Minimal Evidence Group Identification for Claim Verification

Xiangci Li[1,4*]   **Sihao Chen**[2*]   **Rajvi Kapadia**[3]   **Jessica Ouyang**[4]   **Fan Zhang**[3]

[1] AWS AI Labs, [2] Microsoft,
[3] Google Research, [4] University of Texas at Dallas

lixiangci8@gmail.com, sihaoc@cis.upenn.edu, rajvikapadia@google.com
Jessica.Ouyang@utdallas.edu, zhanfan@google.com

Abstract

Claim verification in real-world settings (e.g. against a large collection of candidate evidences retrieved from the web) typically requires identifying and aggregating a complete set of evidence pieces that collectively provide full support to the claim. The problem becomes particularly challenging when there exists distinct sets of evidence that could be used to verify the claim from different perspectives. In this paper, we formally define and study the problem of identifying such *minimal evidence groups* (MEGs) for claim verification. We show that MEG identification can be reduced from Set Cover problem, based on entailment inference of whether a given evidence group provides full/partial support to a claim. Our proposed approach achieves 18.4% & 34.8% absolute improvements on the WiCE and SciFact datasets over LLM prompting. Finally, we demonstrate the benefits of MEGs in downstream applications such as claim generation.

1 Introduction

The task of *claim verification* predicts whether a claim is supported by the presented evidence (Thorne et al., 2018; Chen et al., 2023a). A claim verification model is expected to identify the correct evidence pieces (EPs; e.g. evidence sentences or snippets) among tens of retrieved candidate evidence, but a practical challenge lies in that there might exist multiple sets of evidence that verify the claim from different perspectives. Figure 1 shows an example where, given a claim and some retrieved evidence, there exist two different — but both valid — ways of supporting the claim.

While humans can quickly identify mutually redundant EPs, e.g. e_1 and e_3 in Figure 1, and propose plausible combinations of EPs as *evidence groups* (EGs, formally defined in Section 3.1),
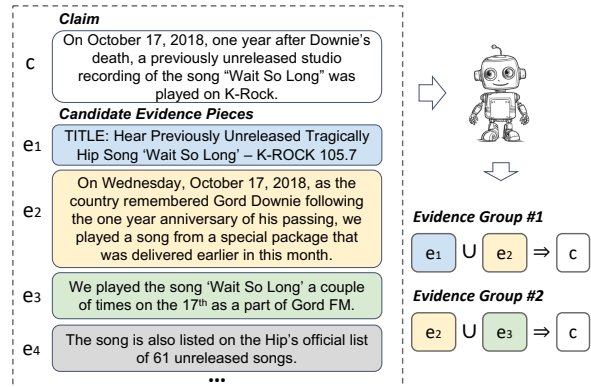

Figure 1: The problem of *minimal evidence group* identification for claim verification: given a claim and a list of candidate evidence pieces, the task is to identify the sets of *minimal, non-redundant* evidence, where each set provides full support for the claim.

existing claim verification systems (Dagan et al., 2005; Thorne et al., 2018; Wadden et al., 2020; Schuster et al., 2021; Chen et al., 2023a,b) focus only on the relationship between the claim and individual EPs, without considering the co-supporting relationships among EPs. This becomes problematic because the retrieved EPs might be redundant, or an individual EP may only partially support the claim. An EG with redundant EPs makes it more difficult to explain the reasoning for supporting the claim, while an EG composed of partially supporting EPs may still not fully support the claim, resulting in logical flaws. These problematic outputs not only confuse human verifiers, but also hurt the performance of downstream tasks.

