MetaWriter: Facilitates Meta-review Authoring with Hybrid Text Generation

Lu Sun
UC San Diego,
San Diego, USA
l5sun@ucsd.edu

Stone Tao
UC San Diego,
San Diego, USA
stao@ucsd.edu

Junjie Hu
University of Wisconsin-Madison, Madison, USA
junjie.hu@wisc.edu

Steven P.Dow
UC San Diego,
San Diego, USA
spdow@ucsd.edu

Abstract

We present MetaWriter, an interactive system that facilitates meta-review writing and fact-checking using automatic tagging and text generation techniques. To support the meta-review process for peer-reviewed conferences, the system offers useful functionalities including decision prediction, review aspect highlights, viewpoint extraction, and draft generations. Our system adopts a hybrid summarization model of both abstractive and extractive summarization modules. We present the experiment performance of our fine-tuned hybrid summarization model on meta-review generation through automatic evaluation as well as a fine-grained user study. We show that MetaWriter is effective in shortening the meta-review writing time and improving the meta-review quality written by users. All model checkpoints and the system will be released upon acceptance.[1]

1 Introduction

Peer review is the cornerstone of scientific research. With a skyrocketing number of submissions, the workload also significantly increases for reviewers and meta-reviewers to carefully assess and justify their decisions. Meta-reviewers or senior editors play an important role in this process. Typically, meta-reviewers synthesize multi-aspects information from authors and different reviewers and then provide a reasonable recommendation for the paper (Shah, 2022) within a short period of time. To reduce the burden of meta-reviewers, we proposed **MetaWriter**, a system that facilitates meta-review authoring, fact-checking and decision-making through aspect tagging and text generation.

MetaWriter consists of three main functions: (1) **decision recommendation** that integrates review ratings and all review comments to predict the decision of accepting or rejecting a paper; (2) **aspect visualization** that utilizes a pre-trained tagger to highlight crucial aspects in each review for decision making in the meta-review process; and (3) **hybrid meta-review generation** that uses a hybrid meta-generation model to first extract important sentences from each review and then generate a draft for meta-reviewers.

Most recently, there have been a few research attempts that explore the possibility of generating meta-reviews automatically (Bhatia et al., 2020; Kumar et al., 2021; Shen et al., 2022b). Notably, Shen et al. (2022b) proposed a controllable meta-review generation method to generate meta-reviews according to the categories of reviews. However, in order to use the generated draft, meta-reviewers still have to briefly understand where the generated draft comes from and proofread the factuality of the generated draft. Hence, instead of a one-step generation, we use a *hybrid model* to generate a meta-review and visualize the extracted sentences from each review. Additionally, different from prior studies that mainly focus on one-step meta-review generation, the MetaWriter system provides a suite of functions (e.g., decision prediction, aspect visualization, meta-review generation) to facilitate not only the writing but also the *fact-checking and decision-making* in the entire meta-review process.

To systematically evaluate the effectiveness of MetaWriter, we compare our hybrid meta-review system with multiple state-of-the-art meta-review summarization methods. Furthermore, we conduct a within-subject experiment where we invite 32 researchers who have review or meta-review experiences to write meta-reviews using MetaWriter versus a plain text editor. The experiment demonstrates that MetaWriter helps to shorten the meta-reviewing time and improves the meta-review quality post-edited by humans. Our contributions are two folds: (1) we develop a hybrid meta-review system with helpful support functions, and (2) we conduct a series of automatic and human evaluations with strong summarization methods.

