ConfReady: RAG Based Assistant for Conference Checklists

Michael Galarnyk* **Rutwik Routu*** **Vidhyakshaya Kannan** **Kosha Bheda**
Prasun Banerjee **Agam Shah** **Sudheer Chava**
Georgia Institute of Technology

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 quantitatively evaluate the checklist assistants responses against human checklist responses, we created a dataset of ACL checklist responses from 1975 papers. Our code is released under the AGPL-3.0 license on GitHub, with documentation covering the user interface and pip-installable Python package available on the project documentation site.

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."

The ACL checklist consists of up to 19 questions about the paper. For example, question A2 is the following: "*Did you discuss any potential risks of your work?*" If the answer is yes, the authors must provide the section number where the risks are discussed. If the answer is no, authors need to provide a reasonable justification. However, Magnusson et al. (2023) reported some bad faith responses to checklist questions such as submitting identical answers to each question and falsely reporting code availability.

In order to mitigate the issue of unreliable checklist answers being submitted, conferences have started to explore the use of Large Language Models (LMs) to improve checklist compliance before submission (Goldberg et al., 2024). While LMs like GPT-4 (OpenAI, 2023a) and Llama-3 (Touvron et al., 2023) have shown to be good at the question answering tasks (Wei et al., 2024; Kojima et al., 2024), LMs are known to hallucinate (Huang et al., 2023). To mitigate this issue, LMs can be augmented with external 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-specific knowledge (Karpukhin et al., 2020).

We introduce **ConfReady**, a retrieval-augmented generation (RAG) tool that helps

* These authors contributed equally to this work

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

[2] https://neurips.cc/Conferences/2021/PaperInformation/PaperChecklist

[3] https://aclrollingreview.org/responsible-nlp-checklist-appendices

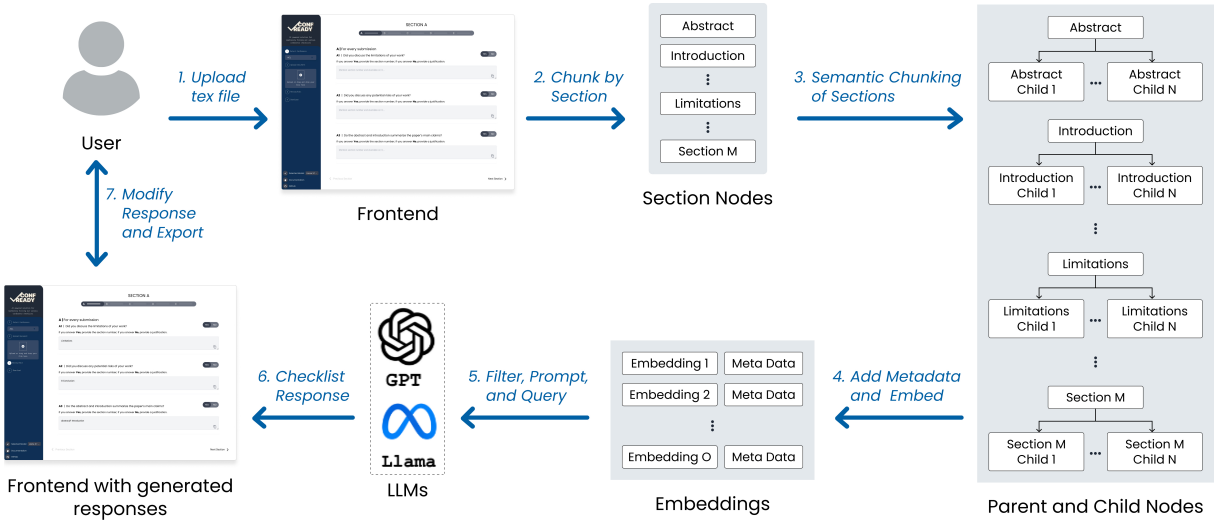

Figure 1: Users can upload TeX Source (TeX file or zipped folder) to the frontend to receive an LM-generated checklist response, which can then be modified and exported.

authors draft Responsible NLP checklist responses grounded in their paper's TeX source.[4] To evaluate ConfReady, we compile a dataset of real checklist submissions (see Section 3) and benchmark its outputs against human-written answers (see Section 4).

Our contributions are:

- **ConfReady Tool**: A RAG-based system for generating grounded responses to Responsible NLP checklist questions.
- **Checklist Dataset**: A structured dataset of 1975 + ACL and NeurIPS papers with parsed checklist responses and metadata.
- **Evaluation**: We benchmark ConfReady against standalone LLMs, finding that RAG-based systems produce more reliable, less hallucinated answers.

