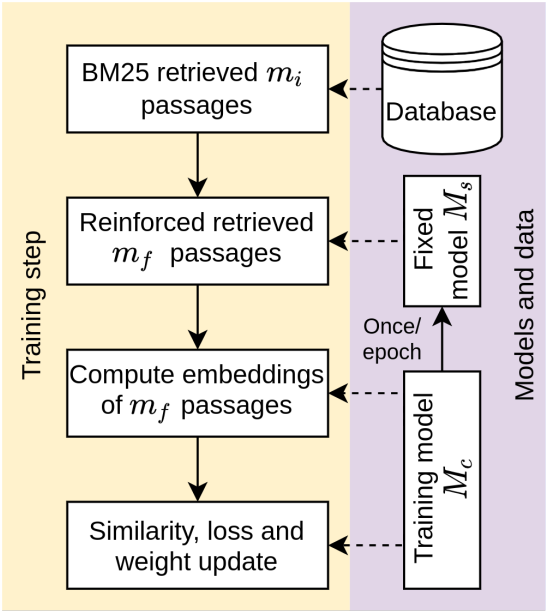
As proven efficient in reinforcement learning (Mnih et al., 2013), updating the sampling weights only once per epoch becomes very important in order to avoid undesired local minima. This principle of training is also used for re-ranking, where it, alas, has less effectiveness improving performance. This extended training scheme is one of the key contributions that were needed to achieve state-of-the-art performance on the FEVEROUS dataset.

4.2 Entailment

NEAT

For the entailment, the main challenge is to use multiple inputs (cells and sentences) to predict the true label. Furthermore, this model should be robust to noisy inputs, because the retrieved passages in the pipeline can potentially contain irrelevant pieces of evidence for the query, which can make it harder for the model to perform well. To account for this, we propose Noisy Entailment through Adapted Training (NEAT). This method consists in training TAPAS Max and TAPAS Joint with a large TAPAS

Figure 3: RARE training scheme. We sample with BM25, then use the frozen model to get better hard negatives that we use for our training loop to train the embedding creation of the algorithm, which we use to update current weights M_c. Once per epoch, we copy those weights to frozen weights M_s in order to use them for sampling.



(Herzig et al., 2020) backbone on gold passages sets as well as the results of the passage retriever. This leads to the input containing both the relevant pieces of evidences and irrelevant passages together. This joint optimisation allowed us to gain crucial points when evaluating the models on the gold dev set and on the noisy one.

Model	Gold dev	Noisy dev	Test
TAPAS Max	80 %	51 %	49.31 %
TAPAS Joint	84 %	58 %	53.24 %

Table 5: Entailment models accuracy on dev (gold and noisy) and test set.

As depicted in Table 5, TAPAS joint outperforms TAPAS Max on both the dev and test sets. When looking at the error analysis, we found out that TAPAS Max struggles to deal with noisy inputs. On one hand, the more noisy evidences we add to the gold sets, the more performance decreases. On the other hand, TAPAS Joint seemed to generalise well on both gold and noisy dev sets. For the test set, we can see that the performances of the two models are not that far apart.

Ensembling

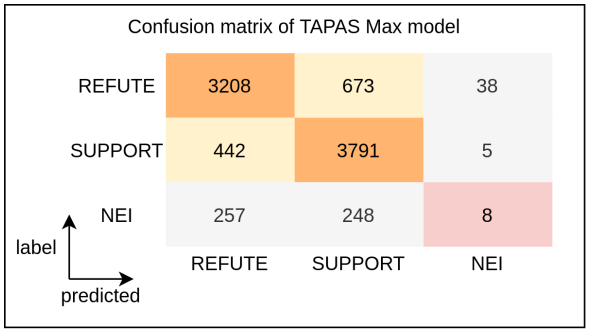
While having close global accuracy, one of the model had an easier time with NEI examples and the other with refuting examples. From that point on, ensembling them as described in Section 3 by training the MLP seemed natural. We take the scores of each model applied on dev set as an input (i.e input of dimension 6) and we output the score for each entailment class. This strategy gave us a boost of around 6% accuracy and 4 points FEVEROUS score on the test set.

Dealing with NEI

The NEIs (Not Enough Info) class represents only 3% of the train set, which makes it hard for the model to learn the right patterns and focus more on "SUPPORT" and "REFUTE" classes. Indeed, the confusion matrix in Figure 4 shows that our performance on NEIs is very low (1.5% precision performance on NEI). To solve that, future works could employ data augmentation strategies, so as to create NEI examples to balance the dataset, and thereby force the model to be equally good on the three classes. The following two strategies are examples of how samples could be generated:

- Create NEIs by concatenating the claim with a random passage from the answering pages that is not in the gold evidence sets.
- Create NEIs by removing one or more evidences from the gold evidence sets.

Figure 4: Confusion matrix of the TAPAS Max model. We can point out that it is very decent on the "SUPPORT" and "REFUTE" labels. The misclassification of NEIs does not seem to go towards one of the other labels in particular and it does not seem the model tries to predict NEI too much, but when it does, it mostly is for examples that are refuting and which do not have much evidence to begin with.



