KSU at CheckThat! 2025: Two-stage approach to fact-checking numerical claims

Notebook for the CheckThat! Lab at CLEF 2025

Keito Fukuoka[1,*,†], Hisashi Miyamori[1,†]

[1]Kyoto Sangyo University of Japan (KSU University), Kamigamo Motoyama, Kita-ku, Kyoto City, Kyoto, Japan

Abstract

The spread of misinformation containing numerical claims online poses a severe threat, undermining the very foundation of democracy. This paper proposes a fact-checking method for automatically determining the veracity of claims that include numerical and temporal elements. The proposed method consists of a two-stage process: evidence retrieval and classification. Specifically, it combines comprehensive evidence retrieval using a Contriever model enhanced by SimCSE-based contrastive learning with a classification method that extracts crucial evidence using a Large Language Model (LLM). For experiments, we used the English dataset provided by CheckThat! 2025 Task 3. In the evidence retrieval task, the Contriever model with SimCSE-based contrastive learning achieved a Recall@100 of 0.524, significantly outperforming conventional methods like BM25. Conversely, in the classification task, the method utilizing search results from BM25 achieved the highest performance with a macro F1 of 0.5054. A significant insight gained from this study is that improvements in evidence retrieval ranking accuracy do not necessarily directly lead to enhanced classification performance.

Keywords

Fact-checking, Numerical claims, Evidence retrieval, Contrastive learning

1. Introduction

The spread of misinformation online, particularly prominent during election periods, not only triggers social and political unrest but also poses a severe threat, undermining the very foundation of democracy[1]. Among various forms of misinformation, verifying claims that include numerical and temporal elements is of paramount importance in fact-checking. Indeed, numerical claims constitute a significant component of political discourse.

This paper addresses the CheckThat! Lab's Task 3: Fact-checking numerical claims [2]. The objective of this task is to determine the veracity of claims containing numerical quantities and temporal expressions. For each claim, participants are provided with a short list of evidence and are required to classify the claim as "True," "False," or "Conflicting" based on this evidence.

We propose a two-stage fact-checking method consisting of an evidence retrieval step enhanced by contrastive learning and a classification step that combines LLM-based crucial evidence extraction. First, in the evidence retrieval step, we observed that claims and their corresponding evidence often have different phrasings, even when their content is highly relevant. To comprehensively retrieve highly relevant evidence, we adopted an evidence retrieval system composed of a Contriever model further trained with SimCSE-based contrastive learning to capture the semantic relevance between claims and evidence. Furthermore, in the classification using the retrieved evidence, we confirmed that gold evidence in this task tends to be lengthy. To mitigate any negative impact on classification, we therefore adopted a method that uses an LLM to extract important information from the evidence and then performs classification based on these extracted results.

CLEF 2025 Working Notes, September 9 – 12 September 2025, Madrid, Spain

i2486200@cc.kyoto-su.ac.jp (K. Fukuoka); miya@cc.kyoto-su.ac.jp (H. Miyamori)

2. Related Work

Automated fact-checking has garnered significant attention as a crucial countermeasure against online misinformation [3, 4, 5]. Existing fact-checking research has largely been limited to synthetic claims [6] and non-numerical claims [7], with a notable lack of focus on claims containing numerical information.

Addressing this gap, Viswanathan et al. constructed QUANTEMP [8], an open-domain benchmark specifically designed for real-world numerical claims. QUANTEMP is a diverse dataset encompassing comparisons, statistics, durations, and temporal aspects, offering detailed metadata and evidence. Using this dataset, they evaluated the limitations of existing methods and presented new challenges in numerical claim verification.

The Task 3: Fact-checking numerical claims that we address in this paper aligns with the challenges posed by QUANTEMP. This task defines two sub-tasks for determining the veracity of claims: an "evidence retrieval task" to search for relevant evidence and a "classification task" to categorize claims based on that evidence.

3. Method

This task broadly consists of the following two components:

- Evidence retrieval task: retrieving evidence relevant to a given claim.
- Classification task: determining whether a claim is True, False, or Conflicting based on the claim and retrieved evidence.

