

Reference-Free Schema Generation for Literature Review Tables via Multi-Faceted Rewards

Sinjoy Saha and Suman Saha and Mahfuza Farooque and Wenpeng Yin

Department of Computer Science & Engineering

The Pennsylvania State University

University Park, PA 16802, USA

{sks7620, sumsaha, mff5187, wenpeng}@psu.edu

Abstract

To accelerate scientific knowledge acquisition, LLMs are increasingly used to synthesize multiple papers into structured tables by inferring schemas and values. While value generation within a fixed schema can often be reduced to extractive question answering, the *schema generation* problem, determining which dimensions to compare a set of documents, lacks a formal mapping to standard NLP tasks. In this work, we formulate schema generation as a reinforcement learning problem and investigate whether these dimensions can be induced without access to gold-standard schemas. We design a multi-faceted reward framework capturing schema coverage, non-redundancy, relevance, and format, and train a small language model on a literature review dataset. Our approach yields consistent improvements over the untuned base model across intrinsic, reference-based, and LLM-judge metrics, and remains competitive with supervised fine-tuned models at $5\times$ the parameter count on structural and diversity dimensions. All code, results and prompts are available in the GitHub repository: <https://github.com/sinjoysaha/rl-schema-generation>

1 Introduction

Large language models (LLMs) have shown significant promise in automating literature search (Ajith et al., 2024). However, structured knowledge synthesis from scientific literature remains a fundamental challenge in research automation. Literature review systems aim to generate literature review tables from a corpus of papers to compare along different dimensions (Singh et al., 2025b; Fok et al., 2025).

The task of literature review table generation comprises two components: (1) Schema generation: determining a set of aspects or attributes that are relevant to and shared across a given corpus, and (2) Value generation: identifying the content

for each paper (row) and attribute (column). Once a schema is established, value generation can largely be reduced to traditional slot-filling or question-answering tasks (Azarbondy et al., 2025) at which modern retrieval-augmented generation (RAG) systems have become increasingly capable (Koshorek et al., 2025). Schema generation, however, still lacks a formal mapping to standard NLP tasks and remains comparatively underexplored (Padmakumar et al., 2025).

Determining the dimensions along which to compare a set of input documents is inherently difficult (Russell et al., 1993). The problem is further compounded by the scarcity of large, high-quality datasets suitable for supervised fine-tuning of schema generation models. Reinforcement learning (RL) presents an alternative paradigm. Rather than imitating gold-standard schemas, a model can be trained to directly optimize for schema quality criteria expressed as reward signals (Sutton and Barto, 2018; Ouyang et al., 2022). However, designing such reward functions is non-trivial. Unlike value generation, where correctness can often be grounded in extractive textual evidence, schema quality is multi-dimensional and weakly supervised. A high-quality schema must simultaneously satisfy: (1) *Relevance*: capturing aspects that are central to the input corpus, (2) *Coverage*: spanning the major conceptual axes of the corpus, and (3) *Non-redundancy*: being semantically distinct and avoid overlapping comparison dimensions.

In this work, we study whether RL with programmatic and model-based rewards can effectively induce these properties without access to gold-standard schemas. We design a reward framework comprising: (1) an *entailment* reward that grounds attributes in the input corpus using a natural language inference (NLI) (Bowman et al., 2015; Wang et al., 2020) model, (2) a *coverage* reward that encourages schemas with a larger number of attributes for broader coverage, (3) a *dissimilar-*

ity reward that penalizes semantically similar attributes, and (4) a *format* reward that enforces desired schema structure. These rewards are inference model-agnostic and require no reference schemas.

We train a small language model (SLM) on the literature review tables dataset, *arxivDIGESTables* (Newman et al., 2024) using our multi-faceted reward framework without access to gold-standard reference tables and evaluate it on a high-quality subset, *arxivDIGESTables-Clean* (Padmakumar et al., 2025). We compare our method to open and closed weight models and prior supervised fine-tuning (SFT) approaches, and find that our method improves over the base model across evaluation axes and is competitive with larger SFT models.

