CSIRO-LT at Context24: Contextualising Scientific Figures and Tables in Scientific Literature

Necva Bölücü, Vincent Nguyen, Roelien C. Timmer, Huichen Yang
Maciej Rybinski, Stephen Wan, Sarvnaz Karimi
[1]CSIRO Data61, Australia
firstname.lastname@csiro.au

Abstract

Finding evidence for claims from content presented in experimental results of scientific articles is difficult. The evidence is often presented in the form of tables and figures, and correctly matching it to scientific claims presents automation challenges. The Context24 shared task is launched to support the development of systems able to verify claims by extracting supporting evidence from articles. We explore different facets of this shared task modelled as a search problem and as an information extraction task. We experiment with a range of methods in each of these categories for the two subtasks of evidence identification and grounding context identification in the Context24 shared task.

1 Introduction

Finding evidence relating to scientific claims in research articles is a time-consuming task. These claims can be supported by the article's text, figures or tables. The *Context24* international shared task[1], organised as part of the *Scholarly Document Processing* workshop at the ACL 2024 Conference, set a challenge for developing methods for two tasks: (1) *Evidence Identification*, where given a scientific claim and a relevant research paper, participating systems should identify key figures or tables from the article that provide supporting evidence for the claim; and, (2) *Grounding Context Identification* where given a scientific claim and a relevant research article, participating systems should identify all grounding contexts that discuss the experiment that led to the claim. The *grounding context* can be found in different research article sections, such as in the results sections, including figures and tables.

Our team participated in both tasks with methods drawing from information retrieval and natural language processing (NLP) research, including statistical ranking models, dense retrieval and the use of large language models (LLMs), including multimodal LLMs.

2 Background and Related Work

Searching for Evidence In 2020, when the COVID-19 pandemic hit the world, the information retrieval community through TREC, established an international shared task called TREC-COVID (Roberts et al., 2021) in response to the overwhelming amount of publications on COVID-19 emerging. This task was designed to research search algorithms to find scientific evidence on what COVID-19 is, what are the symptoms of the disease, and what preventive methods are approved by the public health authorities (e.g., mask-wearing, what type of mask). Due to the public interest in this, there was a need to access reputable information and sift through misleading information (i.e., information without any scientific basis). While this task was largely tackled as an information retrieval problem known as *pandemic information retrieval* (Nguyen et al., 2020), it paved the way for further research in the NLP community. Some data have been proposed in the scientific domain as well, such as SciFact-Open (Wadden et al., 2022a).

Fact Checking and Fake News Detection News is the most common way to disseminate information rapidly. However, fake news— characterised by low-quality, intentionally misleading information— poses significant threats by distorting public perception and causing negative impacts on both individuals and society at large (Shu et al., 2017). Addressing the problem has become imperative, prompting researchers to focus on the automatic verification of facts and detection of fake news.

Numerous studies and public datasets have been developed for the task, covering various domains

[1] https://sdproc.org/2024/sharedtasks.html#context24

Proceedings of the Fourth Workshop on Scholarly Document Processing (SDP 2024), pages 314–323
August 16, 2024 ©2024 Association for Computational Linguistics

Dataset	Domain	Task 1		Task 2	
		Training	Test	Training	Test
akamatsulab	Cell biology	213	51	28	49
BIOL403	Cell biology	60	—	—	—
dg-social-media-polarization	Social sciences (political science, economics, HCI)	78	—	—	—
megacoglab	Various (HCI, psychology, economics, CS, public health)	123	60	14	60
Total		474	111	42	109

Table 1: Dataset specifications. Sizes in the last four columns represent the number of claims in each dataset.

and applications. AVERITEC (Schlichtkrull et al., 2024) and MultiFC (Augenstein et al., 2019) are examples of such datasets. To increase the popularity of the task and the development of effective systems, shared tasks such as Fact Extraction and VERification (FEVER) (Thorne et al., 2018) have been organised. FEVER focuses on verifying factoid claims using evidence from Wikipedia and also introduced new methods, including machine learning algorithms for fact verification (Ahmad et al., 2020; Barnabò et al., 2023; Nie et al., 2019; Sahoo and Gupta, 2021), the application of LLMs for detecting fake news (Hu et al., 2024), and fact-checking using multi-modality sources such as images (Fung et al., 2021; Jindal et al., 2020) and videos (Micallef et al., 2022).