3.1. Task Formulation

This task is formulated as follows. Given a claim $c \in \mathcal{C}$ ($\mathcal{C}$ is the claim space) as a query, and a sequence of top-k retrieved evidences $E = (e_1, e_2, ..., e_k) \in \mathcal{E}$ ($\mathcal{E}$ is the evidence sequence space) obtained by a retrieval system S, a classification function f outputs a label $y \in \mathcal{L} = \{True, False, Conflicting\}$:

$$f : (\mathcal{C}, \mathcal{E}) \rightarrow \mathcal{L} \tag{1}$$

Here, the process of obtaining the evidence sequence E for a claim c by the retrieval system S is expressed as follows:

$$E = \text{top-k}(\text{sort}_{d \in D_c}(\text{score}(c, d))) \tag{2}$$

$$= (e_1, e_2, \ldots, e_k) \text{ s.t. } \text{score}(c, e_1) \geq \text{score}(c, e_2) \geq \cdots \geq \text{score}(c, e_k) \tag{3}$$

where D_c is the set of evidences relevant to claim c, $D_c = \{d_1, d_2, \ldots, d_n\}$, $\text{score}(q, d)$ is a function that returns the relevance score of document d for query q, $\text{sort}_{x \in X}(f(x))$ is a function that sorts each element x in set X in descending order based on the value of function $f(x)$, top-k(X) is a function that returns the top-k elements of sequence X, and e_i represents the i-th evidence.

Furthermore, each label $l \in \mathcal{L}$ represents one of the following three types of content:

- $True$: Based on the retrieved evidence, the claim c is determined to be true.
- $False$: Based on the retrieved evidence, the claim c is determined to be false.
- $Conflicting$: Based on the retrieved evidence, it is not possible to determine whether the claim c is true (insufficient evidence or conflicting content).

The classification model takes the claim c and the retrieved evidence sequence E as input and outputs the probability $P(l|c, E)$ for label l:

$$P(y|c, E) = \text{softmax}(h(c, E)) \tag{4}$$

where $h(c, E)$ represents the feature representation by a neural network, and $y \in \mathcal{L} = \{True, False, Conflicting\}$ represents the predicted label.

The final predicted label $\hat{y}$ is determined as follows:

$$\hat{y} = \arg\max_{y} P(y|c, E) \tag{5}$$

Thus, it is important to note that the classification task depends on the results of the evidence retrieval task, and the ranking performance of the retrieval system affects the accuracy of the classification results.

3.2. Evidence Retrieval

3.2.1. Dataset Construction for Evidence Retrieval Evaluation

Evidence retrieval is the process of selecting highly relevant evidence for a given claim. In this task, explicit claim-evidence pairs are not provided in the supplied data, which makes evaluating ranking performance challenging. To address this, we explicitly constructed claim-evidence pairs by leveraging the gold evidences present in the validation data.

Let $C = \{c_1, c_2, \ldots, c_n\}$ be the set of claims in the validation data, and G_i be the gold evidence corresponding to each claim c_i. We segmented each gold evidence G_i into individual sentences to obtain a set of evidence sentences $S_i = \{s_{i,1}, s_{i,2}, \ldots, s_{i,m_i}\}$. The relevance label $r(c_i, s_j)$ is defined as follows:

$$r(c_i, s_j) = \begin{cases} 1 & \text{if } s_j \in S_i \\ 0 & \text{if } s_j \in \bigcup_{k \neq i} S_k \end{cases} \tag{6}$$

Through this process, we constructed a dataset $D_{cs} = \{(c_i, s_j, r(c_i, s_j)) | i \in \{1, \ldots, n\}, j \in \{1, \ldots, |S|\}\}$ consisting of 13,019 claim-evidence pairs (train: 9,935 pairs, dev: 3,084 pairs), enabling quantitative evaluation of ranking performance in evidence retrieval. Here, $S = \bigcup_{i=1}^{n} S_i$ represents the set of all evidence sentences.