Our contributions are as follows: (i) We formalize schema generation as a reward-driven optimization problem and propose, to the best of our knowledge, the first RL-based training framework for this task that requires no gold-standard schema annotation. (ii) We design a taxonomy of four complementary reward signals spanning structural and semantic quality dimensions. (iii) We train an SLM with this mixed reward signal and demonstrate improvements over the baseline across multiple evaluation dimensions.

2 Related Work

Before LLMs, multi-document summarization was addressed through information extraction (IE) and template-based approaches. White et al. (2001) used predefined slot-value templates, Chiticariu et al. (2018) introduced a user-defined rule-based framework to extract table-like summaries, and Zhang and Balog (2018) induced entity-attribute schemas from query intent and corpus statistics, and filled cells via retrieved evidence snippets.

In the LLM era, progress has been made on broader scientific literature review systems such as AutoSurvey (Liu et al., 2025) and LitLLM (Agarwal et al., 2024) to support survey writing and review workflows. Benchmarks such as SciEval provide multi-level evaluation for scientific reasoning and generation (Sun et al., 2024). Wang et al. (2025) proposed evaluation protocols for paper retrieval and tabular summaries. Various schema structures for scientific papers have also been explored (Hashimoto et al., 2017; Zhu et al., 2023; Bai et al., 2024).

Newman et al. (2024) introduced the ArxivDIGESTables dataset to study literature review table

generation from scientific research papers using various prompting methods, including joint and sequential schema-value generation. Padmakumar et al. (2025) proposed generating candidate schemas conditioned on user-specified table intents and iteratively refining attributes via guided prompting, and showed that SLMs fine-tuned on reference tables can be competitive with state-of-the-art prompted LLMs.

These prior approaches either (i) rely on predefined, rule-driven, or retrieval-induced schemas with rule-based extraction, (ii) schema generation is performed via various prompting methods, or (iii) require gold standard reference tables to fine-tune a model. In this work, we propose a reward engineering framework for the schema generation task that defines corpus-level reward signals—covering corpus entailment, semantic non-redundancy, attribute coverage, and format validity—to train SLMs without access to reference schemas.

3 Methodology

3.1 Problem Formulation

Given a corpus $C = \{d_1, \dots, d_m\}$ of scientific documents, the goal of schema generation is to produce attributes $S = \{a_1, \dots, a_n\}$ defining the column headers of a literature review table. We frame this as an RL problem: an LM policy π_θ receives C as input, produces a schema S , and is scored by a reward function $R(S, C)$ measuring schema quality.

3.2 Multi-faceted Reward Design

A high-quality schema must be structurally valid, comprehensive, corpus-grounded, and semantically non-redundant. We design heterogeneous reward functions approximating these properties without requiring reference schemas. The total reward is a weighted combination of these components.

$$R(S, C) = \lambda_1 R_E(S, C) + \lambda_2 R_N(S) \\ + \lambda_3 R_D(S) + \lambda_4 R_F(S)$$

where $R_E(\cdot)$ and $R_N(\cdot)$ are the entailment and coverage rewards and $R_D(\cdot)$ and $R_F(\cdot)$ are the dissimilarity and format rewards. In this work, we keep the reward weights (λ_i) uniform. The reward components are described next.

Entailment Reward. Embedding and entailment rewards capture semantic properties not verifiable by symbolic checks. Following Padmakumar et al.

(2025), each attribute a_i is expanded into an enriched representation A_i by concatenating the attribute name with a brief model-generated description, providing richer context for downstream computations. Each enriched attribute A_i is used as a hypothesis against each document $d_j \in C$ as premise. A pretrained NLI model \mathcal{M}_{NLI} produces entailment probabilities:

$$p_{ij} = P_{\mathcal{M}_{\text{NLI}}}(\text{entail} \mid d_j, A_i).$$

The reward averages these across all attribute-document pairs:

$$R_E(S, C) = \frac{1}{nm} \sum_{i=1}^n \sum_{j=1}^m p_{ij},$$

rewarding attributes that are consistently inferable from the corpus.