2 ConfReady

The ConfReady tool depicted in Figure 1, operates as follows: (1) the user uploads TeX Source (TeX file or zipped folder), (2) the file is chunked by section, (3) each section is semantically chunked, (4) metadata is added and text is embedded, (5) filtering, prompting, and querying occur, (6) LM-generated checklist responses are sent to the frontend, and (7) the user modifies and exports the responses.

[4]A video demonstration is available at https://youtu.be/ZLtdtoR75GU, with documentation and a pip-installable package at https://confready-docs.vercel.app.

2.1 Parsing, Chunking, and Embedding

Parsing After users upload their paper's TeX Source (TeX file or zipped folder), the document is parsed to remove all comments and all text before the abstract. Additionally, sections like acknowledgments are removed. For figures and tables, only captions are kept.

Maintaining relationships during chunking In order to best utilize the original structure of TeX documents while making it easier for the LM to distinguish between sections, the chunking process is as follows:

1. Every section is chunked into its own node.
2. Metadata is added (section name, previous node, next node). This also makes it easier to filter out irrelevant nodes for some prompts.
3. Section nodes are broken up into parent and child chunks by semantic chunking[5]. This chunking method takes embeddings of sentences and finds breakpoints between sequential sentences using embedding similarity.

Embeddings The text is embedded. When the application is configured for OpenAI models, the default embedding is "text-embedding-ada-002". The application can also be configured to use the open source embedding "m2-bert-80M-8k-retrieval".

[5]https://docs.llamaindex.ai/en/stable/examples/node_parsers/semantic_chunking/

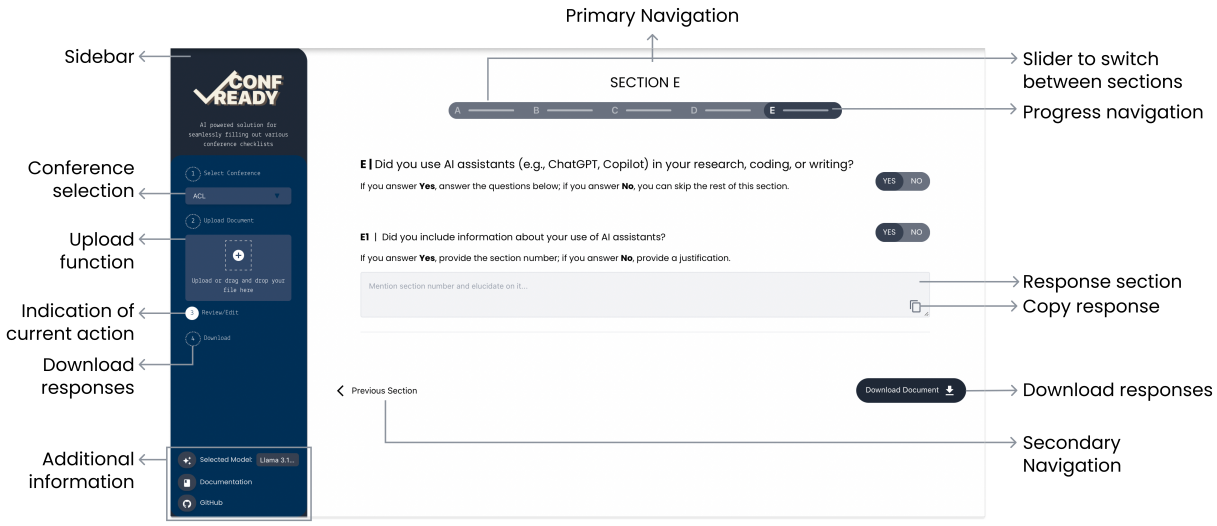

Figure 2: Features of the ConfReady user interface include: an upload function within the sidebar, primary navigation with a slider to switch between sections and progress navigation, and a generated response field with a copy function. The design rationale behind the features is listed in Appendix F.

2.2 Filter, Prompt, Query, and Reranking

Nodes that are not relevant for a specific query can be filtered. For instance, for question A3 ("*Do the abstract and introduction summarize the paper's main claims*") all nodes that are not parent or child nodes of the abstract and introduction sections can be excluded. The app uses recursive retrieval with cosine similarity as the similarity metric. Queries retrieve the smaller child chunks and follow references to the parent chunks. The parent chunks are fed into the LM.

	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 Subset				
Total Papers	46	43	47	50
Avg Tokens (Paper)	12563	19865	14049	24547

Table 1: ACL checklist dataset statistics by venue and paper length.

