Graph Reasoning with Context-Aware Linearization for Interpretable Fact Extraction and Verification

Neema Kotonya[*1], **Thomas Spooner**[2], **Daniele Magazzeni**[2], and **Francesca Toni**[1]
[1]Department of Computing, Imperial College London
[2]J.P. Morgan AI Research
nk2418@ic.ac.uk, {thomas.spooner,daniele.magazzeni}@jpmorgan.com, ft@ic.ac.uk

Abstract

This paper presents an end-to-end system for fact extraction and verification using textual and tabular evidence, the performance of which we demonstrate on the FEVEROUS dataset. We experiment with both a multi-task learning paradigm to jointly train a graph attention network for both the task of evidence extraction and veracity prediction, as well as a single objective graph model for solely learning veracity prediction and separate evidence extraction. In both instances, we employ a framework for per-cell linearization of tabular evidence, thus allowing us to treat evidence from tables as sequences. The templates we employ for linearizing tables capture the context as well as the content of table data. We furthermore provide a case study to show the interpretability our approach. Our best performing system achieves a FEVEROUS score of 0.23 and 53% label accuracy on the blind test data.[1]

1 Introduction

Fact checking has become an increasingly important tool to combat misinformation. Indeed the study of automated fact checking in NLP (Vlachos and Riedel, 2014), in particular, has yielded a number of valuable insights in recent times. These include task formulations such as matching for discovering already fact-checked claims (Shaar et al., 2020), identifying neural fake news (Zellers et al., 2020), fact verification in scientific (Wadden et al., 2020) and public health (Kotonya and Toni, 2020b) domains, and end-to-end fact verification (Thorne et al., 2018), which is the subject of the FEVEROUS benchmark dataset (Aly et al., 2021).

A majority of automated fact checking studies only consider text as evidence for verifying claims. Recently, there have been a number of works which look at fact-checking with structured and semi-structured data, mainly in the form of tables and knowledge bases (Chen et al., 2020) — but fact-checking from both structured and unstructured data has been largely unexplored. Given the sophistication in the presentation of fake news, it is important to develop fact checking tools for assessing evidence from a wide array of evidence sources in order to reach a more accurate verdict regarding the veracity of claims.

In this work, we propose a graph-based representation that supports both textual and tabular evidence, thus addressing some of the key limitations of past architectures. This approach allows us to capture relations between evidence items as well as claim-evidence pairs, borrowing from the argumentation and argument mining literature (Cabrio and Villata, 2020; Vecchi et al., 2021), as well as argument modeling for fact verification (Alhindi et al., 2018).

We experiment with two formulations for graph learning. For the first, we employ a multi-task learning paradigm to jointly train a graph attention network (Velickovic et al., 2018) for both the task of evidence extraction — which we model as a node selection task — and a graph-level veracity prediction task. In the second, we explicitly separate the verification and extraction tasks, where standard semantic search is used for evidence extraction, and veracity prediction is treated as a graph-level classification problem.

For veracity prediction we predict a label for each claim, one of SUPPORTS, REFUTES, or NOT-ENOUGH-INFO (NEI), which is conditioned on all relevant evidence, hence the intuition to frame veracity prediction as a graph-level prediction task. In both formulations, we employ context-aware table linearization templates to produce per-cell sequence representations of tabular evidence and thus construct evidence reasoning graphs where nodes

* Work done while the author was an intern at J.P. Morgan AI Research.

[1]This system was not submitted to the shared task competition, but instead to the after competition leader board under the name **CARE** (**C**ontext **A**ware **RE**asoner).

Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER) at EMNLP 2021, pages 21–30

have heterogeneous evidence types (i.e., representing sentences and tables on the same evidence reasoning graph).

Contributions. The three main contributions of the paper are summarized below:

1. Provide **insightful empirical analysis** of the new FEVEROUS benchmark dataset.
2. Propose a novel framework for interpretable fact extraction using templates to derive **context-aware per-cell linearizations**.
3. Present a **graph reasoning model** for fact verification that supports both structured and unstructured evidence data.

Both the joint model and separately trained models exhibit a significant improvement over the FEVEROUS baseline, as well as significant improvements for label accuracy and evidence recall. Our separated approach to fact extraction and verification achieves a FEVEROUS score of 0.23 and label accuracy of 53% on the blind test data.

