ConfReady: A RAG based Assistant and Dataset for Conference Checklist Responses

Michael Galarnyk[1*✉] **Rutwik Routu**[2*†] **Vidhyakshaya Kannan**[3*†] **Kosha Bheda**[1]
Prasun Banerjee[1] **Agam Shah**[1] **Sudheer Chava**[1]

[1] Georgia Institute of Technology [2] Duke University [3] Sai University
✉ Corresponding author: mgalarnyk3@gatech.edu

Abstract

The ARR Responsible NLP Research checklist website states that the "checklist is designed to encourage best practices for responsible research, addressing issues of research ethics, societal impact and reproducibility." Answering the questions is an opportunity for authors to reflect on their work and make sure any shared scientific assets follow best practices. Ideally, considering a checklist before submission can favorably impact the writing of a research paper. However, previous research has shown that self-reported checklist responses don't always accurately represent papers. In this work, we introduce ConfReady, a retrieval-augmented generation (RAG) application that can be used to empower authors to reflect on their work and assist authors with conference checklists. To evaluate checklist assistants, we curate a dataset of 1975 ACL checklist responses, analyze problems in human answers, and benchmark RAG and Large Language Model (LM) based systems on an evaluation subset. Our code is released under the AGPL-3.0 license on GitHub, with documentation covering the user interface and Python package.

1 Introduction

In order to submit a paper to conferences under the Association for Computational Linguistics like ACL, COLING, CoNLL, EMNLP, and NAACL, authors are required to submit their answers to the ARR Responsible NLP Research checklist[1]. The checklist was mostly developed through a combination of the NLP Reproducibility Checklist (Dodge et al., 2019), the reproducible data checklist (Rogers et al., 2021), and the NeurIPS 2021 Paper Checklist Guidelines[2]. The goal of this process is to address reproducibility, societal impact, and potential ethical issues. Authors are expected to discuss limitations, artifact usage, computational details, human involvement, and use of AI assistants. Starting with EMNLP 2025, checklist responses will be published as appendices alongside accepted papers[3], in order to "help with transparency" and encourage authors to "think more carefully about these issues when they know their answers will be visible to the broader community."

This follows an earlier pilot at ACL 2023, where checklist responses—covering 19 questions per paper—were appended to accepted submissions. For example, question A2 asks: *"Did you discuss any potential risks of your work?"* If authors respond "yes," they must cite the relevant section; if "no," they are expected to provide a justification. However, prior work has noted cases of low-effort or bad-faith responses, such as identical answers across questions and falsely reporting code availability (Magnusson et al., 2023). The ACL 2023 program chairs suggested that checklist sloppiness correlates with sloppiness elsewhere in the work and a lower acceptance rate (Rogers et al., 2023).

To mitigate unreliable checklist answers, conferences have explored Large Language Model (LM) based systems to assist authors (Goldberg et al., 2024), though these were not evaluated against human-written checklist submissions and do not reflect the full complexity of real submissions. To address this, LMs can be augmented with tools like retrieval-augmented generation (RAG), which integrates information retrieval with generative models (Lewis et al., 2020). This approach improves accuracy and relevance, especially for question-answering tasks requiring up-to-date or domain-

* These authors contributed equally.
† Work done as a Volunteer Research Assistant at Georgia Institute of Technology.
[1] https://aclrollingreview.org/responsibleNLPresearch/
[2] https://neurips.cc/Conferences/2021/PaperInformation/PaperChecklist
[3] https://aclrollingreview.org/responsible-nlp-checklist-appendices

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 667–677
November 4-9, 2025

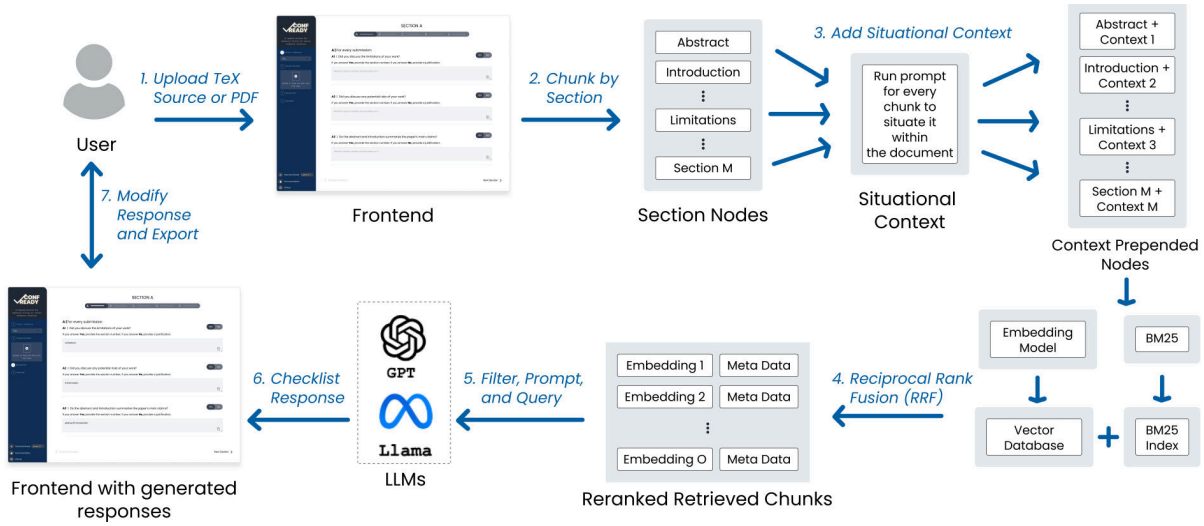

Figure 1: Users can upload either TeX source (single .tex file or a zipped folder) or a PDF to the frontend to receive an LM-generated checklist response, which can then be modified and exported.