Prompt Design There are a total of 18 prompts for conferences under ACL. Each prompt follows a uniform structure: Introduction, Question, Additional Context, and Output Structure. For example, the question A1 prompt is shown in Appendix C.

	Findings		Main	
Model	Short	Long	Short	Long
RAG Framework (LM)				
CRAG (Llama-3.1-405B)	X	X	X	X
CRAG (Llama-3.1-70B)	X	X	X	X
CRAG (DeepSeek-R1)	X	X	X	X
CRAG (GPT-4o)	X	X	X	X
NRAG (Llama-3.1-405B)	X	X	X	X
NRAG (Llama-3.1-70B)	X	X	X	X
NRAG (DeepSeek-R1)	X	X	X	X
NRAG (GPT-4o)	X	X	X	X
LM on TeX				
Llama-3.1-405B	X	X	X	X
Llama-3.1-70B	X	X	X	X
DeepSeek-R1	X	X	X	X
GPT-4o	X	X	X	X
LM on PDF				
Llama-3.1-405B	X	X	X	X
Llama-3.1-70B	X	X	X	X
DeepSeek-R1	X	X	X	X
GPT-4o	X	X	X	X

Table 2: Performance comparison of RAG, LMs on TeX, and LMs on PDFs for ACL Findings and Main papers.

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 aclrollingreview website[6].

[6] https://aclrollingreview.org/

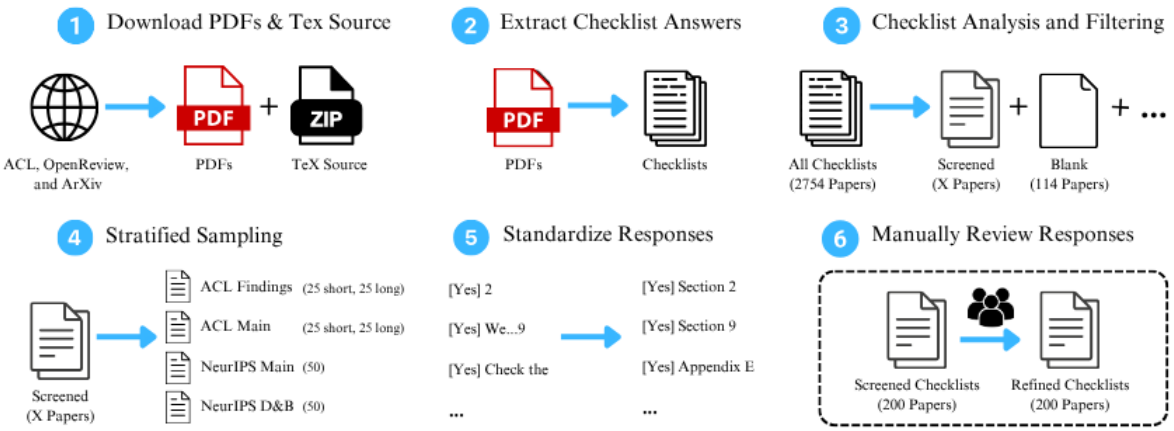

Figure 3: The Checklist dataset generation pipeline consists of six stages. (1) We download a PDF and Tex Source for each paper. (2) Checklist answers are extracted from each paper and responses are verified. (3) We analyze checklists and remove checklists with issues (e.g., blank, missing, improperly filled, etc.). (4) Stratified sampling of the screened papers resulting in 186 papers where 50 papers come from ACL Main (Long), 47 from ACL Main (Short), 43 from ACL Findings (Long), and 46 from ACL Findings (Short). (5) Standardizing response format across checklists. (6) The Checklist dataset is finalized after humans annotators review checklist responses.

The "Output Structure" specifies that the response should be a JSON object with 'answer', 'section name', and 'justification' as the keys. It has shown that JSON format restricting instructions can enhance classification task accuracy and reproducibility (Tam et al., 2024; Es et al., 2024). The section names in the prompt come from the parsed documents. They give the LM a set of valid answer choices.

Tree Summarizing During inference, the LM uses the recursive summarization method `{tree_summarize}` from LlamaIndex. It first summarizes the smaller child text chunks. These are then integrated to form summaries of larger chunks. This method can miss the finer points of the text, but our method uses metadata to mitigate this issue and avoid more computationally complex methods like Raptor which involves recursively embedding, clustering, summarizing, and constructing a tree with different levels of summarizing (Sarthi et al., 2024).