[1]Video Demo at: https://vimeo.com/852308849?share=copy. Code at: https://github.com/LusunHCI/metaGenerator.git

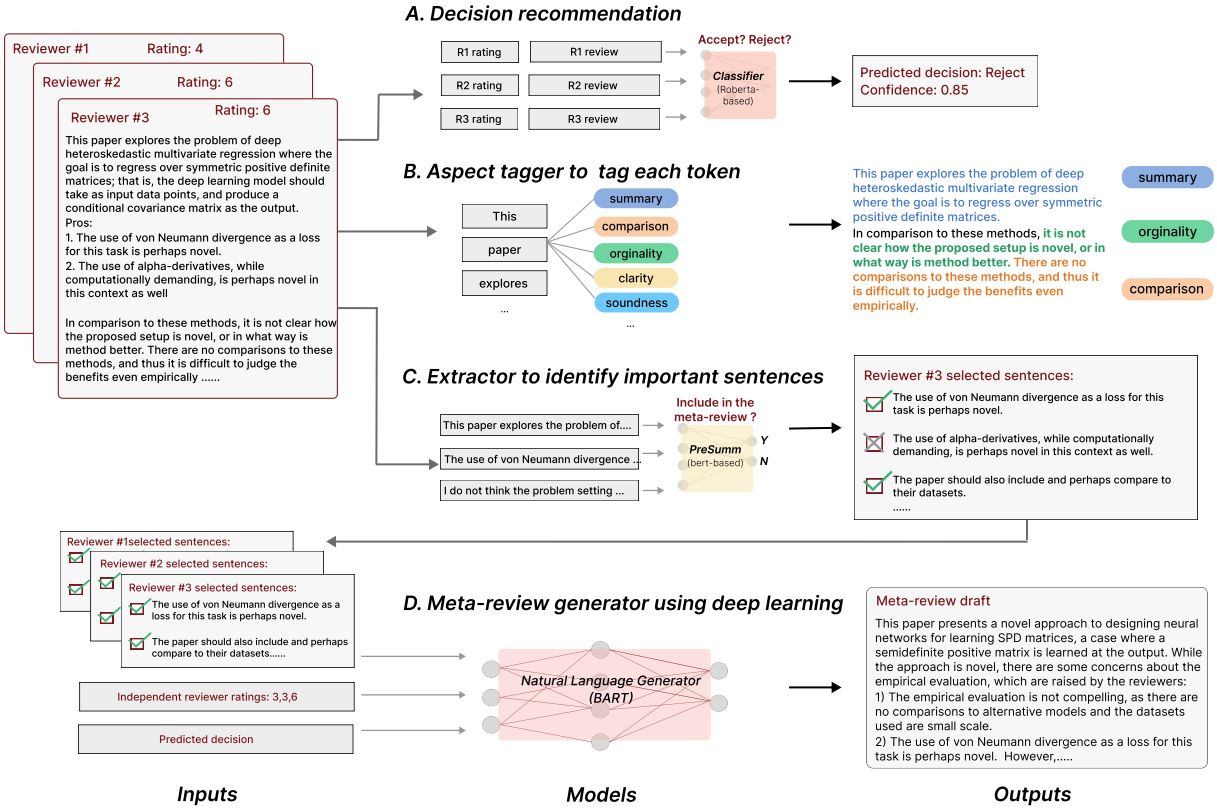

Figure 1: Platform that can predict decisions, run trained tagger, extractive and then abstractive generation model to facilitate the meta-review process.

2 System Description

2.1 System Overview

MetaWriter is an automated end-to-end system shown in Figure 3. Meta-reviewers only need to provide the paper link on OpenReview, and an automatic scrapper can query the paper information. All scrapped information is then stored on the server which runs our models concurrently. Our system sequentially applies the decision prediction model to predict the potential decision of the paper(Figure 3-A), uses the aspect tagger to highlight each review (Figure 3-B), then extracts key sentences from each review by our fine-tuned extractive summarization model (Figure 3-C), and finally takes the extracted key sentences, all reviewers' ratings and the paper's predicted decision as inputs to generate a meta-reviews draft (Figure 3-D). After these prediction steps, all results are uploaded to a database and then rendered to the front-end interface. The front-end interface provides three add-on interactions for meta-reviewers: they can toggle up and down to check each highlighted review aspect; they can hover over the extracted sentences to locate where these sentences are mentioned; and they can select to show and hide the generated draft. Here are three main functions MetaWriter contains:

F1: Decision recommendation MetaWriter provides a decision recommendation with a confidence score to help meta-reviewers make decisions. We use each reviewer's ratings and review content to train a decision prediction model. This function is similar to the existing study focusing on paper decision prediction (Kumar et al., 2022). However, unlike their goal to automate the meta-review decision-making process, our system provides this decision recommendation with a confidence score to facilitate human reviewers in decision-making.