In this paper, we introduce the problem of identifying *minimal evidence groups* (MEGs) from retrieved evidence candidates. Conceptually, an MEG is composed of EPs with the following properties: (1) **Sufficiency:** each MEG fully supports the veracity of the claim; (2) **Non-redundancy:** the EPs in an MEG are not redundant with each other; and (3) **Minimality:** the number of EPs in

each MEG is minimal. We formally define the task of MEG identification and show that classic claim verification approaches cannot effectively solve this problem. We propose a simple yet practical approach to decompose it to support prediction and evidence group merging. Our proposed approach significantly outperforms the baseline of directly prompting a large-language model (LLM) by 18.4% and 34.8% absolute precision scores on the WiCE (Kamoi et al., 2023) and SciFact (Wadden et al., 2020) benchmarks. Finally, we demonstrate the benefit of MEGs for saving computation budget in the downstream task of claim generation.

2 Related Work

Classic claim verification (Thorne et al., 2018; Chen et al., 2023a) consists of three steps: evidence retrieval, evidence selection, and stance prediction. *Evidence retrieval* perform coarse-grained filtering of EPs from thousands of candidates. *Evidence selection* and *stance prediction* perform fine-grained selection of EPs and predict whether the claim is supported by the selected EPs. MEG identification builds on classic claim verification by restricting evidence selection and stance prediction to predict a minimal group of EPs that fully supports the claim.

To address the problem that claim verification systems (Dagan et al., 2005; Wadden et al., 2020; Schuster et al., 2021; Chen et al., 2023b) may predict EPs that only partially support the claim, Laban et al. (2022); Schuster et al. (2022); Kamoi et al. (2023) aggregated individual EPs' entailment scores into EG scores. However, they did not address the problem of redundancy within an EG; we propose MEG identification to fill this gap.

The closest work to ours is SciFact (Wadden et al., 2020), which annotates "minimal evidence sets" for each claim. However, the SciFact shared task does not penalize non-minimal EGs, and consequently models that evaluate on SciFact (Pradeep et al., 2021; Li et al., 2021; Zhang et al., 2021; Wadden et al., 2022) are trained on the union of EGs from different human annotators, which is no longer minimal. Similarly, Thorne et al. (2018); Chen et al. (2023b); Kamoi et al. (2023) collect (possibly redundant) EGs from multiple annotators and use their union as training labels. As a result, existing models prioritize EP recall and are not directly comparable to MEG identification models.

3 Minimal Evidence Groups

3.1 Problem Formulation

MEG identification builds on the classic claim verification task (Thorne et al., 2018; Chen et al., 2023a). Formally, claim verification takes a claim c and a list of candidate EPs $E = \{e_1, e_2, ...\}$ as input. The evidence selection step retrieves all EPs that are *relevant* to c, and the stance prediction step predicts whether the selected EPs support c[1]. In Figure 1, e_1, e_2, e_3 all support c. A set of fully supporting EPs is called an evidence group (EG).

MEG identification requires the EGs to be *sufficient*, *non-redundant*, and *minimal*. We consider a set of EPs $S \subseteq E$ to *fully* or *partially support* a claim c if the EPs in S collectively entail all or only part of c, respectively; S does *not support* c if none of EPs in S entail c. If S *fully supports* c, it is an EG; an MEG is a minimal EG such that none of its EPs are redundant in terms of supporting c. In Figure 1, e_1 and e_3 are redundant; $\{e_1, e_2\}$ and $\{e_2, e_3\}$ are MEGs that fully support c.

3.2 Task Evaluation

We focus on precision-oriented scores (precision and $F_{0.5}$) to penalize predicting non-minimal EGs because we observe from prior claim verification datasets (Thorne et al., 2018; Wadden et al., 2020; Chen et al., 2023b; Kamoi et al., 2023) that (1) one MEG is sufficient for claim verification in practice; (2) humans are good at finding one plausible MEG but struggle to exhaustively find all valid MEGs; and (3) different annotators propose distinct MEGs.

Given a claim c with reference MEGs $RG = \{G_1, G_2, ...\}$, we measured the following metrics:

Exact match of MEGs treats each reference MEG atomically and considers a predicted MEG to be correct if it exactly matches a reference MEG.