specific knowledge (Karpukhin et al., 2020).

In this work, we introduce **ConfReady**, a RAG tool that helps with checklist responses grounded in their paper's TeX source or PDF[4]. To evaluate ConfReady, we compile the ConfReady dataset of real checklist submissions (see Section 3) and evaluate its outputs against human-written answers (see Section 4).

Our contributions are the following:

- **ConfReady Tool**: An end-to-end system for generating checklist responses from TeX source (single .tex file or zipped folder) or PDF, with a user-friendly interface and pip-installable backend.
- **Checklist Dataset**: A structured dataset of 1975 ACL papers with parsed checklist responses and metadata, enabling analysis and benchmarking.
- **Backend Evaluation**: A comparison of RAG and LM only backends on 93 ACL papers, with accuracy measured against human-provided checklist responses.

2 ConfReady

Figure 1 presents the ConfReady pipeline. Users start by uploading either TeX source (single .tex or zipped folder) or a PDF on the frontend. The system then processes the document through a seven-step workflow: (1) upload input, (2) chunk by section, (3) add situational context to each section, (4) perform Reciprocal Rank Fusion (RRF), (5) construct and send prompts to the LM, (6) generate checklist responses, and (7) allow users to review, edit, and export the results. The user-facing interface, shown in Figure 2, supports this workflow with features like section-level navigation, editable response fields, and progress tracking. Appendix A details the design rationale behind each interface component.

2.1 Parsing, Chunking, and Embedding

Parsing Users can upload either TeX Source (single .tex file or zipped folder) or PDF. When TeX Source is provided, the document is parsed to remove comments and pre-abstract content. For PDFs, a simplified pipeline is applied.

Contextual Chunking ConfReady uses contextual chunks (Anthropic, 2024) to reduce retrieval failure rates. Each chunk is annotated with metadata (e.g., section title, neighboring chunk identifiers), and a short LM-generated situational summary is prepended to the chunk before embedding using the following prompt:

> **Prompt:** Provide a concise (50–100 tokens) situational summary for this chunk, capturing its role in the larger section.

Embeddings The enriched chunks (metadata, situational summary, original text) are then converted

[4]A video demonstration is available at https://youtu.be/sNhpKJLfArc?si=0CMCe1nEFwFFUibw, with documentation and a pip-installable package at https://confready-docs.vercel.app.

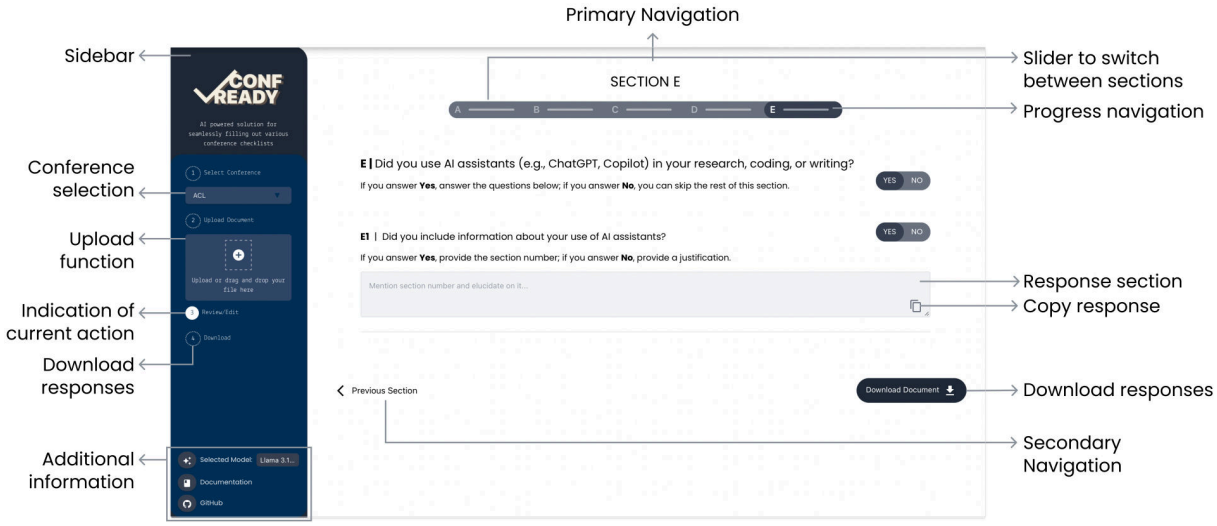

Figure 2: Features of the ConfReady user interface include: an upload function within the sidebar, primary navigation with a slider to switch between sections and progress navigation, and a generated response field with a copy function. The design rationale behind the features is listed in Appendix A.

into vector representations and stored locally. The embedding backend is modular and can be configured to support different models.