F2: Visualizing tagged aspects We apply a pretrained tagger (Yuan et al., 2021) to color-code each word according to its aspect. In our dataset, 30.4% of words are tagged with an aspect, including summary, substance originality, clarity, soundness, etc, as shown in Figure 1-A. MetaWriter color-codes these tags on each independent review to support proofreading. Yuan et al. (2021) released this aspect tagger to explore the possibility of paper review generation. Different from their use case, we run this aspect tagger to help meta-reviewers effi-

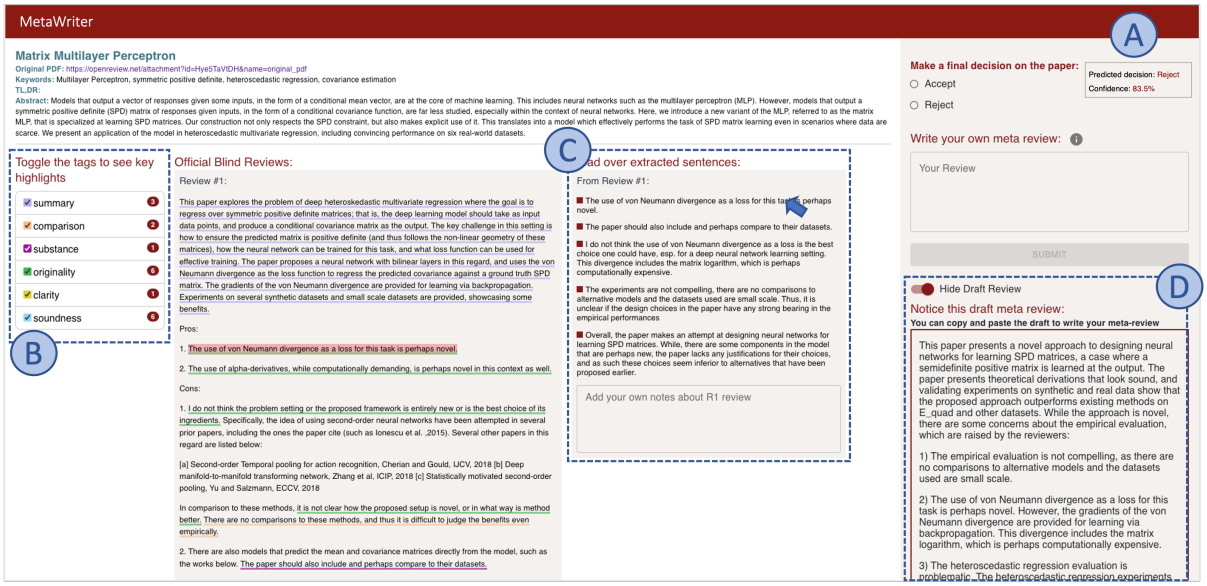

Figure 2: MetaWriter interface. (A) users can see a predicted decision based on reviewers' review content and ratings. (B) users can toggle the tags to highlight specific aspects. (C) users can hover over the extracted sentences to locate the corresponding positions highlighted in each review. (D) users can see or hide the generated draft using the toggle button. Other basic function includes that: Users can review the three original independent reviews and users can make a final decision and write their meta-review on the right.

ciently check the reviews.

F3: Hybrid model for meta-review generation To help meta-reviewers fact-check the generated meta-review draft, we use a hybrid model that first extracts the candidate sentences from each review and then uses the extracted sentences, together with their ratings, predicted decisions, and the paper abstract to generate a meta-review. Unlike the previous studies that directly combined all reviews to fine-tune a generative model (Shen et al., 2022b; Kumar et al., 2021), this hybrid model decomposes the writing procedure into an extraction-then-generation pipeline, which enhances the system transparency and fact-checking since meta-reviewers can easily identify which sentences are used to generate a draft.