Scientific Claim Verification Evaluating the validity of claims in scientific texts is a specialised area of research essential due to the rapid increase in scientific publications. This task aims to determine whether proposed scientific claims are supported or refuted by retrieving relevant evidence from scientific documents, which may include sentences, figures, or tables. Wadden et al. (2020) introduce the task with the SciFACT dataset, consisting of scientific claims with corresponding abstracts that either support or refute the claim, with a baseline method for the task. The success of retrieval-based approaches has led to the development of document-level (Pang et al., 2020) and sentence-level (Soleimani et al., 2020) methods for the task. With advancements in LLMs, new systems on this task employ LLMs for document retrieval (Pradeep et al., 2021a,b), multi-task label prediction for scientific fact verification (Li et al., 2021; Wadden et al., 2022b), and automatically generating verifiable scientific claims and fact-checking (Wright et al., 2022; Singh et al., 2024).

3 Dataset

The claims and papers used in the Context24 shared task come from four separate datasets, each of which comes from a different set of research domains. Table 1 shows a breakdown of the claims per dataset for both sub-tasks. For Task 1, the number of claims is 474 for training and 111 for testing, and for Task 2, it is 42 for training and 109 for testing. Task 1 involves 193 articles in the training set and 46 in the testing set, while Task 2 includes 31 training articles and 44 testing articles, demonstrating that an article may contain one or more claims.

Silver Dataset annotation In addition to the annotated data, we constructed a silver-annotated dataset utilising a silver data corpus containing 17,007 (unannotated) articles provided by the shared task organisers. We used the GPT3.5 model fine-tuned with the training set of Task 2 (see Section 4.2) to automatically generate claims and corresponding evidence for each of the articles (as opposed to generating evidence, given an article and a claim, which was the objective of task 2). Similar to the main dataset, the silver-annotated dataset includes multiple claims per article.

4 Methods

4.1 Task 1: Evidence Identification

As mentioned above, this task requires a ranked list of relevant key figures and tables based on a given scientific claim and a research paper. All images of figures and tables alongside the article's full text are provided.

We tackle evidence identification as an information retrieval task. As described below, we experiment with multiple methods of document preprocessing, representation, indexing, and ranking.

Article Chunking We experiment with three approaches to splitting full-text articles into fragments. We use sentence splitting, fixed-size chunking, and semantic splitting using `semantic_splitter` Python library. The intention is to capture the articles' text content at different granularity levels.

Indexing Logic Our experiments cover two separate base approaches to creating indices. In the first one (henceforth referred to as 'element-guided'), we directly index representations of tables and figures aggregated from article chunks (see previous paragraph) that cite them. We detect table and figure citations using a simple rule-based approach that searches the chunks for specific expressions, such as: 'Table NUM', 'Figure NUM', 'Tab. NUM', etc. For example, to represent FIG 1 in a specific article, we concatenate all citing chunks (i.e., those containing mentions of 'Figure 1', 'Fig. 1', 'FIG 1', etc.) found in this article. This concatenated text becomes a *document* in our index. We then rank the evidence (so, tables and figures) using the claim's text as a query and calculating the relevance of the *documents* (so, representations of tables and figures) to the query using a specific method (as discussed further in this section). We create a single index for all the evidence across the entire article corpus, which is important for indexing methods that use corpus statistics for relevance calculations (in particular, inverted indices with BM25 relevance).