2.2 Retrieval, Fusion, and Reranking

Retrieval Our retrieval workflow is designed to combine the strengths of dense vector embeddings and sparse lexical search.

Dense Retrieval with Embeddings Each chunk is embedded using OpenAI's `text-embedding-3-large`. Queries are also embedded and compared against the document embeddings using cosine similarity.

Sparse Retrieval with BM25 In parallel, we build a lexical index over all contextualized chunks using the BM25 algorithm. This sparse retrieval step is especially effective at capturing exact matches, uncommon terminology, and keyword-based relevance that dense models may miss.

Score Fusion (Vector + BM25) To balance semantic and lexical relevance, we perform RRF on the top results from both the dense and sparse retrieval components.

LM-Based Reranking The fused results are passed through a reranking module, which uses an LM to evaluate the relevance of each chunk in the context of the original query. The top-k chunks from this reranking step are selected as final inputs for generation.

CRAG vs. NRAG We refer to the full workflow described above—retrieval, fusion, and reranking with contextual chunking—as CRAG, short for Contextual Retrieval-Augmented Generation (Anthropic, 2024). For comparison, we also define a baseline, NRAG (Naive Retrieval-Augmented Generation), which uses basic document chunks in place of contextual ones.

2.3 Prompt Design

ConfReady uses modular instructions that can be adapted to different conference checklists and checklist versions. The ACL 2023 checklist, for instance, contained 19 questions labeled A1–D5. Each question is mapped to a dedicated prompt with a uniform structure: Introduction, Question, Additional Context, and Output Structure. Figure 3 shows the prompt for A1 (*"Did you discuss the limitations of your work?"*).

The prompt is designed to provide the LM with the same information humans should consider when answering the question. The "Question" corresponds to an individual question in the checklist. The "Additional Context" is information provided from Guidelines for Answering Checklist Questions on the ACL Rolling Review website[6].

The *Output Structure* specifies that the response should be a JSON object with `answer`, `section name`, and `justification` as keys. Restricting output to JSON has been shown to improve clas-

[6] https://aclrollingreview.org/responsibleNLPresearch/

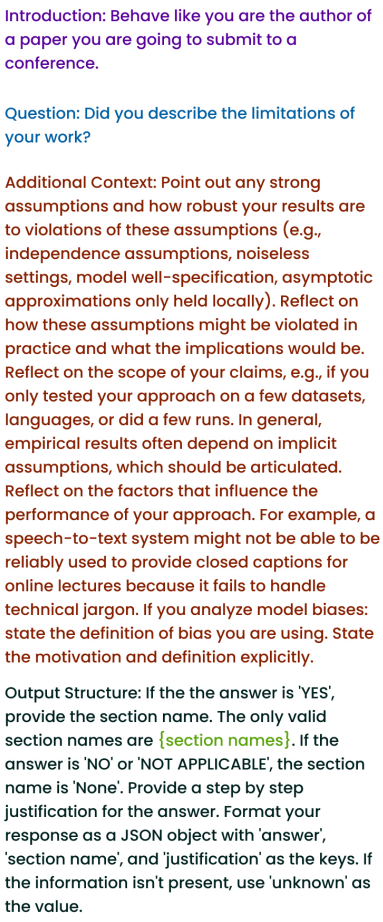

Figure 3: Example prompt for question A1. *Introduction*: instructs the LM to assume the role of an author. *Question*: the checklist item the LM must address. *Additional Context*: provides supporting guidance drawn from checklist documentation. *Output Structure*: instructs the LM to return JSON with fields for `answer`, `section name`, and `justification`. *Section Names*: lists valid section names parsed directly from the parsed files.

sification accuracy and reproducibility (Tam et al., 2024; Es et al., 2024). Section names are taken directly from the parsed files, so the LM is limited to choosing among sections that appear in the paper.

2.4 Frontend with Generated Responses

After inference, the LM output is passed to the frontend in a structured JSON format with three fields: `answer`, `section name`, and `justification`. When the answer is `"Yes"`, the corresponding section name is displayed in the interface (Figure 2). When the answer is `"No"`, the section name defaults to `"unknown"`, and the justification is shown instead.

Backend Hallucination Mitigation RAG applications are known to reduce hallucination inherent in LMs (Shuster et al., 2021). To further mitigate the two hallucination categories, factuality and faithfulness, outlined by Huang et al. (2023), the application implements several safeguards. First, to address instruction inconsistency (a type of faithfulness hallucination) where the LM deviates from the instruction to return a single section, the system withholds a response. Second, because section names are parsed directly from the paper, only those names are allowed as valid answers.