2 Related Work

Graph Reasoning for Fact Verification. Several works explore graph neural networks (GNN) for fact extraction and verification, both for fine-grained evidence modelling (Liu et al., 2020; Zhong et al., 2020) and evidence aggregation for veracity prediction (Zhou et al., 2019). Furthermore, graph learning has also been leveraged to build fake news detection models which learn from evidence from different contexts; e.g., user-based and content-based data (Liu et al., 2020; Lu and Li, 2020). There are also non-neural approaches to fake news detection with graphs (Ahmadi et al., 2019; Kotonya and Toni, 2019). However, to the best of our knowledge, this work is the first to employ a graph structure to jointly reason over both text and tabular evidence data in both single task learning (STL) and multi-task learning (MTL) settings.

Table Linearization. A number of approaches have been adopted in NLP for table linearization. For example, Gupta et al. (2020) study natural language inference in the context of table linearizations, in particular they are interested to see if language models can infer entailment relations from table linearizations. The linearization approach employed by Schlichtkrull et al. (2021) is also used for automated fact verification. However, they linearize tables row- and column-wise, whereas we focus on cells as evidence items in the FEVEROUS dataset are annotated at table-cell level.

3 Data Analysis

Further to the FEVEROUS dataset statistics discussed by the task description paper (Aly et al., 2021), we perform our own data exploration. We present insights from our data analysis of the FEVEROUS dataset, which we use to inform system design choices.

Table types. Wikipedia tables can be categorized into one of two classes: infoboxes and general tables. Infoboxes are fixed format tables which typically appear in the top right-hand corner of a Wikipedia article. General tables can convey a wider breadth of information (e.g., election results, sports match scores, the chronology of an event) and typically have more complex structures (e.g., multiple headers). List items can also be considered as a special subclass of tables, where the number of items is analogous to the number of columns and the nests of the list signify table rows.

Evidence types. The first observation we make is that, similar to the FEVER dataset (Thorne et al., 2018), a sizeable portion of the training instances rely on evidence items which are extracted from the first few sentences of a Wikipedia article. The most common evidence items are the first and second sentences in a Wikipedia article, which appear in 36% and 18% of evidence sets, respectively. The four most frequent evidence cells all come from the first table, with 49% of first tables listed as evidence in the train and dev data being infoboxes. Further, the vast majority of cell evidence items are non-header cells, but these only account for approximately 5.1% of tabular evidence in the train and dev datasets. A summary of these findings is provided in Table 1 for the most common evidence types in the training data.

Evidence item co-occurrences. We investigate the most common evidence pairs, both in individual evidence sets and also in the union of all evidence sets relating to a claim. The most common evidence pair in the training data is (SENTENCE_0, SENTENCE_1), which accounts for 3.2% of evidence co-occurrences. The most common sentence-table cell co-occurrence is (CELL_0_2_1, SENTENCE_0). The most common table cell pair is

Evidence type	% Evidence sets
List items	1.6%
Sentences	67.7%
All tables	58.2%
Infoboxes	26.5%
General tables	33.9%

Table 1: Prevalence of evidence types in the training data by number of evidence sets in which they appear.

(CELL_0_2_0, CELL_0_2_1). All of the ten most common co-occurrences either contain one of the first four sentences in an article or evidence from one of the first two tables.

NEI label. Lastly, we choose to explore instances of the NEI class. We sample 100 instances of NEI claims from the training data and note their qualitative attributes. We pay particular attention to this label as it is the least represented in the data. Unlike the FEVER score, the FEVEROUS metric requires the correct evidence, as well as the label, to be supplied for an NEI instance for credit to awarded. Our analysis is summarized in Table 2. We categorize mutations, using the FEVEROUS annotation scheme, as one of three types: entity substitution, including more facts than available in the provided evidence (i.e., including additional propositions), and paraphrasing or generalizing. We use *Other* to categorize claims with a mutation not captured by one of these three categories.

Mutation Type	% Sample
Entity Substitution	21%
More facts than in evidence	42%
Paraphrasing or generalizing	36%
Other	1%

Table 2: We sample 100 NEI instances and categorize them according to the type of lexical mutation which results in the claim being unverifiable.