Best soft match of MEGs gives partial credits to the predicted MEGs. We calculate the EP-level scores between the predicted MEG G' and the most similar reference MEG chosen by $\hat{G} = \arg\max_{G_i \in RG} F_{0.5}(G', G_i)$.

4 MEG Identification Approach

The challenge of MEG identification is to find the smallest set of EPs that fully supports the claim. As discussed in Section 2, classic claim verification models treat the EP as the basic unit; they are

---

[1] We limit our scope to claim support/non-support, ignoring contradictions for simplicity. See Section 7 for discussion.

Algorithm 1 MEG identification with a support prediction *Model*. Simplified for illustration, see Appendix Section A.2 for details.

Require: $c, E = \{e_1, e_2, ..., e_n\}, Model$
Require: max_size ▷ Max size of EGs to consider.
  $MEG \leftarrow []$ ▷ Proposed MEGs.
  for $size$ in $1... \min(|E|, max\_size)$ **do**
    $CS \leftarrow makeCombinations(c, E, size)$ ▷ List of *notRedundant* combinations of partially supporting EPs.
    for S in CS **do**
      $label \leftarrow Model(c, S)$
      if $label$ is *fully support* **then**
        $MEG.append(S)$
      end if
    end for
    if $len(MEG) > 0$ **then break**
    end if
  end for
  Output MEG

---

neither designed nor trained for *groups* of evidence. Our experiments of prompting directly with LLMs also show poor performance (Table 2, "Direct")[2].

As Algorithms 1 shows, we decompose MEG identification into two steps: (1) predicting whether a candidate set of EPs *fully supports*, *partially supports*, or does *not support* the claim and (2) bottom-up merging *partially supporting* groups in search of a *fully supporting* group. The support prediction *Model* can be implemented by any reasonable approach, such as prompting LLMs or fine-tuning models like T5 (Raffel et al., 2020). When merging groups, we increment the overall group size by 1 at each step. Note that if we only evaluate the base case with $size$=1, this is equivalent to classic claim verification (Thorne et al., 2018; Wadden et al., 2020; Schuster et al., 2021; Kamoi et al., 2023).

Based on the definition of MEG (Section 3.1), we derive three principles to prune the problem space for a tractable solution: (1) any superset of an MEG *fully supports* the claim c; (2) any non-empty subset of an MEG *partially supports* c; and (3) if a set of EPs S *fully supports* or does *not support* c, then S is not a strict subset of any MEG. Therefore, we stop merging sets that are predicted as *fully supporting* or *not supporting* to maintain the non-redundancy and minimality of the candidate EP sets. In addition, when choosing a pair of sets to merge, we prune the candidate merge partners for each set using a redundancy checker *notRedundant* (implemented as a zero-shot LLM prompt; see Appendix A.2). Finally, upon finding a *fully supporting* set, we stop merging and return all *fully supporting* sets of the current *size*.

---

[2]The explicit verification of combinations of EPs reduces from Set Cover and is NP-hard (see proof in Section 5.)

5 Proof of NP-hardness

In this section, we provide a simple proof to show that the MEG identification problem is NP-hard.

5.1 Simplifying to an Ideal Scenario

Inspired by Kamoi et al. (2023), who break complicated claims into subclaims and verify each subclaim individually, we assume the solution of the MEG identification problem explicitly breaks down the claim c into one or more atomic *claim units* $CU = \{cu_1, cu_2, ...\}$ and verifies them one-by-one. Each claim unit cu can be more fine-grained or abstractive than the subclaims introduced by Kamoi et al. If all claim units $cu_i \in CU$ are verified, then c is *fully supported*. Otherwise, if only a subset of CU is verified, then c is only *partially supported*. In an ideal scenario, we have a perfect model that is able to decompose c into CU and output a binary vector for each EP to indicate which cu_i are verified by the EP; this *ideal MEG identification problem* becomes the task of minimizing the number of selected EPs such that all elements in CU can be covered.