Coverage Reward. To encourage broader comparison dimensions, we define:

$$R_N(S) = \min\left(\frac{n}{N_{\max}}, 1\right),$$

where $n = |S|$ and N_{\max} is a saturation threshold, promoting richer schemas without unbounded growth.

Dissimilarity Reward. To discourage semantically overlapping attributes, we compute mean pairwise cosine dissimilarity between enriched representations using embedding model \mathcal{M}_{emb} :

$$R_D(S) = \frac{2}{n(n-1)} \sum_{i < j} \left(1 - \frac{\text{sim}(A_i, A_j) + 1}{2}\right),$$

where cosine similarity is scaled from $[-1, 1]$ to $[0, 1]$. Higher values indicate more semantically diverse attribute dimensions.

Format Reward. The format reward R_F is a graded score in $[0, 1]$ that verifies structural correctness at four levels:

$$R_F(S) = \begin{cases} 0.3 \cdot \mathbf{1}[\text{valid JSON}] \\ +0.2 \cdot \mathbf{1}[\text{non-empty dict}] \\ +0.3 \cdot \mathbf{1}[\text{all attrs have required fields}] \\ +0.2 \cdot \mathbf{1}[\text{all field types valid}], \end{cases}$$

where each indicator is evaluated only if all preceding checks pass. Specifically, the schema S must parse as valid JSON (+0.3); its top-level structure must be a non-empty dictionary

(+0.2); every attribute must contain definition and output_format sub-fields (+0.3); and both sub-fields must be strings (+0.2). A schema that fails to parse at any stage receives the reward accumulated up to that point.

4 Experiment

Dataset. We use the *arxivDIGESTables* dataset (Newman et al., 2024), comprising literature review tables from scientific survey papers across multiple domains. For evaluation, we use the 100-instance high-quality subset *arxivDIGESTables-Clean* released by Padmakumar et al. (2025). For training, we exclude all tables referencing papers in the evaluation set, yielding 1,730 training instances. The paper titles and abstracts are used from each corpus to fit within the model context window.

Training details. We fine-tune *Qwen3-0.6B* (Yang et al., 2025) using Group Relative Policy Optimization (GRPO) (Shao et al., 2024), a policy gradient method that normalizes rewards within a group of K sampled outputs to reduce variance. At each step, K candidate schemas are sampled from π_θ for a given corpus C and scored with $R(S^{(k)}, C)$; the policy is updated to favor higher-rewarded outputs relative to the group mean. A sentence-transformer model is used as \mathcal{M}_{emb} to compute Text embeddings for R_D .¹ A cross-encoder model is used as \mathcal{M}_{NLI} for entailment probabilities in R_E .² Hyperparameters, prompts and examples of generations are given in Appendix A, B and C.

Evaluation metrics. We evaluate generated schemas along three complementary axes. i) **Intrinsic reward metrics.** At test time, we report each of the four reward signals from Section 3.2 directly as metrics: format validity (F), coverage (N), attribute dissimilarity (D), and entailment (E). All four are reference-free and computed directly from the generated schema and input corpus. ii) **Reference-based metrics (BERTScore).** Following Newman et al. (2024) and Padmakumar et al. (2025), we align generated and reference schemas using BERTScore computed over enriched attributes. An attribute pair is considered a match if similarity exceeds a threshold τ ; to avoid sensitivity to any single τ , we compute precision,

