A publishing infrastructure for AI-assisted academic authoring

A DOI-citable version of this manuscript is available at https://doi.org/10.1101/2023.01.21.525030.

This manuscript (permalink) was automatically generated from greenelab/manubot-gpt-manuscript@d0aabcf on May 23, 2024.

Authors

- **Milton Pividori**
  0000-0002-3035-4403 · miltondp · miltondp · @miltondp@genomic.social
  Department of Biomedical Informatics, University of Colorado School of Medicine, Aurora, CO 80045, USA; Department of Genetics, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA 19104, USA · Funded by The National Human Genome Research Institute, K99/R00 HG011898; The Alfred P. Sloan Foundation, G-2023-20989; The Chan Zuckerberg Initiative; The Eunice Kennedy Shriver National Institute of Child Health and Human Development, R01 HD109765

- **Casey S. Greene** ✉
  0000-0001-8713-9213 · cgreene · GreeneScientist · @greenescientist@genomic.social
  Center for Health AI, University of Colorado School of Medicine, Aurora, CO 80045, USA; Department of Biomedical Informatics, University of Colorado School of Medicine, Aurora, CO 80045, USA · Funded by The Gordon and Betty Moore Foundation, GBMF4552; The National Human Genome Research Institute, R01 HG010067; The Eunice Kennedy Shriver National Institute of Child Health and Human Development, R01 HD109765

✉ — Correspondence possible via GitHub Issues or email to Casey S. Greene <casey.s.greene@cuanschutz.edu>.

Abstract

In this work, we investigate the use of advanced natural language processing models to streamline the time-consuming process of writing and revising scholarly manuscripts. For this purpose, we integrate large language models into the Manubot publishing ecosystem to suggest revisions for scholarly texts. Our AI-based revision workflow employs a prompt generator that incorporates manuscript metadata into templates, generating section-specific instructions for the language model. The model then generates revised versions of each paragraph for human authors to review. We evaluated this methodology through five case studies of existing manuscripts, including the revision of this manuscript. Our results indicate that these models, despite some limitations, can grasp complex academic concepts and enhance text quality. All changes to the manuscript are tracked using a version control system, ensuring transparency in distinguishing between human- and machine-generated text. Given the significant time researchers invest in crafting prose, incorporating large language models into the scholarly writing process can significantly improve the type of knowledge work performed by academics. Our approach also enables scholars to concentrate on critical aspects of their work, such as the novelty of their ideas, while automating tedious tasks like adhering to specific writing styles. Although the use of AI-assisted tools in scientific authoring is controversial, our approach, which focuses on revising human-written text and provides change-tracking transparency, can mitigate concerns regarding AI's role in scientific writing.

Introduction

Scholarly writing has evolved since the first scientific journals 350 years ago, adopting practices like external peer review in the last century [1,2]. It often involves dense language to convey new advances or literature summaries [3]. Meanwhile, recent computing advances have enabled large language models (LLMs) like OpenAI's GPT-3 and GPT-4 [4], revolutionizing technologies and applications in various fields, including medical informatics and scientific communication [5,6]. These models promise to streamline scientific writing [7], though their use raises accuracy and ethical concerns [8,9].

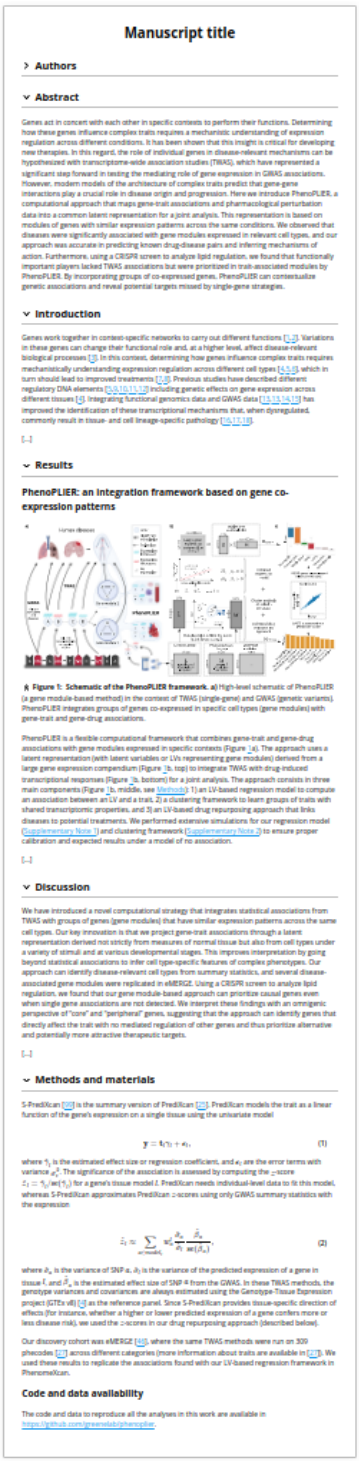
We introduce a human-centric AI method for scholarly writing, leveraging LLMs for draft revision within the Manubot platform, a tool for collaborative publishing [10]. Here, we propose the Manubot AI Editor, which suggests revisions via GitHub, balancing AI's efficiency with human oversight to ensure accuracy. Tested on five manuscripts, we found it maintained the original meaning, improved style, and handled complex expressions, proving a valuable addition to the Manubot suite. We anticipate our tool will help authors more effectively communicate their work.