Mitigating Hallucination ConfReady is a RAG application which is known to reduce hallucination inherent in LMs (Shuster et al., 2021). In order to further mitigate the two hallucination categories, factuality and faithfulness, presented by Huang et al. (2023), the application does the following. First, in order to deal with instruction inconsistency (type of faithfulness hallucination) where the LM outputs for a question deviate from the instructions to return a single section, the software will not return a checklist response for that question. Second, since section names come from parsed documents, we only allow those section names to be answers for the ACL checklist. Finally, this application requires a human in the loop to validate answers.

2.3 Frontend with Generated Responses

After inference, the LM checklist response is sent to the frontend. This response is formatted and added to questions in the user interface shown in Figure 2. If the answer to the question is "yes", the response is formatted as "section name". If the answer to the question is "no", the response is formatted as "None. LM Generated Justification".

User Checklist Modification The LM answers are supposed to assist users with understanding their paper and simplifying the response process. Consequently, users need to check each LM generated answer for accuracy. Any question that deals with the use of AI assistants in research, coding, or writing is only to be answered by users.

Once the user is satisfied with the answers they can export the response to a markdown document. Markdown was chosen due to how easy it is to convert from markdown to other formats (e.g., PDF and LaTeX) and the widespread adoption of README markdown files on GitHub and model cards on Hugging Face (Yang et al., 2024).

`responsibleNLPresearch/`

2.4 System Architecture

User Interface The user interface of our application is developed using React[7], a JavaScript library for building interactive and component-based web applications. To ensure a visually consistent and responsive design across devices, we adopted TailwindCSS[8], a utility-first CSS framework that enables rapid UI development with predefined classes. Data management is handled using Firebase[9] Firestore, a scalable NoSQL document database, to store and quickly retrieve the checklist questions to be shown on the user interface. Overall, this architecture supports a fluid and responsive user experience by minimizing latency and providing instant feedback.

API Orchestration and Backend Workflow The backend of ConfReady is powered by Flask[10], a lightweight WSGI web application framework for Python, which serves as the central component orchestrating interactions between the frontend and backend. Flask handles core functionalities such as file uploads, orchestration of external scripts for processing TeX files, and communication with the user interface. The retrieval-augmented generation (RAG) pipeline is implemented using LlamaIndex (Liu, 2022), enabling the system to dynamically integrate external knowledge during 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, such as inferencing, chunking, and embedding. The user interface of ConfReady is shown in Figure 2

3 Checklist Datasets

To evaluate ConfReady and study how authors respond to checklist questions in practice, we constructed a structured dataset of checklist responses from 1975 papers across ACL, NeurIPS, and NeurIPS Datasets & Benchmarks (D&B) tracks. We included ACL 2023 specifically because it was the only *ACL* conference to date that required Responsible NLP Checklist answers to be published as appendices alongside accepted papers. This provided a rare opportunity to analyze author-written responses at scale and identify patterns

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

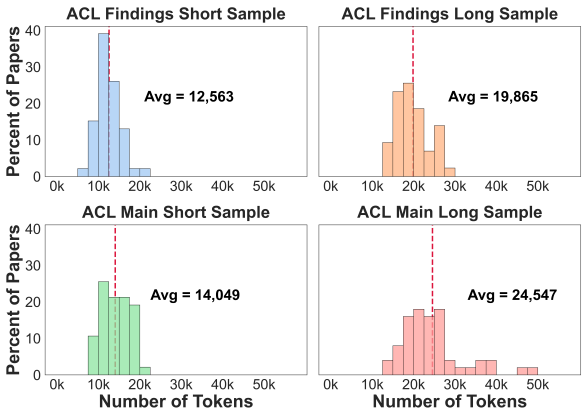

Figure 4: Token count distributions per paper using the Llama-3 tokenizer across each conference category. Red dotted line is the average.

which motivates the need for assistance tools like ConfReady.

Despite known issues with checklist reliability (Magnusson et al., 2023), we use this dataset to benchmark RAG systems and LLMs on their ability to generate grounded, complete, and well-formatted checklist responses.

For each paper, we parsed question-answer pairs, justification text, referenced section names, and metadata such as whether the response was blank, incomplete, or marked "not applicable." We retained only papers with available arXiv TeX source files to support retrieval-augmented generation. Papers were excluded if they contained blank or missing checklist responses.

This dataset supports both quantitative benchmarking and qualitative error analysis, helping assess where LLM-based systems match or exceed the quality of human-authored checklist responses—and where they still fall short.

Despite checklist unreliability Magnusson et al. (2023), we quantitatively evaluate various RAG systems and LLMs across conference response checklists collected from ACL, NeurIPS and NeurIPS D&B submissions in order to better analyze the application's checklist's responses.