5.2 Reduction from Set Cover

Based on the formulation in 5.1, we can trivially many-one reduce the Set Cover problem, which is NP-Complete (Karp, 2010), to *ideal MEG identification* by mapping the universe to CU and the collection of subsets to the full set of EPs $E = \{e_1, e_2, ...\}$. Therefore the *ideal MEG identification problem* is NP-Complete, and the actual MEG identification problem is NP-hard. Because the assumption of explicitly tracking which cu_i are covered/remaining is challenging for state-of-the-art models, it is difficult to develop approximation solutions for MEG identification.

6 Base Model Performance

A base *Model* solves the *base problem* of support prediction: predicting whether a candidate set of EPs *fully supports*, *partially supports*, or does *not support* the claim. This can be implemented by any reasonable approach but we use an LLM-based approach for simplicity.

Implementation. For both the support prediction *Model* and *notRedundant* checker, we prompt PaLM-2L (Anil et al., 2023) with few-shot demonstrations and greedy decoding. We follow Wan et al. (2023) to select the LLM's most confident examples for few-shot demonstrations. To prioritize

Dataset	Accuracy	Precision			Recall			F1			
		Full	Partial	Not	Full	Partial	Not	Full	Partial	Not	Macro F1
WiCE	0.792	0.891	0.373	0.960	0.790	0.612	0.866	0.838	0.464	0.911	0.737
SciFact	0.729	0.833	0.077	0.794	0.741	0.095	0.848	0.784	0.085	0.820	0.563

Table 1: Base model performance.

precision, we take the top-1 predicted MEG, ranked according to the LLM's predicted *fully supporting* score, if multiple MEGs are predicted.

Experimental settings. To assess the support prediction *Model* performance, we construct datasets of 2255 and 462 entailment examples respectively from WiCE test-set and SciFact dev-set. The sampled WiCE subset contains 1139, 322, 794 *fully support*, *partially support*, and do *not support* examples, respectively. We directly use the annotated EGs from *fully* and *partially supporting* examples as inputs and randomly sample 1~3 EPs to serve as negative labels in *not supporting* examples. Similarly for SciFact, we treat each annotated evidence group as *fully supporting* and the subsets of annotated evidence groups as *partially supporting*; we randomly sample 1~3 non-annotated EPs to as negative lables for *not supporting* examples, obtaining 216, 42, and 204 *fully support*, *partially support*, and do *not support* examples, respectively. Table 1 shows the prompt used for the LLM.

Experimental results. Table 1 shows the support prediction base *Model* performance. Overall the model yields satisfactory performance on *fully* and *not supporting* examples but performs poorly on *partially supporting* examples. This is because the *partial support* label is vaguely defined, and presumably the LLM (Anil et al., 2023) did not encounter sufficient partially supporting entailment examples in its pretraining.

7 Intrinsic Evaluation

7.1 Experimental Settings

7.1.1 Datasets

We perform filtering to convert classic claim verification datasets to align with our MEG identification task. Both of the datasets listed below annotate EGs with multiple annotators. We assume that every human-annotated EG *fully supports* its claim, every subset of an EG *partially supports* its claim, and all non-labeled sentences do *not support* the claim. In addition, we assume each reference EG to be an MEG proposed by a different annotator.

SciFact (Wadden et al., 2020) is a biomedical fact-checking dataset and is the only existing dataset whose annotation instructions match the sufficiency, non-redundancy, and minimality requirements of MEGs. We remove all examples whose claims *contradict* the evidence, resulting in 268 samples from the development set. We use the non-contradictory EGs as-is. To distinguish it from the original SciFact dataset and task[3], we call this modified dataset **SciFact-MEG**.

WiCE (Kamoi et al., 2023) distinguishes EGs that *fully* or *partially support* claims from Wikipedia. We use the subclaim-level partition of the dataset and only use samples labeled as *fully supporting*, resulting in 528 samples from the test set. We call this modified dataset **WiCE-MEG**.