¹<https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2>

²<https://huggingface.co/cross-encoder/nli-MiniLM2-L6-H768>

| Gold Used? | Model | Intrinsic Rewards | | | | Ref-based (BERTScore) | | | LLM-Judge | | |
|------------|---------------------------------------|-------------------|-------------|--------------|-------|-----------------------|-------------|--------------|-------------|-------------|-------------|
| | | R_F | R_N | R_D | R_E | R | P | F1 | Rel. | Useful | Clarity |
| Y | Llama (SFT) (Padmakumar et al., 2025) | 0.98 | 0.39 | 0.275 | 0.070 | 0.22 | 0.24 | 0.225 | 0.74 | 0.64 | 0.70 |
| Y | Qwen (SFT) (Padmakumar et al., 2025) | 1.00 | 0.38 | 0.279 | 0.066 | 0.24 | 0.26 | 0.243 | 0.81 | 0.71 | 0.78 |
| N | Qwen-prompting (Newman et al., 2024) | 0.86 | 0.49 | 0.250 | 0.047 | 0.16 | 0.16 | 0.156 | 0.72 | 0.61 | 0.55 |
| N | Qwen 3 0.6B + RL (Ours) | 0.91 | 0.51 | 0.259 | 0.042 | 0.18 | 0.18 | 0.175 | 0.77 | 0.66 | 0.60 |

Table 1: Performance comparison of RL model without gold standard data (our approach) with baseline prompting and SFT models trained on gold standard data on intrinsic (programmatic and model) rewards, reference-based metrics, and LLM-as-a-Judge metrics. Programmatic (P) Rewards (R_F : Schema Format, R_N : Number of attributes or Coverage). Model (M) Rewards (R_D : Attribute Dissimilarity, R_E : Entailment). Reference-based AUC scores: Recall, precision, and F1. LLM-as-a-Judge (GPT-5-mini) metrics: Relevance, usefulness, and clarity. **Bold** denotes the best when gold schema not used for training.

recall, and F1 across $\tau \in [0.40, 1.00]$ in increments of 0.01 and report the area under the curve (AUC) via the trapezoidal rule. iii) **LLM-as-judge metric**. Automated metrics may fail to capture holistic schema quality for literature reviews from the perspective of a researcher. We prompt GPT-5-mini (Singh et al., 2025a) with the input corpus and the generated schema, asking it to score each schema on a scale from 0 to 1 along three dimensions: (i) *relevance* - whether the attributes capture meaningful aspects of the provided papers and support comparison; (ii) *usefulness* - whether the attributes are sufficiently specific and informative; and (iii) *clarity* - whether the attribute names, definitions, and output formats are precise and well-defined.

4.1 Results & Analyses

Table 1 compares our RL-trained model against SFT baselines (Llama 3.2 3B and Qwen 2.5 3B) (Padmakumar et al., 2025) and baseline prompting on Qwen 3 0.6B (Newman et al., 2024). Despite being a 0.6B-parameter model trained without any reference schemas, our approach improves over the base Qwen-3-0.6B not only on intrinsic rewards but also on reference-based and LLM-Judge metrics, demonstrating that reward engineering alone can drive meaningful gains even at a very small model scale. The SFT models, trained on 3B parameters with gold-standard schemas, achieve stronger reference-based and entailment scores, suggesting that supervised training remains advantageous for corpus grounding. Our model’s dissimilarity score and LLM-Judge metrics remain competitive with SFT, indicating that the RL objective successfully induces diverse and meaningful schema attributes even without any supervised signal.

Error Analysis. We report qualitative observations of errors in schemas generated by the different models. A recurring structural failure occurs when

the model generates unescaped quotation marks within a JSON string value. For instance, in some cases, the `output_format` value itself contains an embedded quoted string ("`98.9%`") without proper escape sequences, rendering the entire JSON object unparseable by a standard decoder. A second class of errors concerns the semantic quality of `output_format` values. Many generated schemas specify vague types such as "numbers or strings" rather than one specific type. While such schemas pass the format reward check, they may not offer much guidance for the subsequent value generation step, suggesting that the current reward formulation does not penalize under-specified output formats. These errors point to directions of future reward refinement. An example error is illustrated in Appendix D.