User Checklist Modification LM-generated answers are intended to assist authors rather than replace their judgment. Users must review each response for accuracy, and questions concerning the use of AI assistants in research, coding, or writing must be answered manually. As a final safeguard, authors are required to validate responses before export. The validated checklist can then be exported as a Markdown document. Markdown was chosen because it is lightweight, easy to convert into other formats (e.g., PDF and TeX), and widely adopted in open-source ecosystems such as GitHub READMEs and Hugging Face model cards (Yang et al., 2024).

2.5 System Architecture

User Interface The frontend is built with React[7], a JavaScript library for creating modular and interactive web applications. For consistent and responsive styling across devices, we use TailwindCSS[8], a utility-first CSS framework that accelerates UI development with predefined class utilities.

API Orchestration and Backend Workflow The backend is implemented using Flask[9], a lightweight Python web framework that manages communication between the frontend and backend. It handles file uploads, runs processing scripts (e.g., TeX parsing), and orchestrates interactions with the RAG pipeline. RAG is implemented by LlamaIndex (Liu, 2022), which integrates external context into the LM inference. To maintain real-time communication between the backend and frontend, a server-side event endpoint on the Flask server streams updates to the client during critical stages of the file processing workflow.

[7] https://reactjs.org
[8] https://tailwindcss.com/
[9] https://flask.palletsprojects.com/en/3.0.x/

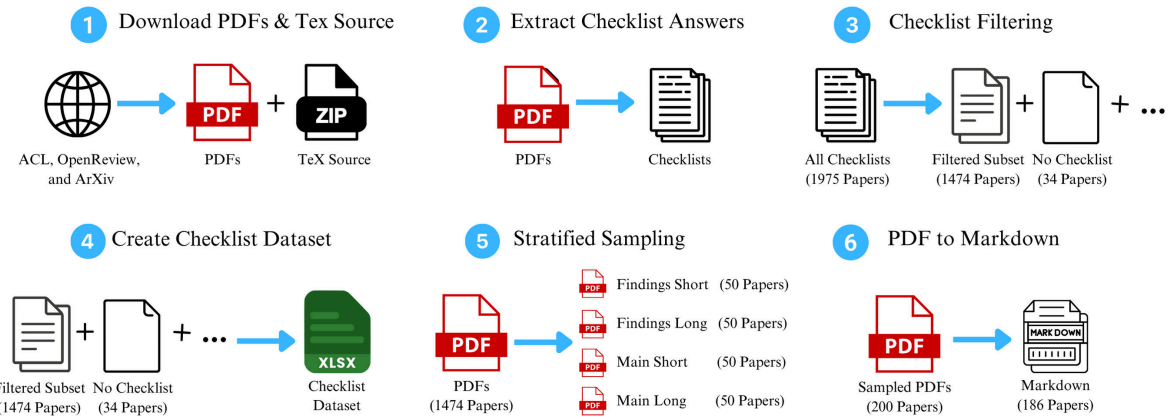

Figure 4: Checklist dataset generation pipeline with six stages: (1) download PDFs and TeX sources for each paper, (2) extract and verify checklist responses, (3) analyze and filter out problematic checklists (e.g., blank or missing responses), (4) organize the remaining checklists and metadata into a structured Excel file, (5) apply stratified sampling across categories, and (6) convert the sampled PDFs into Markdown using the `marker` library[5] for evaluation.

3 ConfReady Dataset

To evaluate ConfReady and understand how authors engage with checklist questions in practice, we curate a dataset of 1975 ACL 2023 papers, the most recent major Association for Computational Linguistics venue to publish checklists alongside accepted submissions. While prior work has documented issues with checklist reliability—such as identical answers across questions or unsupported claims of code availability (Magnusson et al., 2023)—no prior work has open-sourced a large dataset of real, author-written checklist responses. Our dataset addresses this gap, supporting both large-scale analysis of checklist quality and benchmarking of RAG and LM systems.

Figure 4 illustrates the six-stage pipeline used to construct the dataset: (1) download PDFs and arXiv TeX sources for each paper, (2) extract and verify checklist responses, (3) analyze checklists and create a filtered subset without issues (e.g., blank, missing, etc.), (4) organize the remaining checklists and metadata into a structured Excel file, (5) apply stratified sampling to select 50 papers from each category (ACL Findings Short, Findings Long, Main Short, Main Long), and (6) convert the sampled PDFs into Markdown using the `marker` library[10], leaving 186 papers after excluding a small number with conversion failures. Due to this, ConfReady prefers TeX source input when available.

Dataset Statistics To assess the quality and consistency of checklist submissions, we conducted a detailed analysis of all 1975 papers. Table 1 summarizes key statistics, revealing several notable inconsistencies. Some papers omitted checklists entirely (*No Checklist*), while others appended blank templates (*Blank Checklist*) or didn't reference a single section in their responses (*No Section Names*). Notably we also had relatively few disclosures about AI use in writing (*AI Use in Writing*). Finally, to support evaluations of RAG and standalone LM backends on TeX, we also filtered out papers without corresponding arXiv TeX sources.