We note that a number of NEI examples are mutations of SUPPORTS or REFUTES examples. For example the claim in Table 3 is a mutation of a SUPPORTS instance where entity substitution (humans → reptiles) has been used to make the first clause unverifiable, hence changing the label to NEI.

Claim
Nucleoporin 153, a protein which in **reptiles** is encoded by the NUP153 gene, is an essential component of the basket of nuclear pore complexes (NPCs) in vertebrates, and required for the anchoring of NPCs.
Evidence
Nucleoporin 153 (Nup153) is a protein which in **humans** is encoded by the NUP153 gene. It is an essential component of the basket of nuclear pore complexes (NPCs) in vertebrates, and required for the anchoring of NPCs.

Table 3: NEI example where the evidence is highlighted according to the part of the claim to which it refers. The text in **bold** is the substitution which resulted in the label changing from SUPPORTS to NEI.

4 Methods

Our proposed method for fact verification is an end-to-end system comprising three modules:

(1) A robust document retrieval procedure (see Section 4.1).

(2) An evidence graph construction and intermediate evidence filtering process (see Section 4.2).

(3) A joint veracity label prediction and evidence selection layer that reasons over the evidence graph (see Section 4.3).

An illustration of the complete pipeline is provided in Figure 1, and details of each processing stage are provided in the following sections.

4.1 Document Retrieval

For document retrieval, we employ an entity linking and API search approach similar to that of Hanselowski et al. (2018). The WikiMedia API[2] is used to query Wikipedia for articles related to the claim, using named entities and noun phrases from the claim as search terms. These retrieved Wikipedia page titles form our candidate document set. Named entities that are not retrieved by the API are then extracted from the claim as a handful of these identify pages which are present in

[2] https://www.mediawiki.org/wiki/API

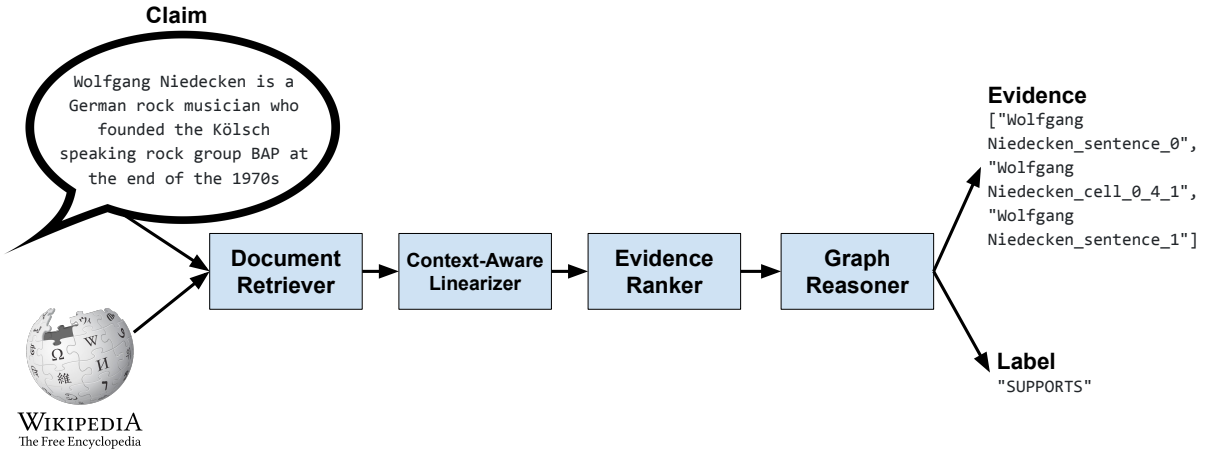

Figure 1: Our fact verification pipeline. We employ two graph reasoning approaches: STL where the evidence extraction and modelled separately, and MTL where further evidence filtering is performed jointly with veracity prediction by the Graph Reasoner.

the Wikipedia dump (e.g., **/wiki/Lars_Hjorth** is present in the provided Wikipedia evidence dump, but is not returned by the WikiMedia API). In the same vein, we discard titles which are returned by the API, but are not in the Wikipedia dump. TF-IDF and cosine similarity are employed to score and rerank the retrieved Wikipedia articles with respect to their similarity to the claim.

As in the approach of Hanselowski et al. (2018), the seven highest ranked pages are chosen at test time. For completeness, we also experiment with approaches to document retrieval which select pages based on a threshold score (Nie et al., 2019). Ultimately, we find these methods yield lower precision.