In our second approach (henceforth referred to as 'chunk-guided'), articles are first chunked at specific levels of granularity (sentences, paragraphs, and chunks of 128, 256, 512, and 1024 tokens). These chunks are indexed in separate indices using Elasticsearch, that is, sentences and paragraphs, and each window-size chunk results in a separate index of chunks. In this setup, the claims are used as queries to rank the snippets of the corresponding document. This produces a ranked list of snippets. We then extract table and figure mentions from the ranked elements of this list. We contemplate three evidence-ranking methods for the chunk-guided approach:

1. Since each snippet may reference more than one figure or table, we score the tables and figures using reciprocal rank fusion – we sum the inverse rank score of all citing snippets for every table and figure. We refer to this approach as 'rank fusion scoring'.
2. One could reasonably assume that relevant evidence should be mentioned in one of the most relevant paragraphs. We, therefore, concatenate the snippets from the relevance-ranked list and rank the tables and figures by the order of mentions in this concatenated text.
3. We delegate the task of producing the ranked list of evidence to an LLM (Llama 3 70B); the ranked list of snippets is included in the LLM prompt together with the claim text.

Indexing and retrieval methods For the element-guided approach, we evaluate two indexing and relevance scoring models. Our baseline experiments use a traditional inverted index with *BM25* relevance scoring. We also evaluate an embedding-based approach, where the evidence representations and the claim are vectorised using a *Mistral E5* embedding model, and relevance is then calculated as a cosine similarity of the embedding vectors.

For the experiments with the chunk-guided approach, we have used a *hybrid index* combining BM25 relevance and embeddings, implemented using DeBIR (Nguyen et al., 2023).[2] In this setup, we evaluate two universal embedding models: Mistral E5 and BGE v1.5 Large.

Image Retrieval Experiment We use a multimodal embedding model CLIP (Radford et al., 2021)– which is OpenAI Contrastive Learning In Pretraining—to generate embedding from images and claim texts. The evidence in this experiment is ranked directly by relevance, i.e., the cosine similarity between the text (claim) embedding and image (figure or table) embedding.

Use of Captions We also carry out some experiments that leverage text extracted with OCR from caption images. In the element-guided approach, we would append the corresponding caption to the (concatenated) textual representation of a specific figure or table. In the chunk-guided approach, each caption is considered a separate document for retrieval.

Use of a LLM to Rewrite the Evidence Representations We also present experiments with the reformulation of the table and figure textual

[2] Preliminary testing on the training set showed that BM25 significantly outperforms embedding-only approaches. We, therefore, only compared against BM25 as we are limited by the number of runs during the shared task.

Run Name (as submitted)	Description
Top-k merge	Round-robin ranking fusion of all our runs except near identical and other round-robin mergers
RS Merge w/ Rank s=2	Reciprocal ranking fusion of all runs, except other round-robin mergers, where the ranking of the run is also considered for fusion.
top-k merge all-runs	Round-robin ranking fusion of all our runs except other round-robin mergers
E5 on LLM-rephrased descriptions	Element-guided Mistral E5 indexing with LLM-rephrased descriptions (original chunk size of 1500 characters with semantic chunking)
E5	Element-guided Mistral E5 indexing with a chunk size of 100 tokens
Hier-Caption+E5+DeBEIR	Hierarchical retrieval approach over all granularities using the DeBEIR scorer with E5 embeddings and rank-score fusion. Captions are excluded from retrieval.
e5+HierAll+DeBEIR+RS	Same as above, except captions, are also considered.
e5+HierCapSent512+DeBeIR	Similar to the above, however, only the captions, sentences and chunk size of 512 are considered for retrieval.
E5 w. semantic chunking	Element-guided Mistral E5 indexing with semantic chunking with a chunk size of 1000 characters
Plain BM25 100 word chunks	Element-guided inverted index with BM25 ranking and 100-token chunking
BM25 (Articles as Document) + UM	Chunk-guided retrieval of documents (chunk size of 512) using unique merging.
BM25 (Articles as Document) + RS	Chunk-guided retrieval of documents (chunk size of 512) using reciprocal rank-score fusion and BM25 relevance scoring.
BGE + DeBEIR + 512 + RS	Chunk-guided retrieval of documents (chunk size of 512) using reciprocal rank-score fusion and BGE embedding model and cosine similarity.
BM25 'element-as-document'	Element-guided inverted index with BM25 ranking and semantic chunking with chunk size of 1000 characters
BM25 with citing contexts and captions	Element-guided inverted index with BM25 ranking and 100-token chunking, with captions
BM25 on LLM-rephrased descriptions	Element-guided inverted index with BM25 with LLM-rephrased descriptions
Llama3-70b+figures merging	Using the articles retrieved with E5 embeddings, DeBEIR and a chunk size of 512. Llama3 was used to extract and order elements.
Image_embedding (BaselLine)	Text-to-image retrieval with CLIP