	Findings		Main	
Metric	Short	Long	Short	Long
All Collected Papers				
Total Papers	189	712	164	910
No Checklist	6	11	3	14
Blank Checklist	8	36	10	60
All Yes Responses	5	7	2	9
No Section Names	7	15	3	28
AI Use in Writing	13	39	11	60
Not on arXiv	50	190	37	190
Evaluation Sample				
Total Papers	46	43	47	50
Avg Tokens (Paper)	12563	19865	14049	24547

Table 1: Summary statistics grouped by track (Findings/Main) and paper length (Short/Long).

Token Length Analysis Figure 5 presents token count distributions across the four evaluation subsets. On average, ACL Main papers are longer than ACL Findings papers in both short and long categories.

[10] https://github.com/datalab-to/marker

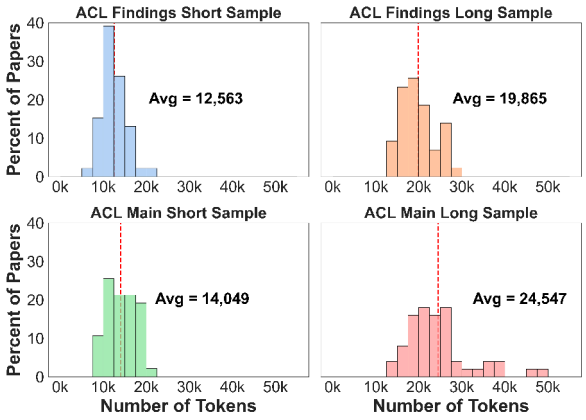

Figure 5: Token count distributions for ACL 2023 papers across four categories in our evaluation subset: ACL Findings Short, ACL Findings Long, ACL Main Short, and ACL Main Long. The y-axis indicates the percentage of papers; the x-axis shows token counts (measured using the Llama-3 tokenizer). Red dashed lines indicate mean token counts.

Final Structure The ConfReady dataset preserves parsed checklist question–answer pairs, justification text, referenced sections, and metadata flags indicating issues such as incomplete or blank submissions. Further details on the data collection and analysis process are provided in Appendix C.

4 Evaluation

	Findings	Main
Model	Long	Long
RAG Framework on TeX		
CRAG (Llama-3.1-405B)	81.72	81.93
CRAG (Llama-3.3-70B)	78.07	78.78
CRAG (GPT-4o)	80.58	79.86
NRAG (Llama-3.1-405B)	78.44	77.36
NRAG (Llama-3.3-70B)	74.64	73.65
NRAG (GPT-4o)	80.43	73.22
LM on TeX		
Llama-3.1-405B	78.87	75.83
Llama-3.3-70B	78.69	79.20
GPT-4o	80.54	78.45
LM on MD		
Llama-3.1-405B	79.86	77.26
Llama-3.3-70B	80.97	81.09
GPT-4o	82.27	77.38

Table 2: Accuracy comparison of RAG, LMs on TeX, and LMs on PDFs for ACL Main (Long) and ACL Findings (Long) papers.

We evaluate Llama-3.1–405B, Llama-3.3–70B (Meta et al., 2024), and GPT-4o (OpenAI, 2023a) on the evaluation sample. Due to compute limits, experiments focus on long-form ACL submissions.

Models are tested in three settings: (i) RAG on TeX with CRAG and NRAG, (ii) LM on parsed TeX, and (iii) LM on MD. Human-annotated answers serve as references, allowing us to evaluate how effectively models can reflect on ethical considerations, reproducibility, and societal impacts in each setup.

Results CRAG consistently outperforms NRAG, confirming the benefit of section-aware retrieval, RRF, and LM reranking (see Table 2). LM on Markdown performs competitively with LM on TeX, and in some cases better, echoing previous work showing that structured Markdown can improve model fidelity (Min et al., 2024; Jain et al., 2025; Galarnyk et al., 2025). Nevertheless, ConfReady favors TeX input with RAG, since TeX parsing better preserves section structure and reduces extraction errors, whereas PDF-to-Markdown conversion is prone to failures that can disrupt retrieval and alignment. Finally, GPT-4o and other models often perform better on Findings than on Main papers, suggesting that longer Main submissions (see Figure 5) introduce added difficulty for checklist answering.

Error Analysis Figure 6 shows accuracy by checklist question for ACL Findings Long. Question C1—*"Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?"*- had the lowest LM accuracy. Appendix B presents typical failure cases on C1. A closer review of these disagreements revealed a recurring pattern: LMs often answered "NO" unless all three components were explicitly mentioned, while human justifications typically cited only section references (e.g., "Section 4" or "Appendix C1") without clarifying what details were provided. This highlights a stricter model standard for completeness versus more lenient human interpretations.