4.2 Evidence Reasoning Graph

Similar to other fact verification systems (Augenstein et al., 2019; Hidey et al., 2020), we jointly train our model for both the evidence selection and veracity prediction tasks. In contrast to these approaches, however, we employ a graph reasoning module for the joint learning of the two tasks. We choose this approach to exploit the permutation invariance of evidence with respect to a claim, as there is no canonical ordering of evidence. Our graph formulation differs from previous graph-based fact verification systems in that we construct a *heterogeneous graph to model both tabular and sequence evidence data*.

In the following sections we will describe two specific approaches that are taken for the fact verification task: (1) where we condition the graph model to learn both node-level, fine-grained evidence selection and graph-level veracity label prediction simultaneously, and (2) where we only learn graph-level veracity prediction.

Linearizing Tabular Data. We linearize both table and list evidence data and generate from these linearizations a contextualized sequence representation which captures information about each cell as well as its surrounding page elements. This is accomplished using templates that distinguish explicitly between infoboxes and general tables. For the latter, we engineer the templates to handle two particular complexities that are present only in general tables: (1) nested headers, and (2) table cells which span multiple rows and multiple columns (see Figure 2). Furthermore, we also employ templates for producing context-rich representations of item lists (see Table 4 for more details).

Club	Season	League		
		Division	Apps	Goals
Santa Cruz	2019	Série C	7	1
Athletico Paranaense	2020	Serie A	0	0
	2021		0	0
	Total		0	0
Guarani (loan)	2020	Série B	5	0

Figure 2: Example of a complex general table taken from **/wiki/Elias_Carioca**. This table contains both multi-row cells and multi-column cells, some of which are headers. They are shown highlighted.

Graph Structure. We construct a fully connected graph $G = (V, E)$, where each node $n_i \in V$ represents a claim-evidence pair, similar to previ-

Evidence Type	Linearization	Example from FEVEROUS dataset
Infoboxes		
Headers	TABLE **has** CELL_I_J [in SUBHEADER]	Brewster Productions has Genres . [**/wiki/Brewster_Productions**]
Non-headers	CELL_I_0 **of** TABLE [in SUBHEADER] **is** CELL_I_J	Current ranking of Barbora Krejčíková in Singles is No. 65 (16 November 2020) . [**/wiki/Barbora_Krejcikova**]
General tables		
Headers	TABLE **has** CELL_I_J [in SUBHEADER]	The 1964 United States Senate election in Maine has Party . [**/wiki/1908_Clemson_Tigers_football_team**]
Non-headers	TABLE/PAGE **has** SUBHEADER_0 CELL_I_0 **in** SUBHEADER_J **of** CELL_I_J	2014 Ladies European Tour has Rank 9 in Player of Florentyna Parker . [**/wiki/2014_Ladies_European_Tour**]
List items		
Without subheaders	TITLE **includes** ITEM_I_J	Site includes Location, a point or an area on the Earth's surface or elsewhere. [**/wiki/Site**]
With subheaders	SUBHEADERS **for** TITLE **includes** ITEM_I_J	The Player Honours for Park Sang-in includes K-League Best XI: 1985 [**/wiki/Park_Sang-in**]

Table 4: Templates for encoding tabular evidence. CELL_I_0, SUBHEADER_0, SUBHEADER_J, SUBHEADERS, TABLE, TITLE and PAGE are all context elements. The content of the evidence item is highlighted . In each case ITEM_I_J denotes list item content and CELL_I_J denotes table cell content.

ous evidence graphs for automated fact checking (Zhao et al., 2020; Zhou et al., 2019). Self-loops are also included in G for each node in order to improve evidence reasoning, so the set of edges for the graph is $E = \{(n_i, n_j) \mid n_i, n_j \in V\}$.

At test time, we take the Wikipedia pages output by the document retrieval module, segment each Wikipedia page into its constituent page items (i.e., sentences, table cells, table captions and list items), and refer to these as evidence items. These evidence items are then filtered. Using an ensemble of pre-trained S-BERT sentence embeddings (Reimers and Gurevych, 2019), we perform semantic search with the claim as our query. Cosine similarity is then used to rank the evidence items. For the joint and single training approaches, we select a different number of evidence nodes; in particular, a larger graph is used with the former. For training, we select nodes to occupy the graph according to the following rule-set:

(1) If gold evidence, include as a node.