Table 2: Summary of submitted runs for Task 1 (evidence identification).

representations in our element-guided approach. We used a Mistral-7B-instruct-v0.2 model with a prompt asking what is presented in a specific figure given its text representation. That is, it cites chunks from the scientific article. The index would then be created using LLM-generated texts (i.e., the reformulations).

Hierarchical approach and ranking fusion experiments Finally, our submissions also include approaches that combine different ranking methods. In particular, we experiment with a *hierarchical* approach where the claim is used to query chunk-guided indices at all levels of granularity and the final result is aggregated using reciprocal rank fusion from the resulting rankings (of different granularities).

We further explore the idea of fusing signals from different ranking systems with a simple *round-robin* approach that merges runs by appending unique results from a specific top portion of each individual ranking, with the top portion growing each of the round-robin passes.

Although the hierarchical and round-robin merged runs employ a similar principle, the intuition behind them is slightly different. With the hierarchical runs, we investigate merging results at different granularities, but we still use a single retrieval approach across the merged runs. With the round-robin merge, the intention is to combine high-confidence results from a reasonably diverse set of techniques.

Submitted Runs In Table 2, we summarise the runs we submitted for evaluation, with a short description of each run. Each description conveys how the run relates to the methods presented above.

4.2 Task 2: Grounding Context Identification

This task requires systems to generate a list of text fragments, based on a specific research article, that provides information on a given scientific claim. The approaches we experiment with for this task are described below.

Retrieval-based Approach In our approach, we employ *BM25* as a retrieval model in various configurations. The documents are segmented into sentences, and the query is the provided claim. For a given article, we retrieve n sentences that are most relevant to the given claim. Different systems

Run Name (as submitted)	Description
rule_based_claude_postprocess	Retrieval with rule-based filtering with Claude LLM to generate evidence with post-processing
gpt_clean_postprocess	LLM (GPT4) zero-shot setting with post-processing
rule-based	Retrieval with rule-based filtering approach
bm25_5_llama3	Retrieval-based approach using retrieval+LLM using a combination of BM25 and Llama3 LLM
claude_zero_shot	LLM (Claude) zero-shot setting
llama3_full_article	LLM (Claude) zero-shot setting
bm25_clean_page	Retrieval-based approach with collected documents
claude_clean	LLM (Claude)zero-shot setting with collected documents
gpt_finetuned_clause	LLM (GPT3.5) fine-tuned setting at clause level
bm25_5+claude	Retrieval-based approach using a combination of BM25 and Claude LLM
debier_256	Retrieval-based approach with a chunk size of 256 using Debier as the retrieval method
gpt_sentence	LLM (GPT4) zero-shot setting at clause level instead of sentence level
bm25_5_left_neighbor	Retrieval-based approach using retrieval+heuristic combination left sentences are selected as neighbour
claude_prompt (who, how, what)	LLM (Claude) zeros-hot setting with prompt using the definition of the shared task for Task 2
llama3-8b ft 5epoch lre-6	LLM (Llama3-8B) fine-tuned setting on the training set of the shared task
sft llama3-8b silver data	LLM (Llama3-8B) fine-tuned setting on silver dataset

Table 3: Summary of submitted runs for Task 2 (Grounding Context Identification).

that leverage the retrieval model are listed below:

- **Simple retrieval:** We apply BM25 directly to the article, which has been split into sentences, to retrieve n sentences as evidence. This method serves as a straightforward way to identify evidence as the most relevant sentences (based on the BM25 scoring).