Implementing AI-based revision into the Manubot publishing ecosystem

We propose a human-centric approach for the use of AI in manuscript writing, which consists of the following steps: 1) human authors write the manuscript content; 2) an LLM revises the manuscript, generating a set of suggested changes; 3) human authors review the suggested changes, and the approved edits are then integrated into the manuscript. By focusing on human review, this approach attempts to mitigate the risk of generating incorrect or misleading information. To implement this human-centric approach, we developed a tool called the Manubot AI Editor, which is part of the Manubot infrastructure for scholarly publishing [10].

Overview of the Manubot AI Editor

a)



b) Section-specific prompt generator:

Revise the following paragraph from the abstract of an academic paper (with the title 'Manuscript title' and keywords 'keyword1, keyword2, …') so the research problem/question is clear, the solution proposed is clear, the text grammar is correct, spelling errors are fixed, and the text is in active voice and has a clear sentence structure.

Revise the following paragraph from the Introduction… so most of the citations to other academic papers are kept, the text minimizes the use of jargon, ...

Revise the following paragraph from the Results… so most references to figures and tables are kept, the details are enough to clearly explain the outcomes, sentences are concise and to the point, ...

Revise the following paragraph from the Discussion… so most of the citations to other academic papers are kept, the text minimizes the use of jargon, ...

Revise the following paragraph from the Methods… so most of the citations to other academic papers are kept, most of the technical details are kept, most references to equations (such as “Equation (@id)”) are kept, all equations definitions (such as “$$ ... $$ {#id}”) are included, the most important symbols in equations are defined, ...

…

Figure 1: AI-based revision applied on a Manubot-based manuscript. a) A manuscript (written with Manubot) with different sections. **b)** The prompt generator integrates metadata using prompt templates to generate section-specific prompts for each paragraph. If a paragraph belongs to a non-standard section, then a default prompt will be used to perform a basic revision only. The prompt for the Methods section includes the formatting of equations with identifiers. All sections' prompts include these instructions: *“the text grammar is correct, spelling errors are fixed, and the text has a clear sentence structure”*, although these are only shown for abstracts. Our tool allows the user to provide a custom prompt instead of using the default ones shown here.

The Manubot AI Editor is an AI-based revision infrastructure integrated into Manubot [10], a tool for collaborative writing of scientific manuscripts. Manubot integrates with popular version control platforms such as GitHub, allowing authors to easily track changes and collaborate on writing in real time. Furthermore, Manubot automates the process of generating a formatted manuscript (e.g., HTML, PDF, DOCX; Figure 1a shows the HTML output). Built upon this modern and open paradigm, our Manubot AI Editor (https://github.com/manubot/manubot-ai-editor) includes three components: 1) a Python library that provides classes and functions to read the manuscript content and its

metadata, calls the LLM for automatic text revision, and writes the results back; 2) a GitHub Actions workflow that uses our Python library within GitHub to preserve provenance information for transparency; 3) a prompt generator that integrates the manuscript's metadata using prompt templates to generate section-specific prompts for each paragraph (Figure 1b).

The GitHub Actions workflow enables users to easily trigger an automated revision task on either the entire manuscript or specific sections of it. When the action is triggered, the manuscript is parsed by section and then by paragraph (Figure 1b), which are then passed to the language model along with a set of custom prompts. The model subsequently returns a revised version of the text. Our workflow leverages the GitHub API to generate a new pull request, allowing the user to review and modify the output before merging the changes into the manuscript. This workflow assigns text attribution to either the human user or the AI language model, which may be important in light of potential future legal decisions that could reshape the copyright landscape concerning the outputs of generative models.