7.1.2 Baseline Approaches

Direct prediction. We zero-shot prompt PaLM-2L (Anil et al., 2023) to predict the MEG via EP indices, given a claim and a list of candidate EPs (Appendix Table 6).

Classic claim verification. To simulate classic claim verification without considering groups of EPs (Thorne et al., 2018; Wadden et al., 2020; Schuster et al., 2021; Kamoi et al., 2023), we use our proposed approach but early stop after computing $size$=1. If we find any *fully supporting* EP, the output MEG will be the same as our proposed approach. Otherwise, we concatenate all *partially supporting* EPs as a single EG.

Classic claim verification with less redundancy (Classic+LR). Given the output from "classic claim verification" above, we additionally remove EPs that cause redundancy, as predicted by the pair-wise *nonRedundant* checker[4].

7.2 Experimental Results

Table 2 shows the top-1 MEG identification performance using the metrics introduced in Section 3.2. For both datasets, our approach significantly

---

[3]As discussed in Section 2, while the SciFact dataset annotates EGs that meet the requirements of MEGs, the task does not evaluate redundancy or minimality, only sufficiency.

[4]We simply remove redundant combinations when $size$=2.

Dataset	Approach	Exact Match Precision	Best Soft Match Prec.	Recall	$F_{0.5}$
WiCE-MEG	Direct	0.456	0.176	0.522	0.203
	Classic	0.568	0.338	**0.554**	0.367
	Classic+LR	0.570	0.425	0.526	0.442
	Ours	**0.640**	**0.809**	0.423	**0.684**
SciFact-MEG	Direct	0.243	0.235	**0.652**	0.269
	Classic	0.479	0.468	0.478	0.470
	Classic+LR	0.479	0.491	0.476	0.488
	Ours	**0.591**	**0.612**	0.352	**0.533**

Table 2: Top-1 minimal evidence group identification performance. Examples with failed outputs are excluded for the baseline approach.

Input Evidence	# Words	# Sents	R-1	R-2	R-L
Direct	172.4	6.81	0.299	0.127	0.263
First-k Direct	34.1	1.15	0.282	0.114	0.250
Classic	85.0	3.20	0.282	0.120	0.250
Classic+LR	69.2	2.45	0.281	0.120	0.250
Our MEGs	*39.5*	*1.29*	*0.289*	*0.121*	*0.254*
Gold MEGs	*37.0*	*1.31*	*0.294*	*0.126*	*0.259*
Gold UEGs	71.7	2.78	0.302	0.128	0.267
First-k gold UEGs	33.0	1.31	0.264	0.101	0.232

Table 3: Budgeted retrieval-augmented generation performance (ROUGE F1).

outperforms all baselines on precision and $F_{0.5}$ scores. The baselines underperform our approach because their predicted MEGs contain too many EPs, especially the "Direct" LLM prompting baseline. Decomposing MEG identification into many individual entailment problems ("Classic") greatly improves the precision score. Further removing pair-wise redundancy ("Classic+LR") slightly improves performance, showing the impact of redundancy. Finally, although requiring significantly more computation, our bottom-up MEG identification approach performs the best because every combination of EPs is explicitly verified.

8 Extrinsic Evaluation

The non-redundancy of MEGs not only makes the evidence more human-readable, it also improves the performance of downstream applications. Inspired by Chen et al. (2023c), we use WiCE-MEG to highlight the MEG's minimality and sufficiency properties using claim generation as an example downstream task, with a computation budget measured in the number of words or sentences.

8.1 Experimental Settings

Since EGs fully entail their claims, they contain the information to reconstruct the claim. We compare the following input settings for the task of claim reconstruction using PaLM-2L (Anil et al., 2023):

MEGs. We use the top-1 MEG obtained with our proposed approaches, each baseline in Table 2, and the human-annotated reference EG with the smallest number of EPs for each claim.

Union of EGs (UEGs). We take the union of reference EGs (from different annotators) for a claim.

First-k. To simulate a low computation budget setting, we follow Chen et al. (2023c) in taking the first k EPs, where k is the size of the top-1 MEG.