- **Retrieval + LLM:** In this approach, BM25 is used to retrieve the most promising candidate sentences from the article, narrowing down the content before it is processed by an LLM. Instead of using the entire article in the prompt for the LLM, we only provide the sentences retrieved by the BM25 method. This can improve the efficiency and focus of the LLM's processing by concentrating on the most relevant sentences.

- **Heuristic:** Here, we augment the retrieval results by concatenating adjacent (neighbour) sentences to those retrieved by BM25, thus using them as additional evidence. This method enhances comprehension and topic continuity by using neighbour sentences to provide additional context.

- **Retrieval with rule-based filtering:** Upon inspecting the training set we observe that evidence fragments often share a common writing style and, consequently, contain similar patterns. In this approach, we build a list of *common expressions*—such as 'assigned to', 'of all prior', and 'in the absence of'—using the ground truth grounding contexts. For each ground truth context, we include the longest sub-sequence of tokens (which includes at least one verb, adjective, noun, or noun phrase) shared by at least one other ground truth context. At inference, we filter BM25 results obtained with a *simple retrieval* using the list of common expressions. That is, we discard candidate sentences that do not contain any of the common expressions.

LLM In addition to the approaches utilising the retrieval method, we also investigate the use of LLMs in zero-shot, few-shot and fine-tuning scenarios.

Zero- and few-shot learning leverages the LLM's pre-trained knowledge to perform tasks without additional task-specific training (Brown et al., 2020; Radford et al., 2019). For this task, we provide the article and claim in the prompt to extract evidence in zero-shot learning and we select k in-context samples from the train set in few-shot learning. Fine-tuning involves adapting a pre-trained LLM to the task using the provided annotated dataset. We finetune two different language models: Llama3 8B (AI@Meta, 2024)[3] and GPT-3.5.

Submitted Runs Table 3 summarises the runs we submitted for evaluation, with a short description

[3] https://github.com/meta-llama/llama3

of each run. Each description conveys how the run relates to the methods presented above.

5 Experimental Setup

We use the default b and $k1$ parameter values in BM25 implementation, utilising rank_bm25 library for the element-guided approach in Task 1 and the retrieval-based approach in Task 2, and Elasticsearch for the chunk-guided approach in Task 1.

In Task 2, we use two different tools to identify sentences in the full-text articles: NLTK (Bird and Loper, 2004) and SpaCy (Honnibal and Montani, 2017), with the best results reported. We use the PoS tagger of the NLTK library for the retrieval with the rule-based filtering approach proposed for Task 2. We retrieved the original pdf files from the publishers using DOIs in the original articles. This led to a cleaner version of the document collection. It also provided us with information missing from the articles, such as the page numbers.

In Task 1, we use the following instructions (prompts) to create E5 query embeddings for element-, and chunk-guided approaches, respectively:

- Given a claim, retrieve descriptions of figures and tables that support the claim
- Given a claim, retrieve documents that contain references to figures and tables that support the claim

In Task 2, we use the following prompt for zero-shot experiments of LLMs:

- Please extract evidence clauses from the given article for the given claim CLAIM [Given Article Start] FULL TEXT [Given Article End]

For fine-tuning the OpenAI GPT model in Task 2, we use the provided API (OpenAI, 2023). The steps applied before fine-tuning are the preparation of the training data and its upload to the OpenAI servers. The format of the training data is JSONL. One sample template of the training data is given below:

```
"messages": [
{"role": "system",
 "content":
    FULL TEXT},
{"role": "user", "content":
 "What are evidences for the
  given claim:" CLAIM},
{"role": "assistant",
 "content": EVIDENCE}
]
```

We use the default hyperparameters of the API for fine-tuning. After fine-tuning the GPT3.5 model, we use the following prompts for inference:

- What are evidences for the given claim: CLAIM
- What are evidence clauses for the given claim: CLAIM

For our experiments with LLaMA-3 8B[4], we fine-tune the model on the training data and silver dataset separately for Task 2. Given the relatively small size of the training dataset.

We eliminate the introductory sentences from LLM outputs (e.g., 'Here are two relevant evidence sentences from the given article:') using pattern-based matching with `colon` (:) as postprocessing in Task 2.