We used the OpenAI API for access to these models. Since this API incurs a cost with each run that depends on the manuscript length, we implemented a workflow in GitHub Actions that can be manually triggered by the user. Our implementation allows users to tune the costs to their needs by enabling them to select specific sections for revision instead of the entire manuscript. Additionally, several model parameters can be adjusted to further tune costs, such as the language model version (including the current GPT-3.5 Turbo and GPT-4, and potentially newly published ones), how much risk the model will take, or the "quality" of the completions. For instance, using Davinci models, the cost per run is under $0.50 for most manuscripts. More details about the implementation, installation, and usage of the Manubot AI Editor can be found in the Supplementary Material.

Evaluations of AI-based revisions

Evaluation setup

We used five different manuscript for the evaluation of our AI-based revision workflow (see below), and during the prompt engineering phase (see below), we also used a unit testing framework to ensure that the revisions produced by our prompts met a minimum set of quality measures (see Supplementary Material).

We evaluated our AI-assisted revision workflow using two models from OpenAI: Davinci (`text-davinci-003`) and GPT-3.5 Turbo (`gpt-3.5-turbo`). The first one is based on GPT-3 Davinci models and used to be a production-ready model, although it was now succeeded by the new GPT-3.5 Turbo and GPT-4 models. We used the most capable GPT-4 Turbo model for evaluating the revisions (LLM-as-a-Judge).

Manuscripts

Table 1: Manuscripts used to evaluate the AI-based revision workflow. The title and keywords of a manuscript are used in prompts for revising paragraphs. IDs are used in the text to refer to them.

Manuscript ID	GitHub URL	Title	Keywords
CCC	greenelab/ccc-manuscript	An efficient not-only-linear correlation coefficient based on machine learning	correlation coefficient, nonlinear relationships, gene expression

Manuscript ID	GitHub URL	Title	Keywords
PhenoPLIER	greenelab/phenoplier_manuscript	Projecting genetic associations through gene expression patterns highlights disease etiology and drug mechanisms	genetic studies, functional genomics, gene co-expression, therapeutic targets, drug repurposing, clustering of complex traits
Manubot-AI	greenelab/manubot-gpt-manuscript	A publishing infrastructure for AI-assisted academic authoring	manubot, artificial intelligence, scholarly publishing, software
Epistasis	quinlan-lab/mutator-epistasis-manuscript	Epistasis between mutator alleles contributes to germline mutation rate variability in laboratory mice	-
BioChatter	biocypher/biochatter-paper	A Platform for the Biomedical Application of Large Language Models	biomedicine, large language models, framework, retrieval-augmented generation, knowledge graph

For the evaluation of our tool, we conducted manual assessments performed by humans and automatic assessments performed by an LLM. For the human assessments, we used three of our own manuscripts (Table 1): the Clustermatch Correlation Coefficient (CCC) [11], PhenoPLIER [12], and Manubot-AI (this manuscript). CCC is a new correlation coefficient applied to transcriptomic data, while PhenoPLIER is a framework consisting of three different methods used in genetic studies. CCC falls under the field of computational biology, whereas PhenoPLIER pertains to genomic medicine. CCC outlines one computational method applied to a specific data type (correlation to gene expression). In contrast, PhenoPLIER describes a framework that integrates three different approaches (regression, clustering, and drug-disease prediction) using data from genome-wide and transcription-wide association studies (GWAS and TWAS), gene expression, and transcriptional responses to small molecule perturbations. Thus, CCC has a simpler structure, while PhenoPLIER is a more complex manuscript with additional figures and tables, along with a Methods section that includes equations. The third manuscript, Manubot-AI, has a much simpler structure and was written and revised using our tool prior to submission, demonstrating a practical AI-based revision use case. For the automatic assessments, we incorporated two external manuscripts (with IDs BioChatter and Epistasis in Table 1).

Evaluation using human assessments

We enabled the Manubot AI revision workflow in the GitHub repositories of the three manuscripts (CCC, PhenoPLIER, and Manubot-AI). This added the “ai-revision” workflow to the “Actions” tab of each repository. We manually triggered the workflow and utilized the `text-davinci-003` language model to generate one pull request (PR) per manuscript. These PRs can be accessed from the “Pull requests” tab of each repository. The PRs display all the differences between the original text and the AI-based revision suggestions.