2.2 Dataset

We collect a large peer review dataset from the online peer-reviewing platform OpenReview [1] for ICLR, one of the largest machine learning conferences. We collect each submission's data using the OpenReview API and scrap reviews from all publicly accessible paper submissions from the year 2018 to 2022. Earlier years submissions were not collected since their meta-reviews were not released. For each submission, we collect the paper information, all official reviews with reviewer ratings and confidence scores, final meta-review with ratings and the final decision. After filtering out submissions that had fewer than 3 reviews and meta-reviews that had less than 20 words, we retain 9,803 submissions along with their meta-reviews and 34,219 independent reviews.

2.3 Hybrid Summarization

Extractive Summarization: Extractive summarization techniques have shown evidence of selecting strong candidate sentences for a summary (Liu and Lapata, 2019). Here, we fine-tune a pre-trained extractive summarization model to select key sentences from each review to shorten participants' time on reading the long reviews as shown in Figure 1-B. We first create an Oracle extractive summarization dataset from our ICLR dataset to train the extractive summarization model. The inputs are individual reviews and the labels are generated via a beam search procedure on the individual review following (Xu and Durrett, 2019; Liu and Lapata, 2019). The goal of this procedure is to label the sentences which got incorporated into the final meta-review. In particular, during beam search for each additional sentence we propose to add to the label, we compute a heuristic cost equal to the ROUGE score of a given sentence with respect to the reference summary written by ICLR meta-reviewers (Lin, 2004). Specifically, we use ROUGE-L as a measurement in this process

[1] https://openreview.net/

as it considers sentence-level structures and finds similarities amongst sentences and n-grams which is ideal for complex, long-form content such as reviews (Lin and Och, 2004; Lewis et al., 2019). Here, we iteratively loop over sentences from each review and only keep sentences when the ROUGE$_L$ score between the selected sentences and the meta-review improves.

Abstractive Summarization: Abstractive summarization techniques can generate a short and concise summary that captures the salient ideas of the source text (Lewis et al., 2019). Similarly, meta-reviews typically synthesize all reviews into one cohesive, natural summary without directly copying sentences from each review. In Figure 1-C, we use an abstractive summarization method that combines similar sentences across independent reviewers together to generate new sentences (Lewis et al., 2019; Shen et al., 2022b). We combine the extracted sentences from all three reviewers using the extractive summarization above, along with their ratings and the predicted decision as inputs, and then use the real meta-review as the output target. We use our dataset to fine-tune a BART model for meta-review generation (Lewis et al., 2019).

2.4 Client-Server Implementation

The client-side UI was developed using React and Typescript. We use a Firebase server to set up user accounts and store data. Given the limits of accessing the server, we run our trained language models to generate tags and drafts on the GPU server offline and then transmit the results to the Firebase server. The client-side program is responsible for rendering the user interface and monitoring user actions on the webpage. We style the components and theme to be similar to the OpenReview platform. The Firebase server stores all reviews and users' interaction data with the system.