8.2 Experimental Results

Table 3 shows that both our predicted and gold MEG settings perform comparably to settings with much lower computation budgets, while significantly outperforming the most constrained "first-k" settings. These results indicate that (1) our proposed approach for MEG identification is effective; (2) MEGs contain complete information for the claim generation task; (3) MEGs are much more compact than EGs from other approaches, with more than 45% fewer words, allowing them to be used in low-computation scenarios.

9 Conclusion

We have addressed the challenging scenario in claim verification where a model is expected to identify a minimal group of evidence pieces (EPs) among a relatively large amount of candidate evidence, and there might exist different sets of evidence that verify the claim from different perspectives. We formally define and study the problem of such minimal evidence group (MEG) identification and show that it can be reduced from a Set Cover-like problem. Our proposed approach achieves significant improvements over direct LLM prompting. Finally, we demonstrate the benefit of MEGs over classic claim verification approaches in downstream applications such as claim generation.

Limitations

Ignoring contradictions. In this work, we only consider supporting/non-supporting evidence for simplicity, and leave contradicting evidence for future work. Our proposed approach avoids the edge case of full/partial entailment problem brought by contradiction. Nonetheless, we claim that contradiction can be regarded as the opposite of support, where our proposed concepts and approaches still apply with minor fix.

Reliability of human annotations. As we point out in Section 1, there is no gold-standard annotated dataset designed for this task, and it is practically difficult to enforce and verify the sufficiency, non-redundancy, and minimality requirements of MEGs in the existing annotations. In practice, unless explicitly stated, it is unknown whether the annotated EGs are simply relevant to or fully support the claim. Although human annotators are good at proposing salient EGs, annotators usually do not exhaustively find all possible EGs. Moreover, human annotators tend to over-select EPs for a better contextualization, which breaks the non-redundancy and minimality requirements. As a result, we argue that the human annotations should only be treated as a reference, instead of an absolute gold standard. Therefore, the measured performance in Table 2 can should be regarded as each approach's agreement with the human annotators, and does not necessarily measure MEG correctness.

Definition of *partial support*. It is challenging to precisely define *partial support*. Even Kamoi et al. (2023), who proposed this label, did not clearly define it. Our proposed approaches do not rely on the precise definition of *partial support* but simply regard it as the intermediate label between *not support* and *fully support* because the precise definition may vary case-by-case in different datasets that the support prediction *Model* is trained on. Because of this ambiguity, *partial support* is the most challenging label for LLMs to predict (Section 6) and hurts the performance of MEG identification.

Computational complexity. Due to the NP-hardness (Section 5) of the MEG identification problem, we inevitably have to trade off running time for higher performance, which results in the worst-case running time for the proposed solution being too long to be practically useful in a production system. Our proposed approach is currently more suitable for high-quality dataset annotation, which requires a robust solution without strict running time requirements. Once a well-annotated dataset is created, future work can distill our slow but accurate approach to a faster and more efficient model.

The apparent conflict between the high computational complexity of our proposed MEG identification approach and the potential benefit of using MEGs when there is low computational budget for the downstream generation system. This conflict arises from a conceptual mismatch: on one hand, identifying MEGs at a high level does reduce the computational burden for downstream generation. On the other hand, the complexity of our approach is necessary because, as we discuss in in Section 4, there is no sufficiently large dataset for training a MEG identification model, and current LLMs do not perform well at directly predicting MEGs. As a workaround, we propose an *exact solution* that reduces MEG identification to the *base* entailment prediction problem, which we have proven to be inefficient (NP-hard in Section 5).

However, more efficient approximations may be possible. For instance, the "Classic+LR" setting in Section 7 simplifies the process by performing only one iteration of redundancy removal. Alternatively, we could employ a distillation approach (Yu et al., 2024), using our *exact solution* to automatically annotate a large training dataset, which would then allow us to train a specialized MEG identification model that predicts MEGs efficiently. However, this dataset creation process is complex, as it requires careful selection of queries, candidate evidence, and a well-performing *base model*, so we leave it for future work.