3.2.2. SimCSE-based Contrastive Learning for Contriever

When retrieving evidence sentences relevant to claims using dense retrieval, models may struggle to generalize to novel topics not present in the training data, potentially performing worse than conventional sparse retrieval methods like BM25. The Contriever model has been shown to outperform BM25 [9] in terms of Recall@100 on various datasets, even when pre-trained in an unsupervised manner, by pre-training a dense retriever with contrastive learning [10]. Therefore, to enable the Contriever model to more comprehensively retrieve evidence sentences relevant to claims, we further trained the model using SimCSE-based contrastive learning on the semantic relatedness between claims and gold evidences.

Contrastive learning is performed using claim c_i and a sentence $s_{i,j}$ extracted from its gold evidence as a positive pair, and claim c_i and a sentence $s_{k,l}$ extracted from another claim's gold evidence as a negative pair.

First, the Contriever encoder is used to convert claim c_i and evidence sentence $s_{i,j}$ (or $s_{k,l}$) into vector representations:

$$\mathbf{h}_c = \text{MeanPooling}(\text{Contriever}(c_i), \text{mask}_c) \tag{7}$$

$$\mathbf{h}_s = \text{MeanPooling}(\text{Contriever}(s_{i,j}), \text{mask}_s) \tag{8}$$

Here, MeanPooling is an average pooling operation that considers the attention mask, and mask_c and mask_s are the attention masks for the claim and evidence sentence, respectively.

The resulting representation vectors $\mathbf{h}_c \in \mathbb{R}^d$ and $\mathbf{h}_s \in \mathbb{R}^d$ are then transformed by an MLP layer:

$$\mathbf{h}'_c = f_{act}(\mathbf{W}\mathbf{h}_c + \mathbf{b}), \quad \mathbf{h}'_s = f_{act}(\mathbf{W}\mathbf{h}_s + \mathbf{b}) \tag{9}$$

where f_{act} is the activation function, $\mathbf{W} \in \mathbb{R}^{d' \times d}$ is the weight matrix, $\mathbf{b} \in \mathbb{R}^{d'}$ is the bias vector, and d' is the dimension after transformation.

With a batch size of B, the transformed representations of all claims in the batch are represented as a matrix $\mathbf{H}_c = [\mathbf{h}'_{c,1}, \mathbf{h}'_{c,2}, \ldots, \mathbf{h}'_{c,B}]^T \in \mathbb{R}^{B \times d'}$, and the transformed representations of all evidence sentences as a matrix $\mathbf{H}_s = [\mathbf{h}'_{s,1}, \mathbf{h}'_{s,2}, \ldots, \mathbf{h}'_{s,B}]^T \in \mathbb{R}^{B \times d'}$.

The similarity matrix $\mathbf{S} = \mathbf{H}_c(\mathbf{H}_s)^T \in \mathbb{R}^{B \times B}$ is computed within the batch, and the model is trained using the SimCSE loss:

$$\mathcal{L}_{SimCSE} = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp(\mathbf{S}_{ii}/\tau)}{\sum_{j=1}^{B} \exp(\mathbf{S}_{ij}/\tau)} \tag{10}$$

Here, $\mathbf{S}_{ii}$ is the similarity between the i-th claim and its corresponding positive evidence sentence, $\mathbf{S}_{ij}(j \neq i)$ is the similarity between the i-th claim and the j-th evidence sentence (negative example), and τ is the temperature parameter.

3.3. Classification Task

3.3.1. Evidence Sentence Processing for Classification Model Training

Two primary approaches can be considered for training the classification model:

- Retrieving relevant evidence using a ranking algorithm like BM25 with the claim as a query, and then training the classification model using these results.
- Directly using the gold evidence corresponding to the claim to train the classification model.