5 Conclusion

In this study, we present an RL framework for schema generation in literature review tables that trains SLMs to optimize a mixed reward signal—comprising format validity, attribute coverage, semantic non-redundancy, and corpus entailment—without access to gold-standard schemas. Experiments on *arxivDIGESTables-Clean* show consistent improvements over the base model across most evaluation dimensions. Compared to SFT baselines at $5\times$ the parameter count, our model remains competitive on structural and diversity metrics while trailing on corpus grounding. Overall, our results suggest that schema generation can be framed as a reward-driven optimization problem, offering an alternative to supervised annotation.

Acknowledgments

This work was supported by NSF DMS-2533995.

Limitations

While this work offers insights into the training dynamics of using programmatic and model-based reward functions for training SLMs for literature review schema generation, it has several limitations. First, our experiments are conducted on a scientific literature review tables dataset, *arxivDIGESTables-Clean*, which, although manually cleaned, contains only 100 scientific tables and may not capture the generalization of structured knowledge synthesis in other domains, such as legal, medical, or news corpora. Second, our evaluation relies on intrinsic metrics, BERTScore computed against reference schemas, and an LLM-judge metric for the usefulness of the proposed schema. While these approximate human judgments, large-scale human evaluation may be an important direction for future work. Additionally, the reward weights are fixed across training. Studying optimal reward weighting may be a future direction. Finally, while we try to formulate the reward signals to complement each other, as with any RL-based approach, there is a risk that the model learns to exploit reward signals in unintended ways. In the future, more robust reward formulations may be investigated.

References

- Shubham Agarwal and 1 others. 2024. LitLLM: A toolkit for scientific literature review. In *Proceedings of the 2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Anirudh Ajith, Mengzhou Xia, Alexis Chevalier, Tanya Goyal, Danqi Chen, and Tianyu Gao. 2024. *LitSearch: A retrieval benchmark for scientific literature search*. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15068–15083, Miami, Florida, USA. Association for Computational Linguistics.
- Hosein Azarbondy, Zi Long Zhu, Georgios Cheirmpis, Zubair Afzal, Vikrant Yadav, and Georgios Tsatsaronis. 2025. *Question-answer extraction from scientific articles using knowledge graphs and large language models*. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '25*, page 1120–1129, New York, NY, USA. Association for Computing Machinery.
- Fan Bai, Junmo Kang, Gabriel Stanovsky, Dayne Freitag, Mark Dredze, and Alan Ritter. 2024. *Schema-driven information extraction from heterogeneous tables*. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10252–10273, Miami, Florida, USA. Association for Computational Linguistics.
- Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. *A large annotated corpus for learning natural language inference*. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.
- Laura Chiticariu, Marina Danilevsky, Yunyao Li, Frederick Reiss, and Huaiyu Zhu. 2018. *SystemT: Declarative text understanding for enterprise*. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 76–83, New Orleans - Louisiana. Association for Computational Linguistics.
- Raymond Fok, Joseph Chee Chang, Marissa Radensky, Pao Siangliulue, Jonathan Bragg, Amy X. Zhang, and Daniel S. Weld. 2025. *Facets, taxonomies, and syntheses: Navigating structured representations in llm-assisted literature review*. *Preprint*, arXiv:2504.18496.
- Hayato Hashimoto, Kazutoshi Shinoda, Hikaru Yokono, and Akiko Aizawa. 2017. *Automatic generation of review matrices as multi-document summarization of scientific papers*. In *BIRNDL@SIGIR*.
- Omri Koshorek, Niv Granot, Aviv Alloni, Shahar Admati, Roei Hendel, Ido Weiss, Alan Arazi, Shay-Nitzan Cohen, and Yonatan Belinkov. 2025. *Structured rag for answering aggregative questions*. *Preprint*, arXiv:2511.08505.
- Yixin Liu, Yonghui Wu, Denghui Zhang, and Lichao Sun. 2025. *Agentic autosurvey: Let llms survey llms*. *Preprint*, arXiv:2509.18661.
- Benjamin Newman, Yoonjoo Lee, Aakanksha Naik, Pao Siangliulue, Raymond Fok, Juho Kim, Daniel S Weld, Joseph Chee Chang, and Kyle Lo. 2024. *ArxivDIGESTables: Synthesizing scientific literature into tables using language models*. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9612–9631, Miami, Florida, USA. Association for Computational Linguistics.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.
- Vishakh Padmakumar, Joseph Chee Chang, Kyle Lo, Doug Downey, and Aakanksha Naik. 2025. *Intent-aware schema generation and refinement for literature review tables*. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages

23450–23472, Suzhou, China. Association for Computational Linguistics.