Metrics For both sub-tasks, we follow the evaluation procedure put in place by the shared task organisers. For Sub-task 1 a typical ranking metric of NDCG (Järvelin and Kekäläinen, 2002) at 5 and 10 is used. For the grounding sub-task, BERT score (Zhang et al., 2020) and ROUGE (Lin, 2004) variants are reported to measure the overlap between the identified context and the ground truth.

6 Experimental Results

6.1 Task 1: Evidence Identification

The test set results of our evidence identification experiments are shown in Table 4.

Our best single-model runs are element-guided E5 ('E5', nDCG@5 of .55) and element-guided E5 with LLM-reformulated texts ('E5 on LLM-rephrased descriptions', nDCG@10 of 0.60). Both runs used fixed-sized chunking with a window size of 100 tokens. While the method outperforms other single model runs, our experiments indicate that it is sensitive to prompt used for query embedding (approximately -10% on development set) and to chunking strategy (see 'E5 w. semantic chunking' run, we also observed similar trend on development set).

[4] We use the code of llama-recipes https://github.com/meta-llama/llama-recipes and models hosted on Hugginface https://huggingface.co/meta-llama/Meta-Llama-3-8B.

Run Name	nDCG@5	nDCG@10
Top-k merge	0.5542	0.6160
RS Merge w/ Rank s=2	0.5598	0.5942
top-k merge all-runs	0.5464	0.6005
E5 on LLM-rephrased descriptions	0.5464	0.6005
E5	0.5542	0.5894
Hier-Caption+E5+DeBEIR	0.5309	0.5654
e5+HierAll+DeBEIR+RS	0.5251	0.5653
e5+HierCapSent512+DeBeIR	0.5116	0.5560
E5 w. semantic chunking	0.4809	0.5304
Plain BM25 100 word chunks	0.4665	0.5195
BM25 (Articles as Document) + UM	0.4636	0.5199
BM25 (Articles as Document) + RS	0.4681	0.5129
BGE + DeBEIR + 512 + RS	0.4639	0.5145
BM25 'element-as-document'	0.4597	0.5183
BM25 with citing contexts and captions	0.4622	0.5098
BM25 on LLM-rephrased descriptions	0.4552	0.5092
Llama3-70b+figures merging	0.4625	0.4954
Image_embedding(BaselLine)	0.2954	0.3681

Table 4: Evidence identification task results based on NDCG.

Run Name	BERT Score	ROUGE-1	ROUGE-2	ROUGE-L
rule_based_claude_postprocess	0.8577	0.3773	0.1955	0.2716
gpt_clean_postprocess	0.8525	0.3663	0.1968	0.2544
rule-based	0.8651	0.3485	0.1782	0.2731
bm25_5_llama3	0.8499	0.3431	0.1950	0.2718
claude_zero_shot	0.8530	0.3686	0.1812	0.2552
llama3_full_article	0.8569	0.3420	0.1781	0.2492
bm25_clean_page	0.8486	0.3462	0.1800	0.2284
claude_clean	0.8492	0.3551	0.1622	0.2367
gpt_finetuned_clause	0.8510	0.3301	0.1865	0.2330
bm25_5+claude	0.8471	0.3443	0.1768	0.2257
debier_256	0.8531	0.3426	0.1589	0.2385
gpt_sentence	0.8477	0.3122	0.1668	0.2114
bm25_5_left_neighbor	0.8411	0.2977	0.1617	0.1920
claude_prompt (who, how, what)	0.7877	0.2941	0.0877	0.1650
sft llama3-8b silver data	0.7751	0.2093	0.0394	0.1325
llama3-8b ft 5epoch lre-6	0.7743	0.2077	0.0398	0.1296

Table 5: Grounding Context Identification task results based on BERT Score.

We have observed some improvement over single system runs with our fused runs. More specifically hierarchical runs outperform single-granularity counterparts. Also, our best overall scores are attained by the round-robin merged runs (although, admittedly, the improvements over the best single model E5 runs used in this scenario are relatively small).