While the former allows for automatic evidence acquisition, it carries the risk of retrieving irrelevant sentences, which could negatively impact classification performance. The latter approach is advantageous for leveraging highly reliable evidence. However, gold evidence is generally lengthy and not suitable for direct use in training a classification model. Therefore, we propose extracting crucial information from the gold evidence and transforming it into a format suitable for classification model training.

Given a claim $c_i \in \mathcal{C}$ ($\mathcal{C}$ is the claim space) and its corresponding gold evidence $G_i \in \mathcal{G}$ ($\mathcal{G}$ is the gold evidence space), an LLM-based crucial segment extraction function f_{extract} outputs a set of important evidence sentences $S^{\text{ext}} = \{s_1^{\text{ext}}, s_2^{\text{ext}}, \ldots, s_m^{\text{ext}}\} \in \mathcal{S}^{\text{ext}}$ ($\mathcal{S}^{\text{ext}}$ is the space of important evidence sentence sets):

$$f_{\text{extract}} : (\mathcal{C}, \mathcal{G}) \rightarrow \mathcal{S}^{\text{ext}} \tag{11}$$

The extraction process is achieved by providing a prompt $Prompt(c_i, G_i)$ as input to a function f_{LLM} corresponding to an LLM model:

$$S_i^{\text{ext}} = f_{\text{LLM}}(Prompt(c_i, G_i)) \tag{12}$$

Here, $S_i^{ext} = \{s_1^{ext}, s_2^{ext}, \ldots, s_m^{ext}\}$ is the set of important evidence sentences extracted by the LLM.

The prompt $P(c_i, G_i)$ is constructed by combining the claim c_i and the gold evidence G_i, taking the following form:

$$P(c_i, G_i) = \text{Template} \oplus c_i \oplus G_i \tag{13}$$

Here, $\oplus$ is the string concatenation operator, and Template is the prompt template specifying the extraction task. Using the extracted evidence sentence set S_i^{ext}, the classification model f_{cl} infers the predicted label l_i as follows:

$$f_{\text{cl}}(c_i, S_i^{\text{ext}}) = l_i \in \mathcal{L} \tag{14}$$

Prompt Template and LLM Details. For the extraction of crucial evidence sentences, we used the prompt template as shown in Figure 1. For all evidence extraction using an LLM (Large Language Model), we utilized the unsloth/Qwen3-8B-bnb-4bit model without employing Chain-of-Thought prompting.

```
Please output the following information in a bulleted list:

### Claim:
[claim]

### Document:
[gold evidence]

### Judgment:
Extract and concisely output only the direct evidence from the document needed
to determine if the claim is [label]. Do not include any unnecessary
explanations or analysis.

Output Example:
- result1
- result2
```

Figure 1: Prompt template for LLM-based evidence extraction

3.3.2. Data Augmentation for Improved Noise Robustness

Training the classification model solely on crucial information extracted by an LLM could lead to training with only correct positive examples. This raises concerns about its ability to effectively learn robustness against erroneous information, which is expected in real-world deployments. Therefore, we decided to intentionally inject irrelevant sentences during the training of the classification model.

For each claim c_i and its LLM-extracted evidence sentence set $S_i^{ext} = \{s_1^{ext}, s_2^{ext}, \ldots, s_m^{ext}\}$, we randomly inject irrelevant sentences as noise. Let $S^{all} = \bigcup_{j=1}^{n} S_j^{ext}$ be the union of all extracted evidence sentences across all claims. The set of noise candidates N_i for claim c_i is defined as follows:

$$N_i = S^{all} \setminus S_i^{ext} \tag{15}$$

Here, N_i is the set of evidence sentences extracted from the gold evidence of claims other than c_i. The noise injection function AddNoise is defined as:

$$\text{AddNoise}(S_i^{ext}, N_i, k_{noise}) = S_i^{noisy} \tag{16}$$

Here, S_i^{noisy} is generated by the following process:

$$S_i^{noisy} = S_i^{ext} \cup \text{RandomSample}(N_i, k_{noise}) \tag{17}$$

$\text{RandomSample}(N_i, k_{noise})$ is a function that randomly selects k_{noise} sentences from the noise candidate set N_i, where k_{noise} is a hyperparameter representing the number of sentences to be injected as noise. The final training evidence sentence set S_i^{train} is expressed as:

$$S_i^{train} = \text{AddNoise}(S_i^{ext}, N_i, k_{noise}) \tag{18}$$

Through this, we anticipate that the classification model will operate robustly even when noise is present during inference. The classification model f_{robust} is trained to minimize:

$$L(f_{\text{robust}}(c_i, S_i^{\text{train}}), l_i^{\text{gt}}) \tag{19}$$

where $L(\cdot, \cdot)$ represents the loss function, and l_i^{gt} represents the ground truth label. Here, cross-entropy loss was used as the loss function.

3.3.3. 4-Class Classification for Irrelevant Evidence Detection

During inference, a retrieval system might present evidence irrelevant to a given claim. To address such situations, we introduced a new label, "Irrelevant," to the classification model. We trained the model to categorize evidence sentences unrelated to the claim under this new label.

For the training data of the "Irrelevant" label, we used evidence sentences $N_i = S^{all} \setminus S_i^{ext}$ extracted from the gold evidence of other claims for each claim c_i. This means that an evidence sentence is defined as irrelevant to claim c_i if it was extracted from the gold evidence of a different claim c_j ($j \neq i$).

Extending the conventional 3-class classification, a 4-class classification function f_{irr} now outputs a label $y \in \mathcal{L}_{\text{irr}} = \{\text{True}, \text{False}, \text{Conflicting}, \text{Irrelevant}\}$:

$$f_{\text{irr}} : (\mathcal{C}, \mathcal{E}) \rightarrow \mathcal{L}_{\text{irr}} \tag{20}$$

This allows the model to identify irrelevant evidence even if appropriate evidence sentences are not retrieved. Consequently, it enables a strategy where the system can re-perform evidence retrieval and re-classify if irrelevant evidence is detected.

4. Experiments

4.1. Experimental Settings

For the claim classification task, the following settings were used for training and evaluation.

- **Model:** FacebookAI/roberta-base
- **Maximum sequence length:** 512
- **Number of labels:** Automatically determined from the data (e.g., Conflicting, False, True, etc.)

Training settings:

- Learning rate: 2×10^{-5}
- Batch size: 128 (training), 128 (evaluation)
- Number of epochs: 10
- Weight decay: 0.01
- Adam epsilon: 1×10^{-8}
- Scheduler: linear
- Warmup ratio: 0.1

All experiments were conducted using a Tesla V100-PCIE-32GB GPU.

4.2. Evidence Retrieval

We evaluated and compared three algorithms for retrieving evidence sentences relevant to claims: BM25, Contriever, and Contriever^{FT} (additional training with SimCSE). Table 1 presents the results.

The fact that the three models showed similarly high performance in P@1 to P@3 indicates that when clear, relevant evidence sentences exist for numerical claims, the differences between retrieval methods are limited. Contriever^{FT} demonstrated significant performance improvements, particularly from P@10 onwards and in Recall metrics, achieving a substantial increase to 0.524 for Recall@100 and 0.731 for Recall@1000. This is likely due to the SimCSE-based contrastive learning enabling more effective learning of semantic relevance between claims and gold evidence. While BM25 showed excellent performance in top-tier precision, it lagged behind other methods in Recall metrics.

For fact-checking tasks, it is considered crucial to collect a wide range of diverse evidence sentences. Therefore, Contriever^{FT}, with its high Recall performance, proved to be the optimal choice. On the other hand, BM25 can still be a viable option when computational resources are limited or when only the highest-ranked evidence is required.