Daniel M. Russell, Mark J. Stefik, Peter Pirolli, and Stuart K. Card. 1993. [The cost structure of sense-making](#). In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, CHI '93, page 269–276, New York, NY, USA. Association for Computing Machinery.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y.K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, Akshay Nathan, Alan Luo, Alec Helyar, Aleksander Madry, Aleksandr Efremov, Aleksandra Spyra, Alex Baker-Whitcomb, Alex Beutel, Alex Karpenko, and 465 others. 2025a. [Openai gpt-5 system card](#). *Preprint*, arXiv:2601.03267.

Amanpreet Singh, Joseph Chee Chang, Chloe Anastasiades, Dany Haddad, Aakanksha Naik, Amber Tanaka, Angele Zamarron, Cecile Nguyen, Jena D. Hwang, Jason Dunkleberger, Matt Latzke, Smita Rao, Jaron Lochner, Rob Evans, Rodney Kinney, Daniel S. Weld, Doug Downey, and Sergey Feldman. 2025b. [Ai2 scholar qa: Organized literature synthesis with attribution](#). *Preprint*, arXiv:2504.10861.

Liangtai Sun, Yang Han, Zihan Zhao, Da Ma, Zhennan Shen, Baocai Chen, Lu Chen, and Kai Yu. 2024. Sci-eval: A multi-level large language model evaluation benchmark for scientific research. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19053–19061.

Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*, 2nd edition. MIT Press.

Weiqi Wang, Jiefu Ou, Yangqiu Song, Benjamin Van Durme, and Daniel Khashabi. 2025. [Can llms generate tabular summaries of science papers? rethinking the evaluation protocol](#). *Preprint*, arXiv:2504.10284.

Zikang Wang, Linjing Li, and Daniel Zeng. 2020. [Knowledge-enhanced natural language inference based on knowledge graphs](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6498–6508, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Michael White, Tanya Korelsky, Claire Cardie, Vincent Ng, David Pierce, and Kiri Wagstaff. 2001. [Multi-document summarization via information extraction](#). In *Proceedings of the First International Conference on Human Language Technology Research*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao,

Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.

Shuo Zhang and Krisztian Balog. 2018. [On-the-fly table generation](#). In *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '18, page 595–604. ACM.

Kun Zhu, Xiaocheng Feng, Xiachong Feng, Yingsheng Wu, and Bing Qin. 2023. [Hierarchical catalogue generation for literature review: A benchmark](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6790–6804, Singapore. Association for Computational Linguistics.

A Hyperparameters

Table 2 reports training hyperparameters.

| Hyperparameter | Value |
|---------------------|-----------|
| N_{\max} | 10 |
| GRPO group size K | 4 |
| Learning rate | 10^{-6} |
| Precision | bf16 |
| Grad. Accum. Step | 1 |

Table 2: Hyperparameter settings used in all experiments.

B Prompts

B.1 Prompt for schema generation

Adapted from *arxivDIGESTables-Clean*.

Schema Generation Prompt

Imagine the following scenario: A user is making a table for a scholarly paper that contains information about multiple papers and compares these papers. To compare and contrast the papers, the user provides the title and content of each paper.

Your task is the following:
Given a list of papers, you should find the appropriate number of attributes that are shared by the given research papers and can be used to compare them. So each attribute is a topic that would be covered in the Related Work section of the user's paper.