When manually assessing the quality of the revisions, we considered whether the revision: 1) preserve the original meaning, 2) preserve important details, 4) introduced new and incorrect information, and 5) preserve the correct Markdown format (e.g., citations, equations).

Evaluation using an LLM as a judge

For this evaluation, we ran our workflow on manuscripts CCC, PhenoPLIER, BioChatter, and Epistasis using the GPT-3.5 Turbo model (`gpt-3.5-turbo`). We then inspected each PR and manually matched all pairs of original and revised paragraphs, across the abstract, introduction, methods, results, and supplementary material sections. This procedure generated 31 paragraph pairs for CCC, 63 for PhenoPLIER, 37 for BioChatter, and 63 for Epistasis. Using the LLM-as-a-Judge method [13], we evaluated the quality of the revisions using both GPT-3.5 Turbo (`gpt-3.5-turbo`) and GPT-4 Turbo (`gpt-4-turbo-preview`) as judges. The judge is asked to decide which of the two paragraphs in each pair is better or if they are equally good (tie). For this, we used prompt chaining, where the judge first evaluates the quality of each paragraph independently by writing a list with positive and negative aspects in the following areas: 1) clear sentence structure, 2) ease of understanding, 3) grammatical correctness, 4) absence of spelling errors. Then, the judge was asked to be as objective as possible and decide if one of the paragraphs is clearly better than the other or if they are similar in quality, while also providing a rationale for the decision. We also accounted for the case of position bias [13] (i.e., the order in which the paragraphs were presented could influence the decision) by swapping the order of the paragraphs. Each assessment was repeated ten times. The full prompt chain can be seen in Supplementary File 4, which includes an example of the output in each step generated by GPT-4 Turbo as a judge.

Human assessments across different sections

Following our criteria for human assessments (see above), we inspected the PRs generated by the AI-based workflow and reported on our assessment of the changes suggested by the tool across different sections of the manuscripts. The reader can access the PRs in the manuscripts' GitHub repositories (Table 1) and also included as diff files in Supplementary File 1 (CCC), 2 (PhenoPLIER), and 3 (Manubot-AI).

Below, we present the differences between the original text and the revisions made by the tool in a `diff` format (obtained from GitHub). Line numbers are included to show the length differences. Unless the AI suggestions represent a complete overhaul of the text, single words are underlined and highlighted in colors to more clearly see the differences within a single sentence. Red indicates words removed by the tool, while green indicates words added; no underlining indicates words kept unchanged. In the GitHub repositories, the full diffs can be seen by clicking on the "Files changed" tab under each PR.

Abstract

1	- Correlation coefficients are widely used to identify patterns in data that may be of particular interest.	1	+ This paper presents the Clustermatch Correlation Coefficient (CCC), an efficient and not-only-linear correlation coefficient based on machine learning models, to identify linear and nonlinear patterns in transcriptomics data.
2	- In transcriptomics, genes with correlated expression often share functions or are part of disease-relevant biological processes.	2	+ We aim to determine if CCC can detect meaningful linear and nonlinear relationships in gene expression data, including those missed by linear-only correlation coefficients, and if highly-ranked gene pairs by CCC are enriched for interactions in integrated networks.
3	- Here we introduce the Clustermatch Correlation Coefficient (CCC), an efficient, easy-to-use and not-only-linear coefficient based on machine learning models.	3	+ When applied to human gene expression data, CCC identifies robust linear relationships and nonlinear patterns associated with sex differences.
4	- CCC reveals biologically meaningful linear and nonlinear patterns missed by standard, linear-only correlation coefficients.	4	+ Our results suggest that CCC can detect functional relationships not captured by linear-only methods.
5	- CCC captures general patterns in data by comparing clustering solutions while being much faster than state-of-the-art coefficients such as the Maximal Information Coefficient.	5	+ CCC is a highly-efficient, next-generation not-only-linear correlation coefficient that can be applied to genome-scale data and other domains across different data types.
6	- When applied to human gene expression data, CCC identifies robust linear relationships while detecting nonlinear patterns associated, for example, with sex differences that are not captured by linear-only coefficients.		
7	- Gene pairs highly ranked by CCC were enriched for interactions in integrated networks built from protein-protein interaction, transcription factor regulation, and chemical and genetic perturbations, suggesting that CCC could detect functional relationships that linear-only methods missed.		
8	- CCC is a highly-efficient, next-generation not-only-linear correlation coefficient that can readily be applied to genome-scale data and other domains across different data types.		