The dataset was constructed using six main steps (put steps here)

The resulting dataset includes checklist responses from 1975 papers with detailed statistics summarized in 3. For each paper, we retain parsed question-answer pairs, justification text, section name references, and metadata such as whether the response was blank or incomplete.

To support LLM-based generation, we filtered

out papers without corresponding arXiv TeX sources, allowing retrieval-augmented systems to generate grounded responses. Papers were excluded if they lacked checklists, had malformed PDFs or deviated from known structural templates.

4 Evaluation

In order to quantitatively evaluate the checklist assistants against human responses, we collected checklist responses from 1975 individual papers across ACL (§A.2), NeurIPS (§A.3), and NeurIPS Datasets & Benchmarks (§A.4). Dataset statistics are reported in Table **??**.

ConfReady has been evaluated with the open source LM ("Meta-Llama-3.1-405B-Instruct-Turbo") and the closed source LM GPT-4o ("gpt-4o-2024-11-20"). Users have the flexibility of selecting the LM they want to use in the UI. The chosen LM is fed a prompt, a query, and enhanced context for each checklist questions. For questions asking whether the users used AI assistants, we mandate that users answer themselves.

Magnusson et al. (2023) previously reported that there are some common self-reported checklist answers issues such as submitting identical answers to each question and falsely reporting code availability. In result, we only evaluate papers in our dataset without these known issues.

Comparing responses from RAG systems and LLMs was used in determining which assistant to use. We quantitatively evaluate how different LLMs and RAG systems do on the task relative to human responses for checklist answers from ACL (§A.2), NeurIPS (§A.3), and NeurIPS Datasets & Benchmarks (§A.4) papers.

Similar to prior work on question answering over scientific articles (Baumgärtner et al., 2025a), we find that providing models with top-ranked passages from a retriever leads to better performance than using full-text inputs.

5 Conclusion

This paper introduces ConfReady, a LM-based system which can be used to empower authors reflect on their work and act as a 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.

Limitations

Structured Output Format A major issue with incorporating LMs into applications is their failure to follow output format inconsistency (faithfulness hallucination). We mitigate this issue and enhance the LM integration by giving LMs the format-restricting instruction to output in JSON format like the JSON mode in the open source OpenAI API and closed source Gemini API (Gemini Team et al., 2024). Additionally, some libraries like the structured outputs library Instructor[11] require JSON.

Other Conference Checklists The current version of the ConfReady application is for conferences under the Association for Computational Linguistics (e.g., ACL, COLING, CoNLL, EMNLP, and NAACL) and NeurIPS. However, the application can be modified for other conferences and applications.

Multi-answer Some authors provide a list of sections even when questions are only asking for a single section. The application currently doesn't mimic this human behavior.

Ethics Statement

Hallucination in LMs LMs are known to hallucinate and generate false or misleading information. For our application, it means the model can output incorrect sections. Users of our prototype application must only use it as a assistant or as a way to reflect on their work, not as a tool for automation.

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.

Tim Baumgärtner, Ted Briscoe, and Iryna Gurevych. 2025a. Peerqa: A scientific question answering dataset from peer reviews.

Tim Baumgärtner, Ted Briscoe, and Iryna Gurevych. 2025b. Peerqa: A scientific question answering dataset from peer reviews. *arXiv preprint arXiv:2502.13668*.

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.

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

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.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey.

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.

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*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2024. Large language models are zero-shot reasoners. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

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.

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.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. 2024. Raptor: Recursive abstractive processing for tree-organized retrieval. In *International Conference on Learning Representations (ICLR)*.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 31210–31227. PMLR.

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.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Zhiruo Wang, Jun Araki, Zhengbao Jiang, Md Rizwan Parvez, and Graham Neubig. 2023. Learning to filter context for retrieval-augmented generation.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2024. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

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 Additional Checklist Dataset Information

To enable quantitative evaluation of the ConfReady assistant, we constructed a structured dataset of checklist responses across three major conferences: ACL 2023 (long papers only), NeurIPS 2023 Main Conference, and NeurIPS 2023 Datasets & Benchmarks (D&B). Each instance in the dataset consists of question-answer (QA) pairs extracted from submitted checklists, with accompanying metadata including justification fields, section references, and arXiv LaTeX source. We prioritized papers that included a valid arXiv version to allow source-level inference.

A.1 Data Collection Pipeline

Due to the volume of papers involved, the dataset was assembled through a partially automated pipeline composed of several stages:

Title and Link Scraping We scraped accepted paper titles for each venue using 'requests' and 'BeautifulSoup'. For ACL, we used the ACL Anthology directly. For NeurIPS, papers were extracted from OpenReview using a Selenium-based crawler. For each paper, we then queried Google to obtain the top five search results and identified the arXiv link if present.