Return a JSON object in the following format:

```
'''json
{{
  "<attribute 1>":
```

```

{{
  "definition":<your definition of why
    this attribute should be an axis
    of comparison>,
  "output_format": <describe the range
    of output values that will be
    filled in, is it numbers, string
    values or another format>
}},
...
}},
,,,
{Paper 1} ... {Paper n}

```

```
}}
```

C Sample Generations

Example Schema Generation Output

```

<think>
Okay, so I need to figure out which
attributes are shared between these
seven papers for a Related Work
section. Let's start by looking at
the titles and abstracts of each
paper to see if there's a common
theme or aspect that can act as a
comparison axis.

```

```

Starting with Paper 1: "Joint Wireless
Information and Energy Transfer in a
K$-User MIMO Interference Channel".
The abstract mentions the use of MIMO,
energy transfer, and different
scenarios. Paper 2's title is "MIMO
Broadcasting for Simultaneous
Wireless Information and Power
Transfer". Here, they mention
broadcasting, WPT, and the use of
MIMO. ...

```

```

Since the problem mentions finding
attributes that are shared, and each
paper has a similar focus, I think
these are the main ones.

```

```

Therefore, the JSON would have keys for
each of these attributes. Each key's
definition would state why they're an
axis of comparison. The output
format should be specified as a
string or numbers depending on the
values. Since the examples in the
problem have output format as strings,
maybe the values would be the counts
or specific terms. However, since
the problem says to output in JSON,
the actual attributes and their
definitions would be the keys. For
example:\n\n"Beamforming Techniques":
{"definition": "This refers to the
use of beamforming techniques in the
context of wireless communication and
energy transfer systems.", "
output_format": "string"},
...

```

```

And similarly for others. I need to make
sure each attribute is unique and
covers the key aspects mentioned in
the papers.
</think>

```

```

""json
{
  "Beamforming Techniques": {
    "definition": "This refers to the use
of beamforming techniques in the
context of wireless communication
and energy transfer systems.",

```

B.2 Prompt for LLM-as-a-Judge evaluation

LLM-Judge Prompt

You are an expert academic evaluator. Your task is to evaluate the quality of comparison aspects (attributes) proposed for a set of research papers.

The model was given the following task:
{original_prompt}

The model generated the following response with proposed attributes:

```

""json
{generated_json}
""

```

Evaluation Task:

For each attribute (the keys in the JSON above), evaluate it on the following criteria (0 to 1 scale):

1. **Relevancy**: Is the attribute relevant to the specific papers provided? Does it capture a meaningful dimension of comparison?
2. **Usefulness**: Would this attribute be useful in a Related Work section to contrast these papers? Is it specific enough?
3. **Clarity**: Is the attribute name, definition, and output format clear and well-defined?

Output Format:

Return your evaluation ONLY as a JSON object with the following structure:

```

{{
  "attribute_evaluations": [
    {{
      "attribute": "<attribute_name>",
      "relevancy": <score 0-1>,
      "usefulness": <score 0-1>,
      "clarity": <score 0-1>,
      "reasoning": "<brief reasoning>"
    }},
    ...
  ],
  "overall_average_score": <average across
all attributes and metrics, scaled
0-1>

```

```

"output_format": "string"
},
"Energy Transfer Methods": {
"definition": "This refers to the
  methods used for transferring
  energy in wireless communication
  systems.",
"output_format": "string"
},
"Information Decoding": {
"definition": "This refers to the
  process of decoding information
  signals in communication systems
  .",
"output_format": "string"
},
"Energy Harvesting": {
"definition": "This refers to the use
  of energy harvesting techniques in
  wireless communication systems.",
"output_format": "string"
},
"System Performance": {
"definition": "This refers to the
  overall performance of the
  communication and energy transfer
  systems.",
"output_format": "string"
}
}
'''