Figure 2: Abstract of CCC. Original text is on the left and suggested revision on the right. Single words are not underlined/highlighed in this case because the revision completely overhauled the text.

We applied the AI-based revision workflow to the CCC abstract (Figure 2). The tool completely rewrote the text, leaving only the last sentence mostly unchanged. The text was significantly shortened, and the sentences were longer than those in the original, which could make the abstract slightly harder to read. The revision removed the first two sentences, which introduced correlation analyses and transcriptomics, and directly stated the purpose of the manuscript. It also removed details about the method (line 5) and focused on the aims and results obtained, ending with the same last sentence, suggesting a broader application of the coefficient to other data domains (as originally intended by the authors of CCC). The main concepts were still present in the revised text.

The revised text for the abstract of PhenoPLIER was significantly shortened (from 10 sentences in the original, to only 3 in the revised version). However, in this case, important concepts (such as GWAS, TWAS, CRISPR) and a proper amount of background information were missing, producing a less informative abstract.

Introduction

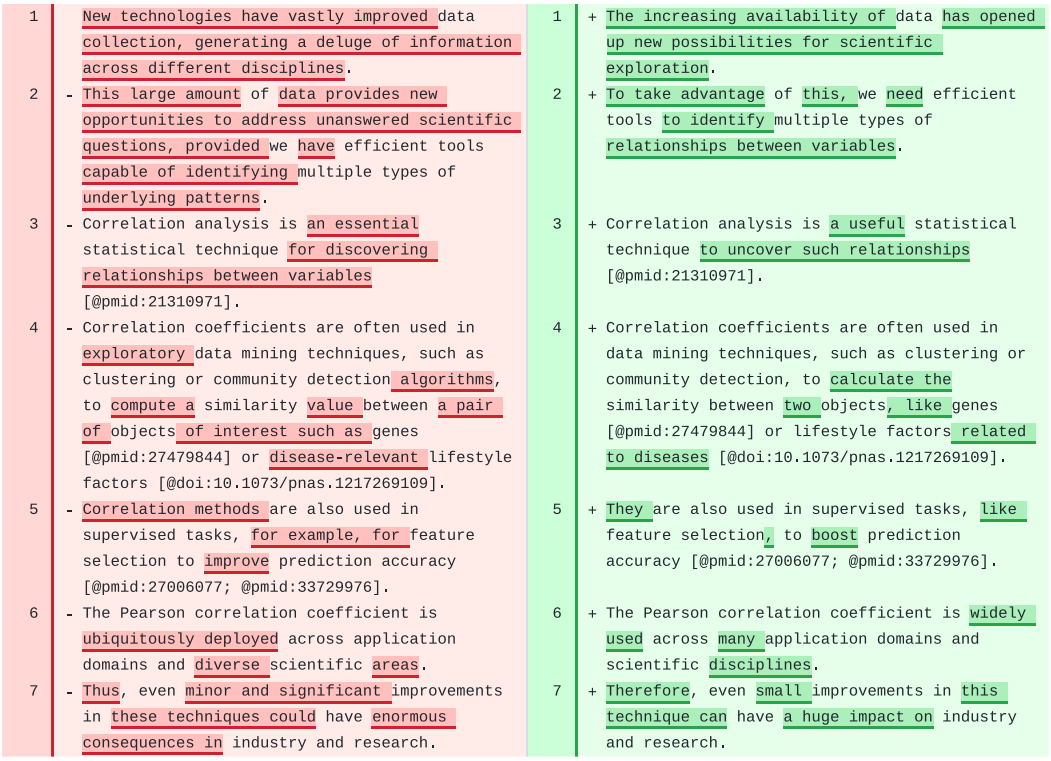


Figure 3: First paragraph in the Introduction section of CCC. Original text is on the left and suggested revision on the right.

The tool significantly revised the Introduction section of CCC (Figure 3), producing a more concise and clear introductory paragraph. The revised first sentence concisely incorporated ideas from the original two sentences, introducing the concept of “large datasets” and the opportunities for scientific exploration. The model generated a more concise second sentence introducing the “need for efficient tools” to find “multiple relationships” in these datasets. The third sentence connected nicely with the previous one. All references to scientific literature were kept in the correct Manubot format, even though our prompts do not specify the references format. The rest of the sentences in this section were also correctly revised and could be incorporated into the manuscript with minor or no further changes.