Ethical Statements

Similar to all prior claim verification works (Dagan et al., 2005; Thorne et al., 2018; Wadden et al., 2020; Schuster et al., 2021; Chen et al., 2023a,b), we stress that the MEG identification problem and the MEGs predicted by our proposed approach only consider the relative entailment relationship between the evidence and the claim. In other words, the MEG identification problem and our proposed approach do not guarantee the absolute correctness of the claim or the EPs or EGs themselves. Any future application must be cautious in distinguishing between retrieving evidence that supports the claim, correct or not, and verifying the absolute factual correctness of the claim.

References

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Jifan Chen, Grace Kim, Aniruddh Sriram, Greg Durrett, and Eunsol Choi. 2023a. Complex claim verification with evidence retrieved in the wild. *arXiv preprint arXiv:2305.11859*.

Sihao Chen, Senaka Buthpitiya, Alex Fabrikant, Dan Roth, and Tal Schuster. 2023b. PropSegmEnt: A large-scale corpus for proposition-level segmentation and entailment recognition. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8874–8893, Toronto, Canada. Association for Computational Linguistics.

Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Dong Yu, and Hongming Zhang. 2023c. Dense x retrieval: What retrieval granularity should we use? *arXiv preprint arXiv:2312.06648*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, pages 177–190. Springer.

Ryo Kamoi, Tanya Goyal, Juan Rodriguez, and Greg Durrett. 2023. WiCE: Real-world entailment for claims in Wikipedia. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7561–7583, Singapore. Association for Computational Linguistics.

Richard M Karp. 2010. *Reducibility among combinatorial problems*. Springer.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

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

Ronak Pradeep, Xueguang Ma, Rodrigo Nogueira, and Jimmy Lin. 2021. 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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Tal Schuster, Sihao Chen, Senaka Buthpitiya, Alex Fabrikant, and Donald Metzler. 2022. Stretching sentence-pair NLI models to reason over long documents and clusters. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 394–412, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get your vitamin C! robust fact verification with contrastive evidence. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 624–643, 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.

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.

David Wadden, Kyle Lo, Lucy Lu Wang, Arman Cohan, Iz Beltagy, and Hannaneh Hajishirzi. 2022. 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, Seattle, United States. Association for Computational Linguistics.

Xingchen Wan, Ruoxi Sun, Hootan Nakhost, Hanjun Dai, Julian Eisenschlos, Sercan Arik, and Tomas Pfister. 2023. Universal self-adaptive prompting. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7437–7462, Singapore. Association for Computational Linguistics.

Ping Yu, Jing Xu, Jason E Weston, and Ilia Kulikov. 2024. Distilling system 2 into system 1. In *The First Workshop on System-2 Reasoning at Scale, NeurIPS'24*.

Zhiwei Zhang, Jiyi Li, Fumiyo Fukumoto, and Yanming Ye. 2021. Abstract, rationale, stance: A joint model for scientific claim verification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3580–3586, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Prompt
Your task is to examine if the given claim is jointly supported by one or more evidence with short contexts. Take a deep breath and reason step by step, and answer with "FULLY_SUPPORTED", "PARTIALLY_SUPPORTED" or "NOT_SUPPORTED" at the end of your answer. FULLY_SUPPORTED means the claim is fully supported by the evidence without requiring other evidence. PARTIALLY_SUPPORTED means the claim is partially covered by the evidence that requires other evidence to collectively fully support the claim. NOT_SUPPORTED means the claim is not supported by the evidence. Example: Claim: {{example claim}} Evidence with contexts: {{example evidence text}} Answer: {{example answer}} Example: ... Your problem: Claim: {{claim}} Evidence with contexts: {{evidence text}} Answer:

Table 4: Prompt for *base problem*.