(2) For claims that require a single evidence item, include the top four candidates returned using our semantic search approach as nodes.

(3) For claims with more than one gold evidence item, retrieve the same number of candidates as gold items.

The union of these sets form the collection of nodes, V, that occupy the evidence graph G.

Node Representations. For the initial node representations, similar to Liu et al. (2020) and Zhao et al. (2020), we represent evidence nodes with the claim to which they refer as context. The claim is concatenated with a constructed context-rich evidence sequence e_i. When constructing the sequences, e_i, we consider the unstructured evidence items (i.e, sentences and table captions) and the structured table and list items separately.

For sentences and table captions the evidence sequence is generated by concatenating the evidence item with the page title which serves as context.

For table cells and list items we perform a per cell linearization, where this linearization forms the evidence sequence for table and list item evidence items (see Table 4 for the templates used). For each evidence item, we feed this claim-evidence sequence pair to a RoBERTa encoder (Liu et al., 2019), and each node $\mathbf{n}_i \in V$ in an evidence graph has the pooled output of the last hidden state of the [CLS] token, $\mathbf{h}_i^0$ as its initial state:

$$\mathbf{n}_i = \mathbf{h}_i^0 = \text{RoBERTa}_{\text{CLS}}(c, e_i). \quad (1)$$

4.3 Evidence Selection and Veracity Prediction

Training graphs. We train two graph networks, one for joint veracity prediction and evidence extraction, and the second solely for the veracity prediction task.

Oversampling NEI Instances. As discussed in Section 3, the FEVEROUS dataset suffers from a significant class imbalance with respect to the NEI instances. Similar to the baseline approach, we employ techniques for generating new NEI instances in order to address this issue. Concretely, we use two data augmentation strategies in order to increase the number of NEI at train time: (1) evidence set reduction, and (2) claim mutation. For the first case, we randomly sample SUPPORTS and REFUTES instances and drop evidence. Given the distribution of entity substituted and non-entity substituted mutations — as discovered in our data analysis (see Section 3) — we make the choice to include in the training data: 15,000 constructed NEI examples made using the first approach, and 5,946 NEI examples constructed using the second. This means that a total of 92,237 NEI examples were used for model training.

STL: Separate Verification and Extraction. For the first model, we perform the tasks of fact extraction and verification of evidence selection and veracity prediction separately. We make use of an ensemble semantic search method for extracting top evidence items for claims. We employ S-BERT[3] to encode the claim and the evidence items separately. We then compute cosine similarity for the claim evidence pair.

The 25 highest ranking tabular evidence items were chosen, and the top-scoring 5 sentences (and captions) for each claim were selected as the nodes of our evidence reasoning graph at test time. This is the evidence limit stated by the FEVEROUS metric.

When constructing the evidence graph at test time, we choose to exclude header cells and list items evidence types as nodes as they account for a very small portion of evidence items (see Section 3), and experimentation shows that the evidence extraction model has a bias to favour these evidence elements over sentences. We use two GAT layers in our graph reasoning model, with: a hidden layer size of 128, embeddings size of 1024, and a global attention layer for node aggregation. The logits generated by the model are fed directly to a categorical cross entropy loss function, and the veracity label output probability distribution $\mathbf{p}_i$, for each evidence graph $G_i \in \mathcal{G}$, is computed using the relation

$$\mathbf{p}_i = \text{softmax}(\text{MLP}(\mathbf{W}\mathbf{o}_i + \mathbf{b})), \quad (2)$$

where

$$\mathbf{o}_i = \sum_{n_i \in V}^{n_i} \text{softmax}\left(h_{\text{gate}}(\mathbf{x}_n)\right) \odot h_{\mathbf{\Theta}}(\mathbf{x}_n). \quad (3)$$

MTL: Joint Verification and Extraction. We also experiment with a joint training or multi-task learning (MTL) approach in order to explore if simultaneously learning the veracity label and evidence items can lead to improvements in the label accuracy metric and also evidence prediction recall and precision. For this approach, we construct larger evidence graphs at test time, including the thirty-five highest ranked evidence items according to the S-BERT evidence extraction module. The intention is for the graph network to learn a binary classification for each claim-evidence pair in the network.