We also observed a high-quality revision of the introduction of PhenoPLIER. However, the model failed to maintain the format of citations in one paragraph. Additionally, the model did not converge to a revised text for the last paragraph, and our tool left an error message as an HTML comment at the top: `The AI model returned an empty string`. Debugging the prompts revealed this issue, which could be related to the complexity of the paragraph. In these cases, rerunning the automated revision might solve this type of issue.

Results

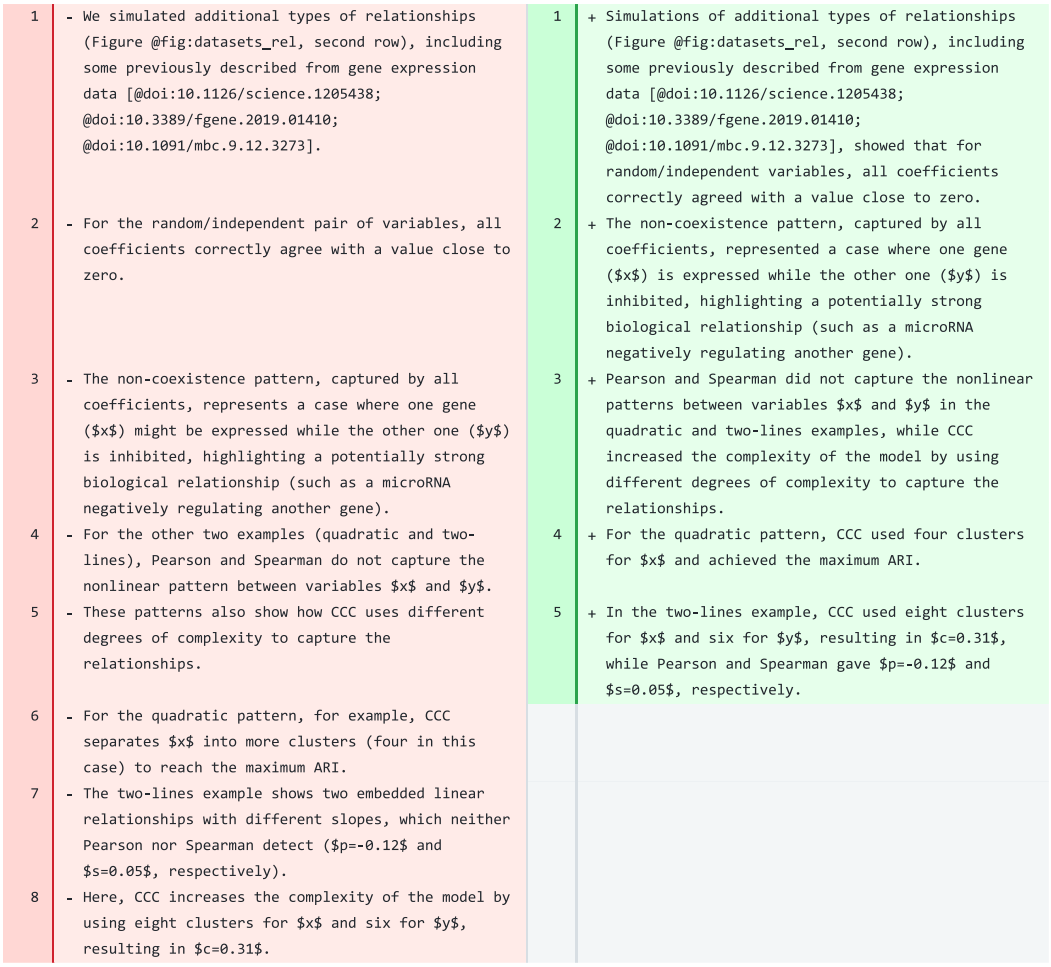


Figure 4: A paragraph in the Results section of CCC. Original text is on the left and suggested revision on the right. Single words are not underlined/highlighed in this case because the revision completely overhauled the text.