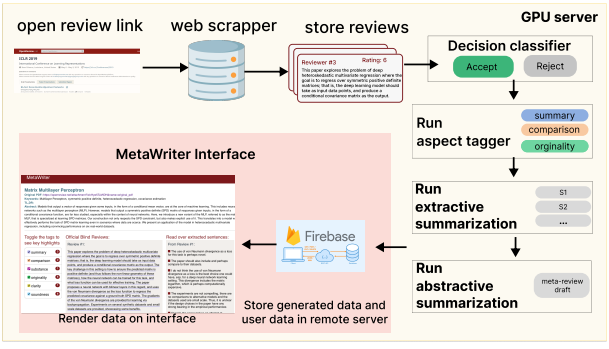

Figure 3: Backend and frontend system architecture

2.5 UI Design

As shown in Figure 2, MetaWriter simulates OpenReview which provides the paper abstract, a link to the paper pdf, keywords, and three original independent reviews. Figure 2-A shows the recommended decision and the confidence score. Figure 2-B shows the output of the aspect tagger as a set of six categories and their frequency of appearing in the reviews. When users toggle the checkbox before the category, the corresponding categories will get underlined with the colors. Figure 2-C lists all sentences extracted by MetaWriter for each review. When participants hover over an extracted sentence, the corresponding sentence will be highlighted in red to help participants locate it in the original review. In Figure 2-C, when the user hovers over the sentence "The user of vonNeumann divergence as a loss for this task is perhaps novel", MetaWriter will highlight the corresponding sentence in red in the middle area. This allows meta-reviewers **cross-check** the location of the original sentences. Figure 2-D provides the generated meta-review draft by the abstractive summarization model based on extracted sentences. Users can select to hide the meta-review if they think the draft may influence their judgment. Instead of predicting the paper's decision, our system only provides the recommendation with a confidence score. The meta-reviewers still have to make their own choice in the upper right corner. To avoid the strong biases that may be contained in the meta-review draft, MetaWriter deletes the sentences that indicate the decision in the meta-review draft. Participants will then post-edit the final meta-review.

3 Analysis and Evaluation

3.1 Main Results

Methods in Comparison We employ two classic extractive summarization baselines and an abstractive baseline: (1) **LexRank** computes a graph-based centrality score to select important sentences (Erkan and Radev, 2004); (2) **TF-Extract** is a Transformer-based extractive model that follows PreSumm (Liu and Lapata, 2019) to use a fine-tuned BERT model for extractive summarization; (3) **TF-Abstract** is a Transformer-based abstractive model fine-tuned from a bart-cnn-large model (Lewis et al., 2019) using all reviewer's original reviews without extraction. Both extractive methods extract the four most important sentences from each review and concatenate all reviewers'

Model	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_L$	AC	DC	Fluency	chrF	METEOR	BLEU	BERTScore-p	BERTScore-r	BERTScore-f
LexRank	0.225	0.049	0.123	0.327	0.717	36.25	17.3	0.252	2.5	0.806	0.846	0.825
TF-Extract	0.341	0.085	0.162	0.367	0.815	43.49	30.5	**0.289**	3.7	0.824	0.847	0.835
TF-Abstract	0.335	0.091	0.203	0.480	0.845	15.45	34.4	0.235	5.3	0.851	0.849	0.850
MetaWriter	**0.345**	**0.095**	**0.207**	0.538	0.845	**5.12**	**35.2**	0.238	5.4	0.851	0.847	0.849
MetaWriter (w/o ratings)	0.325	0.087	0.199	0.420	0.845	20.23	34.0	0.224	4.7	0.848	0.846	0.847
MetaWriter (w/ discussions)	0.345	0.092	0.203	0.535	0.845	5.32	35.3	0.243	5.5	0.848	0.848	0.848

Table 1: Meta-review generation results on the Rouge_1, Rouge_2, and Rouge_L F_1 scores, aspect coverage (AC), decision consistency (DC), fluency(perplexity), as well as chrF, METEOR, BLEU, and BertScores.

extracted sentences as a meta-review. In comparison, our two-step hybrid model (**MetaWriter**) first uses a fine-tuned BERT model to perform extractive summarization on each review, then uses a fine-tuned BART model to generate a meta-review draft using all extracted sentences, the predicted decision and reviewers' ratings. During training, we fine-tune the BERT model in the same way as PreSumm (Liu and Lapata, 2019) and then fine-tune the BART model to generate the final meta-review draft (Lewis et al., 2019). To examine the impact of ratings and discussion contents during the rebuttal, we conduct an ablation study to fine-tune two additional hybrid models: (1) one without ratings in the input; and (2) another one with additional reviewers' replies during discussion.