Table 1
Comparison of Ranking Methods for Evidence Retrieval

Model	P@1	P@2	P@3	P@10	Recall@10	Recall@100	Recall@1000
BM25	0.942	0.895	0.829	0.534	0.205	0.355	0.502
Contriever	0.925	0.885	0.832	0.587	0.227	0.423	0.592
Contriever^{FT}	0.926	0.895	0.858	0.656	0.255	0.524	0.731

4.3. Classification for Fact-Checking

Table 2 presents the results of the fact-checking classification using the development data. Despite Contriever^{FT} demonstrating high accuracy in the evidence retrieval results (Table 1), it achieved the lowest macro F1 in the classification task. This clearly indicates that improvements in retrieval performance do not necessarily translate directly to enhanced classification performance. SimCSE-fine-tuned Contriever improved recall by including a greater number of relevant evidence sentences in the retrieval results. However, the precision at top ranks (e.g., top-1 or top-3) did not sufficiently improve, and thus this did not lead to better overall classification performance. This suggests that while the fine-tuned Contriever is effective at broadly collecting semantically related sentences, it is less effective at ranking the most crucial evidence at the top. Therefore, a two-stage retrieval approach—first using Contriever for initial retrieval to gather a wide range of candidates, followed by a reranking model to place the most relevant evidence at higher ranks—would likely be more effective. BM25-based retrieval achieved the most stable classification performance, proving to be the optimal choice from a practical perspective.

Contrary to expectations, the noise augmentation method led to a performance decrease, particularly a significant drop in Conflicting predictions. This phenomenon can be attributed to the model's tendency, after being exposed to irrelevant sentences during training, to classify ambiguous or weakly supported cases as *True* or *False* rather than *Conflicting*. By learning to make predictions even in the presence of noise, the model becomes less sensitive to ambiguity and is more likely to output a definitive label. As a result, the recall and F1 score for the *Conflicting* class decreased, while misclassifications into the *True* or *False* classes increased. However, a slight improvement was observed for True predictions, partially confirming the effect of improved noise robustness. These findings suggest that noise injection can enhance robustness to irrelevant information while also blurring the criteria for identifying ambiguous cases in the model. In the future, further improvements are needed, such as optimizing data augmentation and loss function design, to enhance robustness against irrelevant information while more accurately identifying ambiguous cases.

Table 2
Fact-Checking Classification Results

Method	macro F1	True F1	Conflicting F1	False F1
Direct Gold Evidence + BM25 Retrieval	0.3919	0.4148	0.0562	0.7045
LLM Evidence Extraction + BM25 Retrieval	0.5054	0.3233	0.4128	0.7803
LLM Evidence Extraction + Contriever Retrieval	0.4925	0.3103	0.3886	0.7787
LLM Evidence Extraction + Contriever^{FT} Retrieval	0.4004	0.0931	0.6998	0.4082
LLM Evidence Extraction + Noise Augmentation + BM25 Retrieval	0.4271	0.3403	0.1690	0.7720

4.4. Irrelevant Evidence Detection

Table 3 shows the results for the 4-class classification model designed for irrelevant evidence detection. The model exhibited a tendency to classify almost all claims as irrelevant, indicating that it was not appropriately trained.

Table 3
Irrelevant Evidence Detection Results

Method	macro F1	True F1	Conflicting F1	False F1	Irrelevant F1
LLM Evidence Extraction + BM25 Retrieval	0.1669	0.0000	0.0000	0.1667	0.6665

5. Conclusion

In this paper, we proposed and validated a method following a two-stage approach for evidence retrieval and classification in fact-checking numerical claims. In the evidence retrieval task, Contriever^{FT}, further trained with SimCSE-based contrastive learning, achieved substantial performance improvements, particularly in Recall metrics, demonstrating its ability to effectively learn the semantic relevance between claims and gold evidence. Meanwhile, BM25 maintained stable performance in top-tier precision, confirming its practicality from a computational efficiency perspective.

However, a crucial insight gained from the classification task was that high accuracy in evidence retrieval does not necessarily directly lead to improved classification performance. The classification model using BM25-based search results achieved the most stable macro F1, indicating its optimality from a practical standpoint. Class-wise analysis revealed that False predictions consistently had the highest F1 score across all methods, while True predictions proved to be the most challenging. Although the noise augmentation method unexpectedly led to an overall performance decrease, a slight improvement was observed for True predictions, suggesting potential for improved noise robustness.