All of our BM25 runs scored close to 0.46 and 0.51 for nDCG@5 and nDCG@10, respectively. This is a somewhat surprising result, as we have observed much more pronounced differences in the effectiveness of different BM25 flavours on the training set.

Ranking evidence directly using an LLM yielded scores lower than most of the BM25 baselines (slightly). Our experiment with image and text embeddings resulted in significantly lower scores.

6.2 Task 2: Grounding Context Identification

Table 5 demonstrates the performance of our methods on the test set of Task 2.

We achieve the highest BERT Score with the retrieval with the rule-based filtering approach (‘rule_based’, BERTScore of 0.87) among all methods. Most methods yield BERT scores ranging between 0.84 and 0.87, except for the fine-tuning LLM (Llama3-8B). Our best runs are based on retrieval-based approach which are retrieval with rule-based filtering (‘rule_based’, ROUGE-L of 0.27), retrieval with a combination of rule-based filtering and LLM (‘rule_based_claude_postprocess’, ROUGE-L of 0.27) and retrieval + LLM (‘bm25_5_llama3’, ROUGE-L of 0.27) in terms of ROUGE-L score. This indicates that selecting evidence from the article enhances semantic similarity and coherence across longer text sequences.

An important observation is that the retrieval-based approach results achieve comparable or better BERT scores to LLMs (e.g., claude_clean and gpt_sentence). This suggests that the extractive approach (BM25) remains superior for the task due to its simplicity, despite LLMs’ remarkable performance in other NLP tasks.

Combining the retrieval method with LLMs, named Retrieval + LLM (e.g., bm25_5+claude and bm25_5_llama3) improves ROUGE scores by focusing the grounding context on relevant fragments only. However, in terms of ROUGE scores, LLM methods with zero-shot settings (gpt_clean_postprocess and claude_zero_shot), as well as the Retrieval + LLM approach (rule_based_claude_postprocess) outperform other methods.

LLMs are sensitive to provided prompts in zero-shot settings (e.g., claude_clean and claude_prompt (who, how, what)). Additionally, post-processing significantly influences ROUGE scores by removing irrelevant parts of generated evidence, as ROUGE evaluates the overlap of n-grams between generated evidence and human-annotated references.

7 Conclusions

We participated in the Context24 Shared Task for multimodal evidence and grounding context identification for scientific claims. There were two subtasks for evidence identification and grounding evidence identification. For the former, we treated it as an information retrieval task where both statistical and dense retrieval methods were investigated. We obtained promising results with the Mistral E5 dense retrieval model and with our ranking fusion experiments. For the second subtask, our best method used a filtering mechanism where frequent expressions were identified from the training data. This method yielded higher results than those we obtained using generative models.

References

Iftikhar Ahmad, Muhammad Yousaf, Suhail Yousaf, and Muhammad Ovais Ahmad. 2020. Fake news detection using machine learning ensemble methods. *Complexity*, 2020(1):8885861.

AI@Meta. 2024. Llama 3 model card.

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.

Giorgio Barnabò, Federico Siciliano, Carlos Castillo, Stefano Leonardi, Preslav Nakov, Giovanni Da San Martino, and Fabrizio Silvestri. 2023. Deep active learning for misinformation detection using geometric deep learning. *Online Social Networks and Media*, 33:100244.

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages

214–217, Barcelona, Spain. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Yi Fung, Christopher Thomas, Revanth Gangi Reddy, Sandeep Polisetty, Heng Ji, Shih-Fu Chang, Kathleen McKeown, Mohit Bansal, and Avirup Sil. 2021. Infosurgeon: Cross-media fine-grained information consistency checking for fake news detection. 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 1683–1698.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1):411–420.

Beizhe Hu, Qiang Sheng, Juan Cao, Yuhui Shi, Yang Li, Danding Wang, and Peng Qi. 2024. Bad actor, good advisor: Exploring the role of large language models in fake news detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 22105–22113.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446.

Sarthak Jindal, Raghav Sood, Richa Singh, Mayank Vatsa, and Tanmoy Chakraborty. 2020. Newsbag: A multimodal benchmark dataset for fake news detection. In *CEUR Workshop Proc*, volume 2560, pages 138–145.