For the multi-task learning model, we increase the dimensions of our graph network by feeding our initial input graphs to two separate GAT components (in order to increase the model's capacity for learning the more complex multi-task objective), the outputs of which, $\mathbf{h}_a$ and $\mathbf{h}_b$, are concatenated to form representation $\mathbf{h}$ over which we compute global attention, where the combined representation takes the form:[4]

$$\mathbf{h} = [\mathbf{h}_a; \mathbf{h}_b]. \quad (4)$$

[3] We use the 'msmarco-distilbert-base-v4' and 'paraphrase-mpnet-base-v2" pretrained models.

[4] We denote the concatenation of vectors $\mathbf{x}$ and $\mathbf{y}$, by $[\mathbf{x}; \mathbf{y}]$.

The binary cross entropy loss is then used for the node-level evidence selection task, and, as with the separated model, we use categorical cross entropy to compute the graph-level veracity prediction, as shown in (2) and (3). The resulting joint graph neural network is then trained with the linear-additive objective

$$\mathcal{L}_{\text{joint}} = \lambda \mathcal{L}_{\text{evidence}} + \mathcal{L}_{\text{label}}, \quad (5)$$

taking the form of a Lagrangian with multiplier $\lambda \geq 0$, where

$$\mathcal{L}_{\text{evidence}} = \text{sigmoid}(\text{MLP}(\mathbf{W}_i \mathbf{h} + \mathbf{b})). \quad (6)$$

As with the previous approach, we feed the model logits to our loss functions and use an Adam optimizer to train the network, and set $\lambda = 0.5$.

4.4 Hyper-parameter Settings

For all models, we make use of a RoBERTA-LARGE model which is pre-trained on a number of NLI datasets including NLI-FEVER (Nie et al., 2020). We use a maximum sequence length of 512 for encoding all claim-evidence concatenated pairs. We experiment with the following learning rates [1e-5, 5e-5, 1e-4], ultimately choosing the learning rate underlined. Training was performed using batch size of 64. We train the single objective model for 20k steps, choosing the weights with the minimum veracity prediction label loss, and train the joint model for 20k steps, taking the model with highest recall for evidence extraction. The Adam optimizer is used in training for both approaches.

5 Results

We report the results of the entire fact extraction and verification pipeline, as well as the evaluation of the pipeline's performance for intermediate stages of the fact verification system, e.g., document retrieval and evidence selection.

Document retrieval. Our method for DR shows significant improvement on the TF-IDF+DrQA approach used by the baseline. In particular we find that our document retrieval module sees gains from querying the Wikipedia dump for pages related to entities which are not retrieved by the WikiMedia API. However, we do note that our approach struggles to retrieve Wikipedia pages in cases relating to specific events which can only be inferred through reasoning over the claim.

For example, consider the following claim from the development dataset: *"2014 Sky Blue FC season number 18 Lindsi Cutshall (born October 18, 1990) played the FW position."*. In this case, the document selection process returns *"Sky Blue FC"*, *"Lindsi Cutshall"*, and *"2015 Sky Blue FC season"*, but does not return the gold evidence page *"2014 Sky Blue FC season"* which is required for verification of the claim.

We report recall@k for $k = \{3, 5, 7\}$ where k is the number of Wikipedia page documents retrieved by the module. Our approach shows significant improvements over the baseline (see Table 5).

Method	Rec@3	Rec@5	Rec@7
Baseline	0.58	0.69	–
Ours	0.65	0.73	**0.80**

Table 5: Document retrieval results measured by Recall@k, where k is the number of documents retrieved. Results reported for the dev set.

Evidence selection and veracity prediction. For evidence selection and veracity prediction, we observe that the approach trained for the single objective of veracity prediction marginally outperforms the jointly trained module (see Table 6). We hypothesize that the difficulty of learning to select the correct evidence nodes along with predicting veracity might be the cause of this. It is possible that performance of the joint model could be improved with better evidence representation or through the use of a different graph structure, e.g., by incorporating edge attributes.

Method	Recall	LA
Baseline	29.51	53.22
STL	**37.20**	**62.89**
MTL	36.25	62.21

Table 6: System performance of the dev set for evidence recall and label accuracy.