5 Conclusion and Future Work

This paper introduces ConfReady, a LM-based system which can be used to empower authors to reflect on their work and act as an assistant to help authors with conference checklists. With ConfReady, authors can get a LM checklist response that they use to reflect on their work or modify it

before submitting. We hope that the open-source application will be responsibly used as an assistant and tool for reflection. As a future work, we are still working on improving the project across several different directions:

- **Local LMs**: While ConfReady currently uses commercial providers to avoid the overhead of self-hosting, future versions will support local open-weight models to enable private, offline usage in settings where data sensitivity or API constraints are a concern.
- **Other Conference Checklists**: ConfReady currently supports checklists from conferences under the Association for Computational Linguistics (e.g., ACL, COLING, CoNLL, EMNLP, and NAACL). While NeurIPS support has been implemented, adapting ConfReady to other venues will require adjustments for different checklist structures and question formats.

Ethics Statement

Structured Output Format A major issue with incorporating LMs into applications is their failure to follow output format inconsistency (faithfulness hallucination). We mitigate this by requiring responses in JSON format, similar to the JSON mode in the OpenAI and Gemini APIs (Gemini Team et al., 2024). Additionally, some libraries such as Instructor[11] also require JSON.

User Reliance and Scope Analysis of ChatGPT usage shows that non-work messages now comprise over 70% of interactions, up from 53% in 2024 (Chatterji et al., 2025). This reflects a broadening role for LMs in everyday life, including learning and creative expression, beyond their original productivity-focused scope. At the same time, models can provide fluent but incomplete or outdated answers when knowledge falls outside their training data (Shah et al., 2025). These patterns highlight the importance of using ConfReady as an aid for reflection and editing, not as a replacement for author responsibility.

References

Saleh Afroogh, Ali Akbari, Emmie Malone, Mohammadali Kargar, and Hananeh Alambeigi. 2024. Trust in ai: progress, challenges, and future directions. *Humanities and Social Sciences Communications*, 11(1):1–30.

Anthropic. 2024. Introducing contextual retrieval. Accessed: 2025-07-01.

Aaron Chatterji, Tom Cunningham, David Deming, Zoë Hitzig, Christopher Ong, Carl Shan, and Kevin Wadman. 2025. How people use chatgpt. Technical report, OpenAI & Harvard University.

Frederick G. Conrad, Mick P. Couper, Roger Tourangeau, and Andy Peytchev. 2010. The impact of progress indicators on task completion. *Interacting with Computers*, 22(5):417–427.

Meredith Davis and Jamer Hunt. 2017. *Visual communication design: An introduction to design concepts in everyday experience*. Bloomsbury Publishing.

Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. 2019. Show your work: Improved reporting of experimental results. 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 2185–2194, Hong Kong, China. Association for Computational Linguistics.

Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. RAGAs: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158, St. Julians, Malta. Association for Computational Linguistics.

Michael Galarnyk, Agam Shah, Dipanwita Guhathakurta, Poojitha Nandigam, and Sudheer Chava. 2025. How inclusively do LMs perceive social and moral norms? In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4859–4869, Albuquerque, New Mexico. Association for Computational Linguistics.

Gemini Team et al. 2024. Gemini: A family of highly capable multimodal models.

Alexander Goldberg, Ihsan Ullah, Thanh Gia Hieu Khuong, Benedictus Kent Rachmat, Zhen Xu, Isabelle Guyon, and Nihar B. Shah. 2024. Usefulness of llms as an author checklist assistant for scientific papers: Neurips'24 experiment.

Mariam Guizani. 2022. A decade of information architecture in hci: A systematic literature review. *arXiv preprint arXiv:2202.13412*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions.

[11] https://github.com/instructor-ai/instructor

Arihant Jain, Purav Aggarwal, and Anoop Saladi. 2025. AutoChunker: Structured text chunking and its evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 983–995, Vienna, Austria. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474.

Jerry Liu. 2022. LlamaIndex.

Ian Magnusson, Noah A. Smith, and Jesse Dodge. 2023. Reproducibility in NLP: What have we learned from the checklist? In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12789–12811, Toronto, Canada. Association for Computational Linguistics.

Meta et al. 2024. The llama 3 herd of models.

Dehai Min, Nan Hu, Rihui Jin, Nuo Lin, Jiaoyan Chen, Yongrui Chen, Yu Li, Guilin Qi, Yun Li, Nijun Li, and Qianren Wang. 2024. Exploring the impact of table-to-text methods on augmenting LLM-based question answering with domain hybrid data. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, pages 464–482, Mexico City, Mexico. Association for Computational Linguistics.

Pawarat Nontasil and Chatpong Tangmanee. 2024. Investigating the impact of progress indicator design on user perception of delay. *Journal of System and Management Sciences*, 14:333–344.

OpenAI. 2023a. Gpt-4 technical report. Technical report, OpenAI. Available at https://doi.org/10.48550/arXiv.2303.08774.

Anna Rogers, Timothy Baldwin, and Kobi Leins. 2021. 'just what do you think you're doing, dave?' a checklist for responsible data use in NLP. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4821–4833, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Anna Rogers, Marzena Karpinska, Jordan Boyd-Graber, and Naoaki Okazaki. 2023. Program chairs' report on peer review at acl 2023. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages xl–lxxv, Toronto, Canada. Association for Computational Linguistics.