Automatic Evaluation of Generation We use the standard evaluation metrics that show high correlations with human performances on text summarization (Fabbri et al., 2021), i.e., ROUGE scores (Lin, 2004), chrF(Popović, 2015; Post, 2018), METEOR (Banerjee and Lavie, 2005) and BertScore (Zhang et al., 2019). We report these measures in Table 1. MetaWriter generates comparable or slightly better meta-reviews than the baselines in terms of ROUGE and BertScores. Interestingly, TF-Extract has a higher METEOR score, suggesting that it provides more unigram overlaps with the meta-review. In the ablation study, we find that removing ratings from the input leads to suboptimal performance. MetaWriter with discussion content performs similarly to MetaWriter.

Additionally, we also evaluate the usefulness of generated meta-reviews by three heuristic criteria: *coverage*, *consistency with the decision* and *fluency* (Yuan et al., 2021; Zhang and Song, 2022). Specifically, we measure the coverage by the percentage of aspect tags in the ground-truth meta-review covered by the generated meta-review. We use the aspect tagger trained by Yuan et al. (2021) to predict aspects in the generated meta-reviews and ground-truth meta-reviews. Note that we use the reviews with detected tags as a rough proxy as the aspect tagger only detects tags for 35% of the reviews. To measure the consistency with the decision, we use the fine-tuned Longformer to predict the decision recommendation based on the generated meta-review and the original reviewer's ratings, and measure the prediction accuracy. Finally, we measure the fluency of the generated meta-review by the perplexity of text using a pretrained GPT-2 with a maximum sequence length of 1024 tokens. Results in Table 1 show the superior performance of MetaWriter over the other baselines. Our MetaWriter generates more fluent outputs (i.e., lower perplexity) and covers more aspects than the others. However, decision consistency does not show an obvious discrepancy.

Decision Recommendation We concatenate all reviewers' ratings together with their reviews as inputs and then fine-tune a Longformer model with logistic regression to classify the paper decision (Beltagy et al., 2020). To study the effect of inputs for decision prediction, we fine-tune two more decision prediction models, having inputs of only reviewers' scores or reviewers' reviews. We fine-tune each model for 15 epochs to get the best performance. The highest accuracy of the classification model is 89.6%. As shown in Table 2, ratings play the dominant role in the paper acceptance prediction.

Input	Accuracy	Precision	Recall	F1
score only	0.895	0.893	0.893	0.893
reviews only	0.674	0.657	0.674	0.674
scores and reviews	0.896	0.895	0.895	0.895

Table 2: Input Ablation of Decision Recommendation

3.2 Case Study: Human Evaluation

Different from prior works that perform a *comparative* user study to select the best system among multiple system outputs according to certain quality aspects, we perform an *interactive* user study to evaluate the efficiency of writing a meta-review

with and without the aid of MetaWriter in terms of time and quality. Specifically, we conduct a within-subjects experiment with 32 participants who have ML conference review experience. We simulate a realistic meta-reviewing process and invite participants to play the role of a meta-reviewer to summarize independent reviews and write meta-reviews. Each participant completes two meta-reviews: one with MetaWriter (MetaWriter condition) and another with only a plain text editor (Baseline condition). We counterbalance the order of each condition through random assignments. Our study selects two borderline rejected papers that (1) have their reviews and meta-reviews of average word length and (2) have high ROUGE$_L$ scoring meta-reviews generated by MetaWriter.

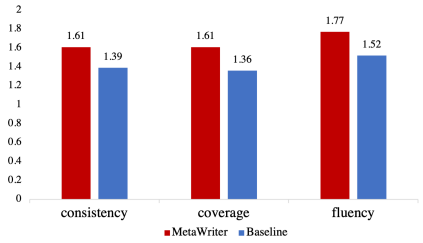

Figure 4: Expert ratings on the consistency, coverage, and fluency of written meta-reviews using the MetaWriter versus Baseline.

Efficiency Improvement We use the analysis of covariance (ANCOVA) test to examine the effect of the two conditions on writing time and control the length of the meta-review as a co-variate. Participants spent significantly less time writing meta-reviews when using MetaWriter than the baseline.