Checklist Extraction For ACL, we observed that the checklist was consistently found in the last two pages of each PDF. Using PyPDF2, we extracted only those pages for faster processing. In contrast, NeurIPS checklists could be found anywhere in the document. As a result, full paper OCR and regex-based parsing were required to locate and segment checklist content.

Parsing Strategy Each checklist was tokenized using conference-specific regex heuristics:

- **ACL:** Questions identified by the presence of question labels (e.g., A1, A2, ...), followed by responses marked with checkmarks or yes/no indicators, plus optional justifications.
- **NeurIPS Main:** Structured blocks starting with "Answer:" and often followed by a "Justification:" were parsed into fields.
- **NeurIPS D&B:** These used alphanumeric identifiers (e.g., 1a, 3b), and responses were directly embedded with [Yes]/[No] indicators.

Storage and Structure Each paper's checklist was stored in a structured JSON or Excel format with fields for paper ID, question ID, raw text, answer, justification, and section name. These entries were cross-referenced with the arXiv tarball to ensure the TeX source aligned with the conference submission.

In order to comparing responses from RAG systems can be helpful in determining which assistant to use. We quantitatively evaluate how different LLMs and RAG systems do on the task relative to human responses for checklist answers from ACL (§A.2), NeurIPS (§A.3), and NeurIPS Datasets & Benchmarks (§A.4) papers.

A.2 ACL Evaluation

ACL Checklist Dataset 1975 * 18 (or whatever the number is) checklist questions and answers from 1975 ACL long conference papers. We focused on long conference papers because we wanted to test the application's ability to deal with long-context failures and the limited context length of models like GPT-4o. Similar to other works, we encountered low effort and incomplete checklists (Magnusson et al., 2023; Goldberg et al., 2024). After removing 13 papers because they had unfilled checklists (13.8% of all papers), the evaluation dataset consisted 1539 questions and answers.

The only selection criteria was that each selected paper had to have an arXiv version so that the LM responses could be generated from the TeX Source (TeX file or zipped folder). The human checklist responses were obtained from the conference version of the paper.

Of the 1539 questions, 45 questions (2.92%) were checked with a "Yes" or "No" but lacked a justification or section name in the response. Additionally, 477 questions (30.99%) were left unchecked,

	NeurIPS	
Metric	Main	D & B
All Collected Papers		
Total Papers	387	458
Questions per Checklist	15	18
No Checklist	0	57
Wrong Checklist	0	42
Blank Checklist	1	39
All Yes Responses	1	6
No Section Names	N/A	N/A
AI Use in Writing	N/A	N/A
Not on arXiv	26	49
Sampled Evaluation Subset		
Total Papers	N/A	N/A
Avg Tokens (Paper)	N/A	N/A

Table 3: Metadata and statistics for the NeurIPS portion of the checklist dataset. Includes main conference and Datasets & Benchmarks (D&B) track papers. Some fields are marked as not available (N/A) due to missing or inconsistent reporting.

either because they were marked "not applicable", or there was no response altogether.

ACL Edge Cases

ACL Checklist Performance This a section where we can include a table on how the RAG models do on

A.3 NeurIPS Evaluation

NeurIPS Checklist Dataset We compiled 135 checklist questions and answers from 9 Neurips Main Conference Papers. Similar to the ACL Checklist dataset, our sole selection criterion was that each paper must have an arXiv version to generate LM responses from its TeX source. Human checklist responses were obtained from the conference version of the paper. Out of the 135 questions, 10 (0.07%) were answered "Yes", "No" or marked "not applicable" without any justification.

NeurIPS Checklist Performance

A.4 NeurIPS Datasets & Benchmarks Evaluation

NeurIPS Datasets & Benchmarks Checklist Dataset We compiled 162 checklist questions and answers from 9 Neurips Dataset and Benchmarks papers. Similar to our ACL and NeurIPS Main Checklist, the papers were selected based on the availability of an arXiv version, and human checklist responses were obtained from the conference version of the paper. Our of the 162 questions, 73 (45%) were answered without justification.

NeurIPS Datasets & Benchmarks Checklist Performance

B Why a Recursive Retrieval RAG System

Below we provide additional rationale for why we picked a recursive retrieval RAG system over other RAG systems and LMs.