Agam Shah, Liqin Ye, Sebastian Jaskowski, Wei Xu, and Sudheer Chava. 2025. Beyond the reported cutoff: Where large language models fall short on financial knowledge.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhi Rui Tam, Cheng-Kuang Wu, Yi-Lin Tsai, Chieh-Yen Lin, Hung yi Lee, and Yun-Nung Chen. 2024. Let me speak freely? a study on the impact of format restrictions on performance of large language models.

Xiao Yang, Wei Liang, and Jie Zou. 2024. Navigating dataset documentations in ai: A large-scale analysis of dataset cards on huggingface. In *Proceedings of The Twelfth International Conference on Learning Representations*. ICLR.

A Features of the User-Interface

The features incorporated into the user interface, along with the underlying rationale for their design, are detailed below. These design decisions aim to enhance usability, ensure accessibility, and support an intuitive and efficient user experience throughout the checklist completion workflow.

1. *Side Bar/Upload:* The side bar incorporates the visual identity of the platform. It has been visualized to resemble file tabs to help users connect with the overarching action being performed using visual connotation (Davis and Hunt, 2017). It contains the upload function which allows users to upload their paper's TeX source (single .tex file or zipped folder) and visual indication of the user's current action. The bottom of the sidebar contains a model selector and links to the documentation[12] and GitHub[13] placed according to information hierarchy principles (Guizani, 2022).

2. *Conference Selection:* Users select a conference checklist. Currently, the platform allows users to select from ACL checklists, NeurIPS, and NeurIPS Datasets and Benchmarks (NeurIPS D&B).

[12] https://confready-docs.vercel.app/docs/walkthrough

[13] https://github.com/gtfintechlab/ConfReady

3. *Primary navigation:* The top bar of the interface provides the users with functionality of switching between sections. It also indicates the progress for each section, keeping the users informed through visually represented data (Nontasil and Tangmanee, 2024).

4. *Secondary navigation:* To refrain from disrupting the user's workflow while performing important tasks like checking or editing responses, the secondary navigation allows movement to the next page without needing to return to the primary navigation. The intention with this navigation is reducing extraneous cognitive overload.

5. *Response sections:* Responses are filled in and users need to verify responses.

6. *Download:* Only after users have reviewed each section are they allowed to download all of their responses from either the sidebar or from the Download button in the final section. The intention is to encourage users to be responsible for verification of AI-driven results (Afroogh et al., 2024).

The ConfReady user journey is shown in the ConfReady documentation[14]. To use the application, users upload the TeX source (single .tex file or zipped folder) or PDF. Next, in order to enhance task completion (Conrad et al., 2010), a progress screen appears to let users know of the backend RAG progress. This feature was added to the platform after informal interviews where it was noted that users wanted to get some indication on how long they needed to wait before they can check/edit responses, and download results.

B Qualitative Analysis of Checklist Responses

We provide qualitative insights into system behavior on ACL 2023 checklists. Details of dataset construction and statistics are provided in Appendix C.

B.1 Discrepancies on Question C1

Question C1 of the ACL checklist asks: *"Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?"* To better understand the low accuracy on this question, we reviewed a sample of cases where the LM response diverged from the human-provided checklist. Table 3 shows representative examples. In all cases, human annotators answered "YES" while the LM answered "NO." This reflects a recurring failure mode: the LM applied a stricter standard, requiring explicit mentions of model parameters, compute budget, and infrastructure to justify a "YES," while human authors often responded affirmatively based on partial evidence.

Notably, compute resource reporting was almost always absent. The LM typically acknowledged some modeling information but still answered "NO" if all three elements were not present. This conservative behavior highlights a mismatch between human and model expectations of completeness.

B.2 ACL Edge Cases

While the ConfReady system accurately processed most ACL submissions, several edge cases highlighted current limitations. One issue involved exceeding the model's context length limit, leading to inference failures. Additionally, some model-generated responses were phrased awkwardly or contained placeholder-like text (e.g., "x and y should be..."), suggesting room for prompt refinement or output filtering.

C Checklist Dataset Construction

We describe the pipeline used to construct the structured checklist dataset.

C.1 Data Collection and Extraction

Due to the scale of the dataset, most of the collection was automated using Python scripts. The multi-stage pipeline proceeded as follows:

1. **Link Retrieval** Automated scripts using `requests` and `BeautifulSoup` were used to identify and retrieve arXiv links as well as ACL Anthology PDF links for each paper.

2. **PDF Retrieval and Preprocessing** Normalized PDF links were obtained from ACL Anthology URLs. To extract checklists, only the last two pages of each ACL paper were parsed using `PyPDF2`, since checklists consistently appeared there.

3. **Checklist Parsing** Each checklist was parsed using conference-specific regex templates.