```

```

},
{
"attribute": "Information Decoding",
"relevancy": 0.95,
"usefulness": 0.6,
"clarity": 0.6,
"reasoning": "Information decoding (ID)
  appears in all papers (ID vs EH
  receivers, SINR constraints). However,
  as defined it is vague and overlaps
  with receiver architecture/
  constraints; output_format 'string'
  is unspecific. More precise
  attributes (e.g., receiver types,
  decoding capabilities, interference
  cancellation) would be more useful."
},
{
"attribute": "Energy Harvesting",
"relevancy": 1.0,
"usefulness": 0.65,
"clarity": 0.6,
"reasoning": "Energy harvesting
  constraints and objectives are
  present in every paper (harvested
  power, EH receiver models). The
  attribute is relevant but redundant
  with 'Energy Transfer Methods' and
  insufficiently specific (should
  specify metrics like harvested power
  targets, EH model, efficiency)."
```

Example LLM-as-a-Judge Output

```

{
"attribute_evaluations": [
{
"attribute": "Beamforming Techniques",
"relevancy": 1.0,
"usefulness": 0.9,
"clarity": 0.7,
"reasoning": "All papers focus on
  beamforming/precoding design (MIMO/
  MISO, rank-one beams, MRT/ZF, SDP-
  based designs). The attribute is
  highly relevant and useful, but the
  provided definition is generic and
  the output_format ('string') is not
  specific (should allow enumerated
  methods or descriptors)."
```

D Schema Generation Error

```

<think>
Okay, let's see. The user wants me to
create a JSON object based on the
given papers where each attribute is
shared by multiple papers and can be
used in the Related Work section.
Each attribute should have a
definition, output format, and
possible values.

First, I need to look through all the
papers and identify common attributes.
Let's start by checking each paper's

```

title and abstract to find shared topics.

Looking at Paper 1: the focus is on deep learning for malware detection and using explainable algorithms. Paper 2 talks about malware evasion and ML classifiers. Paper 3 is about attack methods using COPYCAT. Paper 4 discusses adversarial examples. Paper 5 is about adversarial examples in malware detection. Paper 6 compares ML vs AV. Paper 7 is about black-box attacks on RNN. Paper 8 is about generating adversarial examples in deep learning. Paper 9 and 10 involve black-box attacks and evasion.

Now, I need to find attributes that are shared across multiple papers. Let's think about topics like "explanation methods", "machine learning approaches", "adversarial techniques", "robustness", etc.

For example, Paper 1 uses explainable techniques (feature attribution), Paper 3 uses COPYCAT, Paper 4 and 8 use adversarial examples. Paper 5's title mentions generating adversarial examples. Paper 6 talks about robustness. Paper 7 and 12 focus on adversarial attacks.

So possible attributes could be:

- "Explaining Machine Learning Techniques": This is common across papers 1, 4, 8, 12, etc.
- "Adversarial Examples Generation": Papers 4, 5, 8, 11, 12 all talk about this.
- "Machine Learning Models Used": Papers 1, 2, 5, 7, 8, 9, 10, 12 all use ML in their work.

I need to map each of these to their definitions, output formats, and values. For example, "Explaining Machine Learning Techniques" would have a definition that says it's a method to interpret black-box decisions, and output format numbers or strings.

Now, create the JSON structure with these attributes. Make sure to follow the required format and include all papers.

</think>

```
““json
{
  "ExplainableMachineLearningTechniques":
  {
    "definition": "This attribute refers
to the use of explainable
techniques such as feature
```

attribution to interpret the black-box decisions of deep neural networks, which are central to this work.”,

```
"output_format": "Numbers or strings
representing values from 1 to 100
(e.g., 90 for 90% accuracy)"
```

```
},
```

```
"AdversarialExamplesGeneration": {
  "definition": "This attribute refers
to the generation of adversarial
samples that fool a model, which
is a core focus of this paper.",
  "output_format": "A string indicating
the percentage of generated
samples that were misclassified (e
.g., "98.9%") or a numerical value
for exact rate (e.g., 98.9)"
```

```
},
```

```
"AdversarialAttackMethods": {
  "definition": "This attribute
represents the attack techniques
used, such as COPYCAT, to bypass
traditional detection systems.",
  "output_format": "A string indicating
the name or type of method used, e
.g., 'COPYCAT' or 'GAN'"
```

```
}
```

```
}
““
```