Xiangci Li, Gully A Burns, and Nanyun Peng. 2021. A paragraph-level multi-task learning model for scientific fact-verification. In *SDU@ AAAI*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Nicholas Micallef, Marcelo Sandoval-Castañeda, Adi Cohen, Mustaque Ahamad, Srijan Kumar, and Nasir Memon. 2022. Cross-platform multimodal misinformation: Taxonomy, characteristics and detection for textual posts and videos. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 16, pages 651–662.

Vincent Nguyen, Sarvnaz Karimi, and Zhenchang Xing. 2023. DeBEIR: A python package for dense bi-encoder information retrieval. *Journal of Open Source Software*, 8(87):5017.

Vincent Nguyen, Maciek Rybinski, Sarvnaz Karimi, and Zhenchang Xing. 2020. Pandemic literature search: Finding information on COVID-19. In *Proceedings of the 18th Annual Workshop of the Australasian Language Technology Association*, pages 92–97.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining fact extraction and verification with neural semantic matching networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6859–6866.

OpenAI. 2023. Fine-tuning. https://platform.openai.com/docs/guides/fine-tuning/fine-tuning. OpenAI Platform.

Liang Pang, Jun Xu, Qingyao Ai, Yanyan Lan, Xueqi Cheng, and Jirong Wen. 2020. Setrank: Learning a permutation-invariant ranking model for information retrieval. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 499–508.

Ronak Pradeep, Xueguang Ma, Rodrigo Nogueira, and Jimmy Lin. 2021a. Scientific claim verification with VerT5erini. In *Proceedings of the 12th International Workshop on Health Text Mining and Information Analysis*, pages 94–103, online. Association for Computational Linguistics.

Ronak Pradeep, Xueguang Ma, Rodrigo Nogueira, and Jimmy Lin. 2021b. Scientific Claim Verification with VerT5erini. In *Proceedings of the 12th International Workshop on Health Text Mining and Information Analysis*, pages 94–103.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Kirk Roberts, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, Kyle Lo, Ian Soboroff, Ellen Voorhees, Lucy Lu Wang, and William R. Hersh. 2021. Searching for scientific evidence in a pandemic: An overview of TREC-COVID. *Journal of Biomedical Informatics*, 121(C).

Somya Ranjan Sahoo and Brij B Gupta. 2021. Multiple features based approach for automatic fake news detection on social networks using deep learning. *Applied Soft Computing*, 100:106983.

Michael Schlichtkrull, Zhijiang Guo, and Andreas Vlachos. 2024. Averitec: A dataset for real-world claim verification with evidence from the web. *Advances in Neural Information Processing Systems*, 36.

Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter*, 19(1):22–36.

Nikhil Singh, Lucy Lu Wang, and Jonathan Bragg. 2024. Figura11y: Ai assistance for writing scientific alt text. In *Proceedings of the 29th International Conference on Intelligent User Interfaces*, pages 886–906.

Amir Soleimani, Christof Monz, and Marcel Worring. 2020. Bert for evidence retrieval and claim verification. In *Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part II 42*, pages 359–366. Springer.

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.

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 *PEMNLP*, pages 7534–7550.

David Wadden, Kyle Lo, Bailey Kuehl, Arman Cohan, Iz Beltagy, Lucy Lu Wang, and Hannaneh Hajishirzi. 2022a. SciFact-open: Towards open-domain scientific claim verification. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4719–4734, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

David Wadden, Kyle Lo, Lucy Lu Wang, Arman Cohan, Iz Beltagy, and Hannaneh Hajishirzi. 2022b. MultiVerS: Improving scientific claim verification with weak supervision and full-document context. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 61–76.

Dustin Wright, David Wadden, Kyle Lo, Bailey Kuehl, Arman Cohan, Isabelle Augenstein, and Lucy Lu Wang. 2022. Generating scientific claims for zero-shot scientific fact checking. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2448–2460, Dublin, Ireland. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.