[14] https://confready-docs.vercel.app/docs/walkthrough

Human Response	LM Response	Human Justification	LM Justification
Yes	No	Section A.1	The text does not discuss the experimental setup, including hyperparameter search and best-found hyperparameter values. While it mentions experiments and evaluations of different models, it does not provide details on how hyperparameters were tuned or selected.
Yes	No	Appendix D	The paper does not report the number of parameters in the models used, the total computational budget, or computing infrastructure used. Although it mentions the use of specific models like BERT and the number of GPUs used for training, it does not provide detailed information about the model sizes or computational resources.
Yes	No	Section 3.2	The text does not mention the number of parameters in the models used, the total computational budget (e.g., GPU hours), or computing infrastructure used. The models used are RoBERTa-large models, but the text does not provide information about the number of parameters or computational resources used.
Yes	No	Please see Section 4.1.	The text does not mention the number of parameters in the models used, the total computational budget (e.g., GPU hours), or computing infrastructure used.

Table 3: Discrepancies on checklist question C1 in ACL Findings Long. The justifications provided are from CRAG and MD outputs generated using Llama 3.1 405B. All example have human answers "Yes" and LM answers "No," reflecting stricter model standards for reporting parameters, compute budget, and infrastructure.

ACL checklists featured labeled questions (e.g., A1) with symbolic ticks and optional justifications; answers were matched using proximity-based heuristics.

4. **Structured Storage** The extracted checklist information was standardized, manually reviewed, and stored in structured Excel sheets for incremental updates. Each paper was also mapped to its arXiv `.tar.gz` TeX source to allow linking between TeX and PDF files.

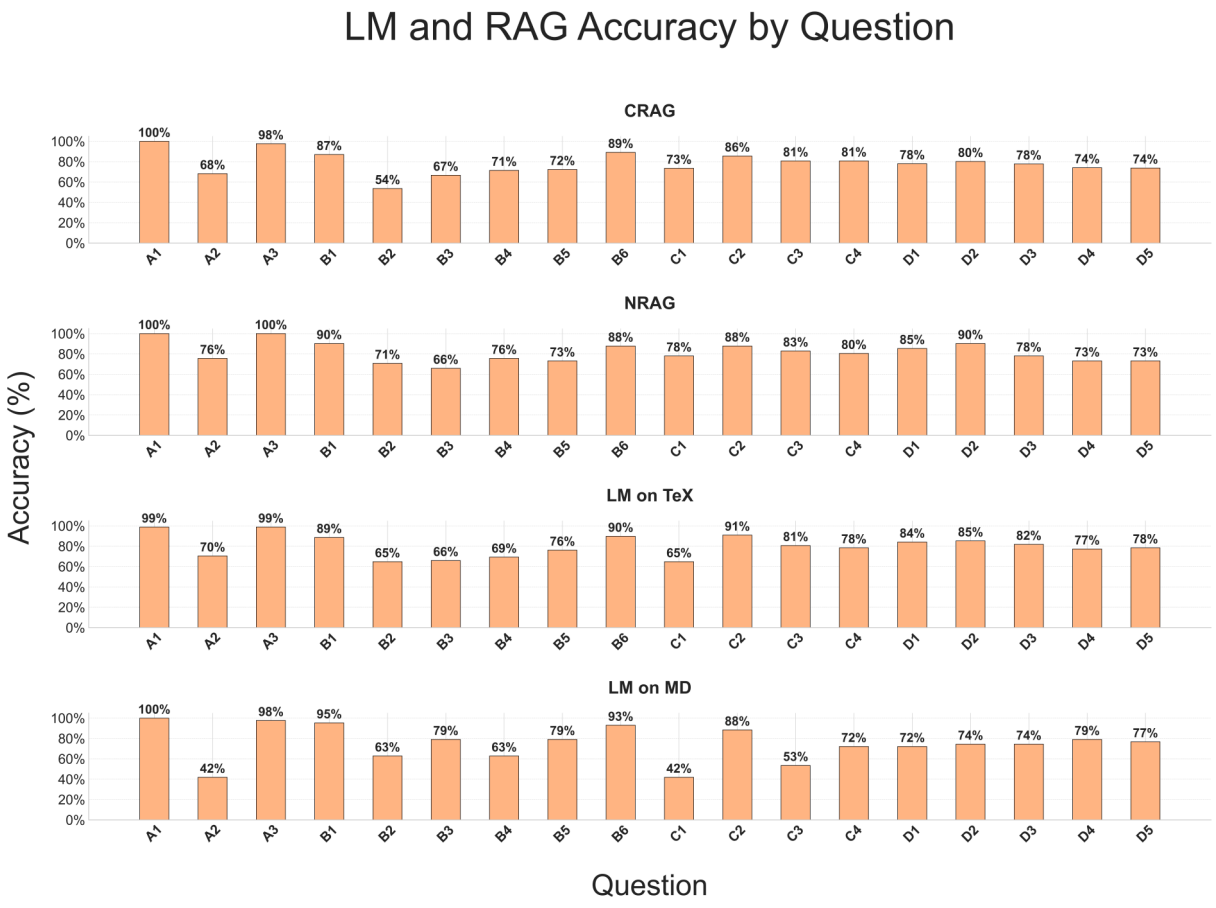

Figure 6: Accuracy by checklist question for ACL Findings Long on Llama-3.1-405B across four setups: CRAG, NRAG, LM on TeX, and LM on MD.