- *Goal of ConfReady*: ConfReady is designed to empower authors to reflect on their work and assist authors with conference checklists. Consequently, the application needs to not only give high quality answers, but also needs to give them quick enough for users to find the application valuable.
- *Why use RAG instead of a LM assistant*: RAG is known to reduce hallucinations (Lewis et al., 2020) and mitigate the issue of LMs performing worse with long context. LMs were evaluated because it is challenging to determine

Introduction: Behave like you are the author of a paper you are going to submit to a conference.

Question: Did you describe the limitations of your work?

Additional Context: Point out any strong assumptions and how robust your results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only held locally). Reflect on how these assumptions might be violated in practice and what the implications would be. Reflect on the scope of your claims, e.g., if you only tested your approach on a few datasets, languages, or did a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated. Reflect on the factors that influence the performance of your approach. For example, a speech-to-text system might not be able to be reliably used to provide closed captions for online lectures because it fails to handle technical jargon. If you analyze model biases: state the definition of bias you are using. State the motivation and definition explicitly.

Output Structure: If the the answer is 'YES', provide the section name. The only valid section names are {section names}. If the answer is 'NO' or 'NOT APPLICABLE', the section name is 'None'. Provide a step by step justification for the answer. Format your response as a JSON object with 'answer', 'section name', and 'justification' as the keys. If the information isn't present, use 'unknown' as the value.

Figure 5: Example prompt for question A1. *Purple:* The introduction instructs the LM to assume the role of an author. *Blue:* The question is the primary query that the LM needs to address. *Brown:* The additional context attempts to cover all relevant aspects related to the question. *Black:* The output structure makes it easier to transfer the LM response to the frontend. *Green:* The section names that the LM should consider are taken from parsed TeX files.

whether RAG systems are recalling information from training or utilizing given context. information or truly utilizing external tools Additionally most models have maximum context length [12] and max input [13] that prevent long papers from directly being input into LMs.

C Prompts

Figure 5 shows the prompt for question A1 of the ARR Responsible NLP Research checklist.

[12] https://platform.openai.com/docs/models

[13] https://platform.openai.com/docs/guides/embeddings

D RAG Chunk Filtering

Giving LMs all retrieved content can increase computational cost and increase the risk of hallucination (Shi et al., 2023). In the case of RAG applications, this is due to retrieved chunks containing both relevant and irrelevant context (Gao et al., 2024). For the papers in our evaluation dataset, there are 19 questions in the ARR Responsible NLP Research checklist. A few of these questions can be more easily answered by using metadata to filter out irrelevant chunks. Consequently, ConfReady labels every chunk with meta information such as section name, whether it has citations in it, and whether each chunk has license information (e.g., CC-BY 4.0).

D.1 Filters for ACL 2023 Responsible NLP Checklist

Other RAG systems has demonstrated the ability to filter out part of retrieved chunks (Wang et al., 2023). Due to some questions in the checklist being about specific required sections, we filter for section names in question A1 and A3.

A1. Did you describe the limitations of your work?

A3. Do the abstract and introduction summarize the paper's main claims? For question B1, the application only keep chunks with the latex commands for cite.

B1. Did you cite the creators of artifacts you used?

E Dataset Details

To support evaluation of checklist response quality and benchmarking of the ConfReady RAG system, we constructed a structured dataset composed of checklist question-answer pairs from major ML and NLP conference submissions. We focused on ACL 2023, NeurIPS 2023 Main Conference, and NeurIPS 2023 Datasets & Benchmarks (D&B), prioritizing papers that listed an arXiv LaTeX version to facilitate downstream LLM analysis.

E.1 Collection Process

Due to the scale of the dataset that ought to be constructed, most of the collection was automated using Python scripts. The multi-stage automated collection pipeline includes:

Scraping Titles of All Papers: Accepted paper titles for ACL were scraped from the respective conference websites using a combination of the python packages ‘requests‘ and ‘BeautifulSoup‘, while NeurIPS papers which were hosted on OpenReview were extracted from OpenReview pages using a Selenium-based crawler. For each paper, we then queried Google Search programmatically to obtain the first five relevant links. Assuming the arXiv version exists and is situated within the first five links of the search, we obtain the first relevant link with "arXiv.org" and store it in an excel file with multiple pages indicating multiple conferences.

PDF Retrieval and Preprocessing: Upon storing the links of the source of each papers, the next step is to extract the checklist answers for each conference and store it. For ACL, ‘.pdf‘ links were normalized from ACL Anthology URLs, while NeurIPS papers linked directly to OpenReview-hosted PDFs. To extract checklist answers, the last two pages of each paper were parsed using PyPDF2 for ACL conferences. This proved to be effective since the checklist is always situated in the last two pages of a paper. This also helped increase efficiency as only the last two pages of each file were downloaded strategically to optimise processing times. The checklist contents was then extracted and using simple regex both the question and its respective answer was easily identified. OCR was used to extract the visual "tick" information on each question. However, for Neurips papers, this process is not very straightforward since the placement of the checklist can be random. The entire contents of the paper was first extracted and then using regex the contents of the checklist was identified using a similar pipeline for ACL.