Longer Meta-reviews with Higher Quality We find that participants write statistically significantly longer meta-reviews with MetaWriter. To measure the meta-review quality, we recruit two experts who have experience being an ICLR meta-reviewer to rate the quality of all final meta-reviews (N=64) with a 3-dimension rubric: coverage, consistency, and fluency, using a three-point scale (0-2). Figure 4 shows that MetaWriter helps reviewers write more decision-consistent, fluent meta-reviews with higher coverage.

4 Related Work

Existing studies have explored various aspects to facilitate peer review process, such as paper decision prediction (Kang et al., 2018; Kumar et al., 2022), mining review aspects (Hua et al., 2019; Cheng et al., 2020; Wang et al., 2020; Yuan et al., 2021), or generate reviews or meta-reviews (Bartoli et al., 2016; Wu et al., 2022; Shen et al., 2022b; Zeng et al., 2023; Li et al., 2023; Shen et al., 2022a). Several works collected annotated datasets to understand peer review practice (Dycke et al., 2022; Bharti et al., 2023) or to support meta-review generation (Shen et al., 2022b; Bhatia et al., 2020). Recently, Shen et al. (2022b) proposed a controllable metareview generation method using categories of reviews. However, meta-reviewers still have to scan through each review and decide which aspects to control for. Instead, MetaWriter automatically predicts aspects and extracts a structured draft from each review, which enables faster proofreading. MetaWriter takes the extract-then-generate approach in meta-review generation to facilitate fact-checking on long reviews (Chen and Bansal, 2018; Gehrmann et al., 2018; Pilault et al., 2020; Dou et al., 2020). Also, different from Bhatia et al. (2020), we further incorporate ratings and discussion content as training input and use more recent pre-trained language models that gain better performance. MetaWriter is the first interactive system that conducts both automatic evaluation and human-controlled experiments in meta-review writing.

5 Discussion and Future Work

MetaWrite is an interactive system that facilitates the meta-review authoring process. Different from previous studies, we simulate a realistic meta-reviewing environment and we believe this is the first work that evaluates meta-review generation techniques in a real user study. As it stands, the system allows meta-reviewers to write high-quality meta-reviews more efficiently and provides interactive ways to fact-check summarizations. However, some users in our case study expressed concerns about over-reliance on such a system. We argue that MetaWriter is designed to **facilitate** the writing and decision-making process instead of **replacing** meta-reviewers in the peer-review. Besides, 12.5% users in the case study disagreed with the recommended decision and changed the tone of the meta-review draft while writing the meta-review. So, one future work is to encourage decision-aware meta-review generation. Furthermore, fact-checking function between sources and generated text is valued by participants given it provides the transparency of the black-box language models. Future work can potentially develop fact-checking approaches to make NLP systems more trustworthy.

Ethics and Broader Impacts

We argue that MetaWriter cannot replace human meta-reviewer. We position the system to help meta-reviewers better identify arguments from each meta-review in order to make a fair decision.

References

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Alberto Bartoli, Andrea De Lorenzo, Eric Medvet, and Fabiano Tarlao. 2016. Your paper has been accepted, rejected, or whatever: Automatic generation of scientific paper reviews. In *CD-ARES*.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Prabhat Kumar Bharti, Meith Navlakha, Mayank Agarwal, and Asif Ekbal. 2023. Politepeer: does peer review hurt? a dataset to gauge politeness intensity in the peer reviews. *Language Resources and Evaluation*, pages 1–23.

Chaitanya Bhatia, Tribikram Pradhan, and Sukomal Pal. 2020. Metagen: An academic meta-review generation system. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1653–1656.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. *ArXiv*, abs/1805.11080.

Liying Cheng, Lidong Bing, Qian Yu, Wei Lu, and Luo Si. 2020. Argument pair extraction from peer review and rebuttal via multi-task learning. In *Conference on Empirical Methods in Natural Language Processing*.

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2020. Gsum: A general framework for guided neural abstractive summarization. *ArXiv*, abs/2010.08014.

Nils Dycke, Ilia Kuznetsov, and Iryna Gurevych. 2022. Nlpeer: A unified resource for the computational study of peer review. *arXiv preprint arXiv:2211.06651*.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Sebastian Gehrmann, Yuntian Deng, and Alexander M. Rush. 2018. Bottom-up abstractive summarization. *ArXiv*, abs/1808.10792.