As future work, we aim to build a two-stage retrieval system that leverages the high Recall performance of Contriever^{FT}. By applying re-ranking techniques to comprehensively retrieved candidate documents, we expect to achieve performance improvements that balance both retrieval coverage and ranking accuracy, by placing more relevant evidence sentences higher in the results.

Acknowledgments

A part of this work was supported by JSPS KAKENHI Grant Number 23K11342.

Declaration on Generative AI

During the preparation of this work, the author utilized Gemini for revisions related to grammar and clarity. These tools were employed to refine sentence structure, correct typographical errors, and enhance the overall quality of the language. They were also used for translating content into English. No generative content was used in the analysis, figures, or experimental sections. After using these tools/services, the author reviewed and edited the content as needed and assumes full responsibility for the content of this publication.

References

[1] Z. Guo, M. Schlichtkrull, A. Vlachos, A survey on automated fact-checking, Transactions of the Association for Computational Linguistics 10 (2022) 178–206.

[2] V. Venktesh, V. Setty, A. Anand, M. Hasanain, B. Bendou, H. Bouamor, F. Alam, G. Iturra-Bocaz, P. Galuščáková, Overview of the CLEF-2025 CheckThat! lab task 3 on fact-checking numerical claims, in: G. Faggioli, N. Ferro, P. Rosso, D. Spina (Eds.), Working Notes of CLEF 2025 - Conference and Labs of the Evaluation Forum, CLEF 2025, Madrid, Spain, 2025.

[3] M. Mori, P. Papotti, L. Bellomarini, O. Giudice, Neural machine translation for fact-checking temporal claims, in: R. Aly, C. Christodoulopoulos, O. Cocarascu, Z. Guo, A. Mittal, M. Schlichtkrull, J. Thorne, A. Vlachos (Eds.), Proceedings of the Fifth Fact Extraction and VERification Workshop

(FEVER), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 78–82. URL: https://aclanthology.org/2022.fever-1.8/. doi:10.18653/v1/2022.fever-1.8.

[4] J. Chen, A. Sriram, E. Choi, G. Durrett, Generating literal and implied subquestions to fact-check complex claims, in: Y. Goldberg, Z. Kozareva, Y. Zhang (Eds.), Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 3495–3516. URL: https://aclanthology.org/2022.emnlp-main.229/. doi:10.18653/v1/2022.emnlp-main.229.

[5] I. Augenstein, C. Lioma, D. Wang, L. Chaves Lima, C. Hansen, C. Hansen, J. G. Simonsen, MultiFC: A real-world multi-domain dataset for evidence-based fact checking of claims, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 4685–4697. URL: https://aclanthology.org/D19-1475/. doi:10.18653/v1/D19-1475.

[6] A. Sathe, S. Ather, T. M. Le, N. Perry, J. Park, Automated fact-checking of claims from Wikipedia, in: N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis (Eds.), Proceedings of the Twelfth Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2020, pp. 6874–6882. URL: https://aclanthology.org/2020.lrec-1.849/.

[7] M. Schlichtkrull, Z. Guo, A. Vlachos, Averitec: A dataset for real-world claim verification with evidence from the web, 2023. URL: https://arxiv.org/abs/2305.13117. arXiv:2305.13117.

[8] V. V, A. Anand, A. Anand, V. Setty, Quantemp: A real-world open-domain benchmark for fact-checking numerical claims, 2024. URL: https://arxiv.org/abs/2403.17169. arXiv:2403.17169.

[9] S. Robertson, H. Zaragoza, et al., The probabilistic relevance framework: Bm25 and beyond, Foundations and Trends® in Information Retrieval 3 (2009) 333–389.

[10] G. Izacard, M. Caron, L. Hosseini, S. Riedel, P. Bojanowski, A. Joulin, E. Grave, Unsupervised dense information retrieval with contrastive learning, arXiv preprint arXiv:2112.09118 (2021).