Finally, we submitted our blind test results for STL, which is our best performing method, to the after-competition FEVEROUS leaderboard. Our system outperforms the baseline significantly on both the FEVEROUS metric and also label accuracy as reported in Table 7. Furthermore, our results on the blind test data show almost no degradation from development to test set with respect to

the evidence recall which remains at 37%. So the cause of our reduced FEVEROUS score between the development and test data is mainly due to a decrease in label accuracy from 63% on the development data to 53% for the test data. We are confident that this could be improved with better label accuracy for the NEI class.

Method	Dev		Test	
	LA	FS	LA	FS
Baseline	53.22	19.28	47.60	17.70
Ours	**62.81**	**25.71**	**53.12**	**22.51**

Table 7: Results for label accuracy (**LA**) and FEVEROUS score (**FS**) for the full pipeline on both the development and blind test datasets.

5.1 Case Study and System Interpretability

We present an example of a claim from the development dataset, which requires both tabular and textual evidence to be verified. We show how it is labelled by our pipeline (see Table 8). For this example, our evidence selection module correctly identifies all three evidence items required to fact-check the claim. Furthermore, two of the three evidence items receive the highest relevance scores from our evidence selection module. Of the irrelevant evidence items retrieved for this claim, eleven out of twenty-two come from an unrelated Wikipedia page ("Scomadi Turismo Leggera"). The correct label of SUPPORTS is also predicted for this instance.

In order to explore the interpretability system predictions, for this same instance, we analyse the node attention weights for the first GAT layer, they are shown in parenthesis for each predicted evidence item in Table 8. We can see that the two evidence nodes with the highest values both correspond to items in the gold evidence set. However the third gold evidence item, SCOMADI_SENTENCE_15, has a much lower weight than a number of items which are not in the gold evidence set.

Claim "In 2019, Scomadi, a private limited company with limited liability, was bought by a British owner which changed Scomadi's management structure."	
Evidence Scomadi_cell_0_0_1, Scomadi_sentence_14, Scomadi_sentence_15.	
Predicted Evidence	
(1) Scomadi_cell_0_0_1	(*0.1794*),
(2) Scomadi_sentence_14	(*0.1203*),
(3) Scomadi_table_caption_0	(*0.0871*),
(4) Scomadi_cell_0_3_1	(*0.0685*),
(5) Scomadi_cell_0_7_1	(*0.0561*),
(6) Scomadi_cell_0_2_1	(*0.0472*)
(7) Scomadi_cell_0_8_1	(*0.0405*)
(8) Scomadi_sentence_15	(*0.0360*),
(9) Scomadi_sentence_11	(*0.0324*),
(10) Scomadi_sentence_0	(*0.0292*),
(11) Scomadi_cell_0_6_1	(*0.0266*),
(12) Scomadi_cell_0_5_1	(*0.0243*),
(13) Scomadi_cell_0_1_1	(*0.0224*),
(14) Scomadi_cell_0_4_1	(*0.0208*).
Label	SUPPORTS
Predicted Label	SUPPORTS

Table 8: Example claim from the development dataset which requires extracting both tabular and textual evidence in order for it to be verified. For brevity we only show the top fourteen (out of twenty-five) extracted evidence items, correctly predicted evidence is highlighted.

6 Conclusion and Future Work

In this work, we have demonstrated two novel approaches for fact extraction and verification that support both structured and unstructured evidence. These architectures were motivated by literature in argumentation, and also by the empirical analysis presented in Section 3. Our results show significant improvement over the shared task baseline for both the joint and separated models, with the latter generating a marginal improvement on the FEVEROUS metric compared with the former. Overall, we conclude that the use of graph-based reasoning in fact verification systems could hold great promise for future lines of work.

We hypothesize that exploring varied task formulations could potentially yield strong improvements in model performance, for example: constructing reasoning graphs on an evidence set level, or using the FEVER dataset to augment the NEI claims used during training, or further fine-tuning sentence embeddings on the FEVEROUS dataset. Furthermore, we believe further insights could be gained by evaluating our table linearization approach on other datasets related to fact verification over tabular data. In addition to this, we hope to conduct

further experiments with our graph based approach using structured and unstructured evidence independently, to further investigate which aspect of our approach led to the improvement on the FEVEROUS score.