Xinyu Hua, Mitko Nikolov, Nikhil Badugu, and Lu Wang. 2019. Argument mining for understanding peer reviews. *ArXiv*, abs/1903.10104.

Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine Van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. 2018. A dataset of peer reviews (peerread): Collection, insights and nlp applications. *arXiv preprint arXiv:1804.09635*.

Asheesh Kumar, Tirthankar Ghosal, and Asif Ekbal. 2021. A deep neural architecture for decision-aware meta-review generation. In *2021 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 222–225. IEEE.

Sandeep Kumar, Hardik Arora, Tirthankar Ghosal, and Asif Ekbal. 2022. Deepaspeer: towards an aspect-level sentiment controllable framework for decision prediction from academic peer reviews. In *Proceedings of the 22nd ACM/IEEE Joint Conference on Digital Libraries*, pages 1–11.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Miao Li, Eduard Hovy, and Jey Han Lau. 2023. Towards summarizing multiple documents with hierarchical relationships. *arXiv preprint arXiv:2305.01498*.

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

Chin-Yew Lin and Franz Josef Och. 2004. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 605–612.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. *arXiv preprint arXiv:1908.08345*.

Jonathan Pilault, Raymond Li, Sandeep Subramanian, and Christopher Joseph Pal. 2020. On extractive and abstractive neural document summarization with transformer language models. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, page 9308–9319.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Nihar B Shah. 2022. An overview of challenges, experiments, and computational solutions in peer review (extended version). *Communications of the ACM*.

Chenhui Shen, Liying Cheng, Lidong Bing, Yang You, and Luo Si. 2022a. Sentbs: Sentence-level beam search for controllable summarization. In *Conference on Empirical Methods in Natural Language Processing*.

Chenhui Shen, Liying Cheng, Ran Zhou, Lidong Bing, Yang You, and Luo Si. 2022b. MReD: A meta-review dataset for structure-controllable text generation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2521–2535, Dublin, Ireland. Association for Computational Linguistics.

Qingyun Wang, Qi Zeng, Lifu Huang, Kevin Knight, Heng Ji, and Nazneen Fatema Rajani. 2020. Reviewrobot: Explainable paper review generation based on knowledge synthesis. *arXiv preprint arXiv:2010.06119*.

Po-Cheng Wu, An-Zi Yen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2022. Incorporating peer reviews and rebuttal counter-arguments for meta-review generation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2189–2198.

Jiacheng Xu and Greg Durrett. 2019. Neural extractive text summarization with syntactic compression. *arXiv preprint arXiv:1902.00863*.

Weizhe Yuan, Pengfei Liu, and Graham Neubig. 2021. Can we automate scientific reviewing? *arXiv preprint arXiv:2102.00176*.

Qi Zeng, Mankeerat Sidhu, Hou Pong Chan, Lu Wang, and Heng Ji. 2023. Meta-review generation with checklist-guided iterative introspection. *arXiv preprint arXiv:2305.14647*.

Hanqing Zhang and Dawei Song. 2022. Discup: Discriminator cooperative unlikelihood prompt-tuning for controllable text generation. *arXiv preprint arXiv:2210.09551*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Appendix

Length Analysis Figure 5 shows the distribution of the generated meta-review length and the original ground truth meta-review length for each model described in Section 3.1. The hybrid model and abstraction summarization model has better overlap with the ground truth distribution. However, all models tend to generate longer meta-reviews compared with the ground truth meta-reviews.

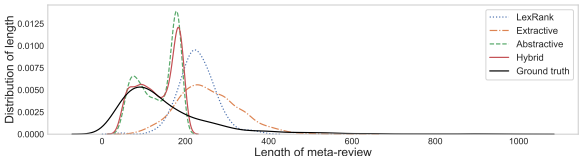

Figure 5: Distribution of generated meta-review length, compared with the ground truth length.