Our models struggle to converge using these augmentations, necessitating more investigation on new sampling strategies or compatible models.

5 Conclusion

Overall, this multi-modal task creates a plethora of new challenges to overcome and opens exciting avenues of research for the future of automated fact verification.

Our design improved on the classical retrieve-then-entail approach by giving special treatment to both modalities before putting them in a common multi-hop entailment model, that is necessary for multi-hop entailment. Thereby, we reached improvement of the state-of-the-art for that specific task.

Through novel and adapted training schemes, we overcame certain challenges like maximising the full set evidence recall, and allowing to have entailment work with a fixed number of evidence, discarding irrelevant pieces of evidence for its decision. This paradigm allows us to re-think the future of entailment on this type of real-world dataset, so as to see it as a selection mechanism too.

A direct consequence of this is the belief that doing entailment in an interpretable way will boost both the evidence recall and accuracy. Indeed, using the interpretation will allow deriving the passages that were useful for entailment and not those that were matching the most in terms of information overlap. This might prove to be the only way to get the right evidence set even when the refuting evidence set only deals with a small detail of the claim.

The way the dataset is constructed gives rise to the idea of an iterative process that starts with finding evidence, then performing entailment and then going back to retrieve more. Such an approach would provide a more natural way of distinguishing between NEI and SUPPORT labels and allow for reasonable runtimes. Ultimately, this will be accompanied by both more interpretable entailment and better scalability. This will in turn make the algorithm even more useful in the real world as a tool against misinformation and a general researching tool for fact verification.

Finally, a model that can deal with multiple modalities more universally and scalably is a question left for the future, making room for new research possibilities.

Those intuitions are entailed by both the insights

we gained on this new dataset, and the fact that our best effort with a more classic approach could not yield human-level results on this complex problem.

Acknowledgements

The authors want to thank the reviewers and contributors of the Buster.Ai team.

Additional elements were taken from the formatting instructions of the *International Joint Conference on Artificial Intelligence* and the *Conference on Computer Vision and Pattern Recognition*.

References

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

Mostafa Bouziane, Hugo Perrin, Aurélien Cluzeau, Julien Mardas, and Amine Sadeq. 2020. Team buster. ai at checkthat! 2020 insights and recommendations to improve fact-checking. In *CLEF (Working Notes)*.

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. *arXiv preprint arXiv:1909.02164*.

Julian Martin Eisenschlos, Syrine Krichene, and Thomas Müller. 2020. Understanding tables with intermediate pre-training. *arXiv preprint arXiv:2010.00571*.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*.

Jonathan Herzig, Paweł Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Martin Eisenschlos. 2020. Tapas: Weakly supervised table parsing via pre-training. *arXiv preprint arXiv:2004.02349*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019a. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Zhenghao Liu, Chenyan Xiong, and Maosong Sun. 2019b. Kernel graph attention network for fact verification. *CoRR*, abs/1910.09796.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. 2013. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602.

Oxford University Reuters Institute for the Study of Journalism. Digital news report 2021. https://reutersinstitute.politics.ox.ac.uk/digital-news-report/2021. Accessed: 2021-08-04.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018a. The fact extraction and VERification (FEVER) shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018b. The FEVER2.0 shared task. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*.

Wenhan Xiong, Xiang Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Scott Yih, Sebastian Riedel, Douwe Kiela, and Barlas Oguz. 2021. Answering complex open-domain questions with multi-hop dense retrieval. In *International Conference on Learning Representations*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

A Not Enough Information example

- **Claim:** The Alexander Faribault House is a historic house museum in 12 First Avenue, Faribault, MN built by Alexander Faribault, a local tradesman and philanthropist.
- **Evidence 1:** Alexander Faribault House The Alexander Faribault House is a historic house museum in Faribault, Minnesota, United States.
- **Evidence 2:** Alexander Faribault House The local address of the house is 12 First Avenue, Faribault, MN.
- **Evidence 3:** Alexander Faribault House It was built by fur trader Alexander Faribault in the Greek Revival style.

B Inference hardware

- **GPU:** NVIDIA RTX 3090 with 24 GB of VRAM, 1950 MHz boost and 10496 CUDA cores. The GPU driver version was 465.19.01, and CUDA version was 11.3.

- **CPU:** Intel i9 10920X with 12 cores and 24 threads, running at stock clocks.
- **RAM:** 64 GB of DDR4 with CAS 16 latency running at 3000 MHz, in 4 times 16 GB configuration.

C Re-filtering with BM25 example

- **Claim:** Raffaele Celeste Rosso, born on September 19, 1927 in San Michele Mondovì, was active from 1948 to 1994 under Durium Label.

Passage id	Rank retriever	Rank re-filtered
cell 0 1 1	1	1
cell 0 6 1	9	2

Table 6: Re-filtering example. In this case, we see that the cell 0 6 1 is not retrieved at the top but gets re-ranked higher by BM25 which is more sensitive to the exact date matching than the embedding cannot entirely catch.

- **cell 0 6 1:** Nini Rosso Years active Nini Rosso 1948 - 1994
- **cell 0 1 1:** Nini Rosso Born Nini Rosso Raffaele Celeste Rosso (1927-09-19) September 1