Incorporating prior knowledge or constraints into the training procedure would also be an interesting direction. Finally, we believe that our graph-based approach lends itself well to the extraction of veracity prediction *explanations* (Kotonya and Toni, 2020a), obtained from evidence extracted from our underpinning graphs as justifications for claims. The ability to provide evidence for a claim, and to justify this, would better enable the integration of these techniques in practical systems.

References

Naser Ahmadi, Joohyung Lee, Paolo Papotti, and Mohammed Saeed. 2019. Explainable fact checking with probabilistic answer set programming. In *Proceedings of the 2019 Truth and Trust Online Conference (TTO 2019), London, UK, October 4-5, 2019*.

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.

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.

Isabelle Augenstein, Christina Lioma, Dongsheng Wang, Lucas Chaves Lima, Casper Hansen, Christian Hansen, and Jakob Grue Simonsen. 2019. MultiFC: A real-world multi-domain dataset for evidence-based fact checking of claims. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4685–4697, Hong Kong, China. Association for Computational Linguistics.

Elena Cabrio and Serena Villata, editors. 2020. *Proceedings of the 7th Workshop on Argument Mining*. Association for Computational Linguistics, Online.

Wenhu Chen, Hongmin Wang, Yunkai Zhang Jianshu Chen, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020. Tabfact : A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations (ICLR)*, Addis Ababa, Ethiopia.

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.

Andreas Hanselowski, Hao Zhang, Zile Li, Daniil Sorokin, Benjamin Schiller, Claudia Schulz, and Iryna Gurevych. 2018. UKP-athene: Multi-sentence textual entailment for claim verification. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 103–108, Brussels, Belgium. Association for Computational Linguistics.

Christopher Hidey, Tuhin Chakrabarty, Tariq Alhindi, Siddharth Varia, Kriste Krstovski, Mona Diab, and Smaranda Muresan. 2020. DeSePtion: Dual sequence prediction and adversarial examples for improved fact-checking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8593–8606, Online. Association for Computational Linguistics.

Neema Kotonya and Francesca Toni. 2019. Gradual argumentation evaluation for stance aggregation in automated fake news detection. In *Proceedings of the 6th Workshop on Argument Mining*, pages 156–166, Florence, Italy. Association for Computational Linguistics.

Neema Kotonya and Francesca Toni. 2020a. Explainable automated fact-checking: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5430–5443, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Neema Kotonya and Francesca Toni. 2020b. Explainable automated fact-checking for public health claims. In *Proceedings of the 2020 Conference on*

Empirical Methods in Natural Language Processing (EMNLP), pages 7740–7754, 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.

Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2020. Fine-grained fact verification with kernel graph attention network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7342–7351, Online. Association for Computational Linguistics.

Yi-Ju Lu and Cheng-Te Li. 2020. GCAN: Graph-aware co-attention networks for explainable fake news detection on social media. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 505–514, Online. Association for Computational Linguistics.

Yixin Nie, Songhe Wang, and Mohit Bansal. 2019. Revealing the importance of semantic retrieval for machine reading at scale.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Michael Sejr Schlichtkrull, Vladimir Karpukhin, Barlas Oguz, Mike Lewis, Wen-tau Yih, and Sebastian Riedel. 2021. Joint verification and reranking for open fact checking over tables. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6787–6799, Online. Association for Computational Linguistics.

Shaden Shaar, Nikolay Babulkov, Giovanni Da San Martino, and Preslav Nakov. 2020. That is a known lie: Detecting previously fact-checked claims. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3607–3618, Online. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Eva Maria Vecchi, Neele Falk, Iman Jundi, and Gabriella Lapesa. 2021. Towards argument mining for social good: A survey. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1338–1352, Online. Association for Computational Linguistics.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

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.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2020. Defending against neural fake news.

Chen Zhao, Chenyan Xiong, Corby Rosset, Xia Song, Paul Bennett, and Saurabh Tiwary. 2020. Transformer-xh: Multi-evidence reasoning with extra hop attention. In *International Conference on Learning Representations*.

Wanjun Zhong, Jingjing Xu, Duyu Tang, Zenan Xu, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2020. Reasoning over semantic-level graph for fact checking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6170–6180, Online. Association for Computational Linguistics.

Jie Zhou, Xu Han, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2019. GEAR: Graph-based evidence aggregating and reasoning for fact verification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 892–901, Florence, Italy. Association for Computational Linguistics.