Checklist Extraction: Each paper’s checklist was processed using regex-based parsing tailored to conference-specific checklist templates:

- **ACL:** The ACL checklist is embedded on the last two pages. Answers were marked using symbols followed by optional justifications. These markings were non-standard, often inconsistently aligned or visually formatted (e.g., ‘A1. ...‘). Matching between symbol and question ID was handled via proximity-based heuristics.
- **NeurIPS Main:** The checklist consisted of numbered sections (1–15), each beginning with an ‘Answer: [Yes/No]‘ and optionally followed by a justification and guidelines. These were parsed using block-level regex patterns that captured structured ‘Answer:‘ and ‘Justification:‘ segments.
- **NeurIPS D&B:** This checklist was alphabetically segmented (e.g., ‘1a‘, ‘3b‘) and directly annotated with ‘[Yes]‘ or ‘[No]‘ within the text. Checklist sections were sometimes interrupted by additional formatting, requiring both tolerance for whitespace and lenient matching across hyphenated line breaks. Parsing used a custom regex system that preserved explanation blocks and removed checklist headers.

Storage: Each paper was saved with a structured mapping of its PDF filename to the extracted arXiv ‘.tar.gz‘ (source) file, allowing clean post-hoc linkage between PDF and TeX artifacts. Answers were stored in per-paper Excel sheets to enable incremental dataset growth.

E.2 Dataset Composition

The final dataset consists of (Since not ready yet, I’m not able to post final stats):

- Over 1975 papers scraped across ACL 2023, NeurIPS 2023 Main, and NeurIPS 2023 D&B.
- For each paper: title, conference-specific checklist answers (20–25 fields), arXiv link, and PDF source.
- 80–85% of papers include both complete checklist responses and valid arXiv TeX sources.
- Questions were mapped uniformly into column headers such as `acl_question_a1` or `neurips_question_3c` to enable downstream processing.

E.3 Selection Criteria

We included only papers that met all of the following conditions:

1. Have an arXiv submission linked through Google search results.
2. Successfully downloadable PDF from ACL Anthology or OpenReview.

3. The conference that the paper was submitted to matches the version of the LaTex version of the paper downloaded from arXiv.

We excluded papers without checklists, malformed PDFs, or cases where the checklist structure deviated from known templates.

E.4 Tokens

We report distributions of the number of tokens for each paper category (e.g., ACL Main Long, ACL Findings Short). We collect a representative sample of papers and use the AutoTokenizer module from the Hugging Face Transformers library to compute token counts using the Llama-3 tokenizer. This provides a standardized metric for comparing the textual length and complexity across different paper types. Our tokenization strategy aligns with the methodology used in PeerQA (Baumgärtner et al., 2025b), which also employs the Llama-3 tokenizer to ensure consistency in token-based measurements across scientific domains.

To ensure fair comparison, we preprocess the raw LaTeX source of each paper to strip out non-content elements such as comments, pre-abstract material, and non-body sections like acknowledgments or references. Only the content from the abstract to the conclusion is retained before tokenization. This filtering allows us to isolate substantive paper content and eliminate artifacts of formatting or boilerplate text.

Token counts are particularly relevant in the context of our ConfReady system, which uses retrieval-augmented generation (RAG) with large language models (LLMs). Understanding the average number of tokens is essential for system design decisions such as chunking strategy, prompt window sizing, and embedding throughput. Longer token sequences may exceed the context window limits of certain LMs, necessitating recursive summarization or chunk filtering. Thus, this token-based analysis not only quantifies document complexity but also informs key components of our RAG pipeline. Figure 4 illustrates the variation in average token counts across the different categories of papers considered.

E.5 Dataset Limitations

While the dataset is extensive, several limitations are noteworthy:

- **Checklist variability:** Minor formatting inconsistencies (e.g., broken lines, spacing issues) sometimes caused regex patterns to fail and those checklists had to go through another round of manual checking.

All preprocessing was done via automated scripts with manual oversight for integrity checks and sample-level validation.

F Features of the User-Interface

The features for the user interface and the rationale behind them are listed below:

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 (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[14] and GitHub[15] 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).

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.

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

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

6. *Download:* Only after users have reviewed each section are 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[16]. To use the application, users upload the TeX source (single TeX or zipped folder). 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.

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