We tested the tool on a paragraph from the Results section of CCC (Figure 4). This paragraph describes Figure 1 of the CCC manuscript [11], which showcases four different datasets, each with two variables, and various relationships or patterns labeled as random/independent, non-coexistence, quadratic, and two-lines. The revised paragraph, while having fewer sentences, is slightly longer and consistently uses the past tense, unlike the original one which has tense shifts. The revised paragraph also retains all citations, which, although not explicitly mentioned in the prompts for this section (as it is for introductions), is important in this case. The original LaTeX format was maintained for the math, and the figure was correctly referenced using the Manubot syntax. In the third sentence of the revised paragraph (line 3), the model generated a good summary of how all coefficients performed in the last two nonlinear patterns, and why CCC was able to capture them. As human authors, we would make a single change at the end of this sentence to avoid repeating the word "complexity": *"..., while CCC increased the model's complexity ~~by using different degrees of complexity~~ to capture the relationships."* The revised paragraph is more concise and clearly describes what the figure shows and how CCC works. It's remarkable that the model rewrote some of the concepts in the original paragraph (lines 4 to 8) into three new sentences (lines 3 to 5) with the same meaning, but more concisely and clearly. The model also produced high-quality revisions for several other paragraphs that would only need minor changes.

However, other paragraphs in CCC required extensive changes before they could be incorporated into the manuscript. For instance, the model generated revised text for certain paragraphs that was more concise, direct, and clear. However, this often resulted in the removal of important details and occasionally altered the intended meaning of sentences. To address this issue, we could accept the simplified sentence structure proposed by the model but reintroduce the missing details for clarity and completeness.

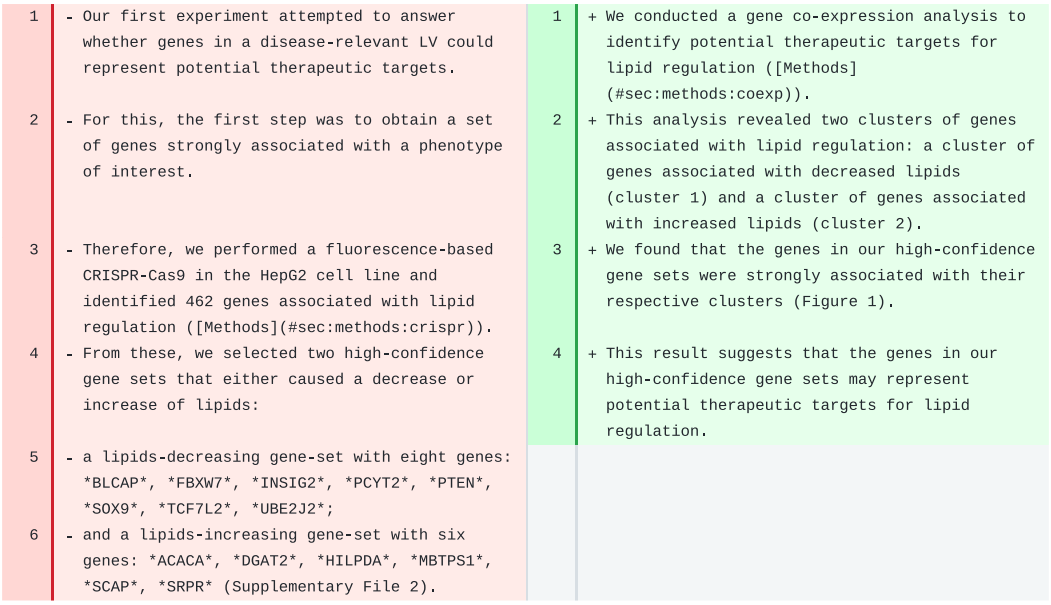


Figure 5: A paragraph in the Results section of PhenoPLIER. Original text is on the left and suggested revision on the right. Single words are not underlined/highlighed in this case because the revision completely overhauled the text.

When applied to the PhenoPLIER manuscript, the model produced high-quality revisions for most paragraphs while preserving citations and references to figures, tables, and other sections of the manuscript in the Manubot/Markdown format. In some cases, important details were missing, but they could be easily added back while preserving the improved sentence structure of the revised version. In other cases, the model's output demonstrated the limitations of revising one paragraph at a time without considering the rest of the text. For instance, one paragraph described our CRISPR screening approach to assess whether top genes in a latent variable (LV) could represent good therapeutic targets. The model generated a paragraph with a completely different meaning (Figure 5). It omitted the CRISPR screen and the gene symbols associated with the regulation of lipids, which were key elements in the original text. Instead, the new text describes an experiment that does not exist, with a reference to a non-existent section. This suggests that the model focused on the title and keywords of the manuscript (Table 1) that were part of every prompt (Figure 1). For example, it included the idea of "gene co-expression" analysis (a keyword) to identify "therapeutic targets" (another keyword) and replaced the mention of "sets of genes" in the original text with "clusters of genes" (closer to the keyword including "clustering"). This was a poor model-based revision, indicating that the original paragraph might be too short or disconnected from the rest and could be merged with the next one, which describes follow-up and related experiments.