Prompt
Each of the following two evidence individually partially support the claim: "{{claim}}". Partial support means the claim is partially supported by the evidence that requires other evidence to collectively fully support the claim. Evidence 1: "{{evidence text 1}}". Evidence 2: "{{evidence text 2}}". Are evidence 1 and 2 redundant to each other in terms of how they support the claim, i.e. are they talking about the same thing, and is one of the evidence unnecessary? Take a deep breath and think step by step, and finally answer YES or NO.

Table 5: Prompt for checking redundancy of merged candidate EGs.

A Implementation Details

A.1 Additional Preprocessing

For the WiCE-MEG dataset, since the majority of the candidate EPs are not relevant to the claim, but some may be selected as part of the EGs by the LLM, we additionally filter out sentences without any stemmed token overlap with the claim in advance. This filtering removes 55.6% of candidate EPs but affects only 6.7% of gold EGs, significantly speeding up inference with minimal performance loss.

Prompt
Given the following claim: "{{claim}}", and evidence sentences prepended with indices: {{evidence text}} Select the best minimal non-redundant group of evidence sentences that fully supports the claim. Only output sentence indices, separated by comma. Answer:

Table 6: Prompt for directly predicting MEG.

Prompt
Write a claim that is fully supported by the given following evidence sentences: {{evidence text}}

Table 7: Prompt for claim reconstruction.

A.2 Detailed Algorithm

To avoid redundant computation, we iteratively merge two *partially supporting* set of EPs to a larger candidate set and store it in PGs in Algorithm 2. Therefore, PGs is implemented by a Python dictionary with size of the set of EPs as keys and another nested Python dictionary CS as values. Each CS has a key of the merged set of EPs $G_1 \cup G_2$, and values of pair of the (G_1, G_2). Algorithm 2 & 3 presents the full pseudo code of our implementation. In Algorithm 3 we prepare non-redundant candidate sets of EPs by running $notRedundant$ checker implemented by a zero-shot LLM prompt (Table 5).

A.3 Inter-annotator Disagreement

In WiCE (Kamoi et al., 2023) dataset, we observe some inter-annotator disagreements where some human-labeled EGs are supersets of the other EGs for the same claim, but in these cases we still include both EGs as references.

Algorithm 2 Minimal Evidence Group Identification with a support prediction Model.

Require: c, $E = \{e_1, e_2, ..., e_n\}$, $Model$
Require: max_size ▷ Max size of EGs to consider.
  $MEG \leftarrow []$ ▷ Proposed MEGs.
  $PGs \leftarrow \{\}$ ▷ Dict[size: Dict[G: {G}]]
  for $size$ in $1... \min(|E|, max\_size)$ **do**
    $PGs \leftarrow MergePartialGroup(c, E, size, PGs)$
    $CS \leftarrow PGs[size].keys()$ ▷ All candidate sets of EPs
with size $size$
    for S in CS **do**
      $label \leftarrow Model(c, S)$
      if $label$ is $fully\ support$ **then**
        $MEG.append(S)$
        pop $PGs[size][S]$
      else if $label$ is $not\ support$ **then**
        pop $PGs[size][S]$
      end if
    end for
    if $len(MEG) > 0$ **then break**
    end if
  end for
  Output MEG

Algorithm 3 Merging partial evidence groups with redundancy checking.

Require: $notRedundant$ ▷ Redundancy Checker.
  function MERGEPARTIALGROUP(c, E, $size$, PGs)
    $CS \leftarrow \{\}$ ▷ Dictionary of Sets.
    if $size = 1$ **then**
      for e in E **do**
        $CS[(e,)] \leftarrow set([])$
      end for
    else
      for each pair $G_1 \in PGs[|G_1|]$ & $G_2 \in PGs[|G_2|]$ s.t. $|G_1 \cup G_2| = size$ & $notRedundant(c, G_1, G_2)$ **do**
        $CS[G_1 \cup G_2].add((G_1, G_2))$
      end for
    end if
    $PGs[size] \leftarrow CS$
    return PGs
  end function