Discussion

In both the CCC and PhenoPLIER manuscripts, revisions to the discussion section appeared to be of high quality. The model kept the correct format when necessary (e.g., using italics for gene symbols), maintained most of the citations, and improved the readability of the text in general. Revisions for some paragraphs introduced minor mistakes that a human author could readily fix.

Figure 6: A paragraph in the Discussion section of CCC. Original text is on the left and suggested revision on the right.

One paragraph from CCC discusses how not-only-linear correlation coefficients could potentially impact genetic studies of complex traits (Figure 6). Although some minor changes could be incorporated, we believe the revised text reads better than the original. It is also interesting to see how the model understood the format of citations and built more complex structures from it. For instance, the two articles referenced in lines 2 and 3 of the original text were correctly merged into a single citation block and separated with a ";" in line 2 of the revised text.

Methods

Prompts for the Methods section were the most challenging to design, especially when the sections included equations. The prompt for Methods (Figure 1) is more focused in keeping the technical details, which was especially important for PhenoPLIER, whose Methods section contains paragraphs with several mathematical expressions.

Figure 7: A paragraph in the Methods section of PhenoPLIER. Original text is on the left and suggested revision on the right.

We revised a paragraph in PhenoPLIER that contained two numbered equations (Figure 7). The model made very few changes, and all the equations, citations, and most of the original text were preserved. However, we found it remarkable how the model identified an incorrect reference to a mathematical symbol (line 8) and corrected it in the revision (line 7). Indeed, the equation with the univariate model used by PrediXcan (lines 4-6 in the original) includes the *true* effect size γ_l (`\gamma_l`) instead of the *estimated* one $\hat{\gamma}_l$ (`\hat{\gamma}_l`).

In PhenoPLIER, we found one large paragraph with several equations that the model failed to revise, although it performed relatively well in revising the rest of the section. In CCC, the revision of this section was good overall, with some minor and easy-to-fix issues as in the other sections.

We also observed issues arising from revising one paragraph at a time without context. For instance, in PhenoPLIER, one of the first paragraphs mentions the linear models used by S-PrediXcan and S-MultiXcan without providing any equations or details. These were presented in the following paragraphs, but since the model had not yet encountered that information, it opted to add those equations immediately (in the correct Manubot/Markdown format).

1	- We implemented the AI-based revision infrastructure in Manubot [@doi:10.1371/journal.pcbi.1007128].	1	+ We implemented an AI-based revision infrastructure in Manubot [@doi:10.1371/journal.pcbi.1007128], a tool for collaborative writing of scientific manuscripts.
2	- Manubot is a tool for collaborative writing of scientific manuscripts.		
3	- It utilizes version control and a continuous integration workflow to facilitate efficient and transparent collaboration among authors.		
4	Manubot integrates with popular version control platforms such as GitHub, allowing authors to easily track changes and collaborate on writing in real time.	2	Manubot integrates with popular version control platforms such as GitHub, allowing authors to easily track changes and collaborate on writing in real time.
5	- Additionally, Manubot automates the process of generating a formatted manuscript (such as HTML, PDF, DOCX; Figure {@fig:ai_revision}a shows the HTML output), reducing the time and effort required for manuscript preparation and submission.	3	+ Furthermore, Manubot automates the process of generating a formatted manuscript (such as HTML, PDF, DOCX; Figure {@fig:ai_revision}a).
6	Built on this modern and open paradigm, our AI-based revision software was built using GitHub Actions, which allows the user to easily trigger an automated revision task on the entire manuscript or specific sections of it.	4	Built on this modern and open paradigm, our AI-based revision software was built using GitHub Actions, which allows the user to easily trigger an automated revision task on the entire manuscript or specific sections of it.
		5	+ The AI-based revision task is based on a machine learning model trained on a corpus of scientific papers from the same field as the manuscript (Figure {@fig:ai_revision}b).
		6	+ This model is used to identify and suggest revisions to the text, resulting in a modified version of the manuscript that is ready for submission.
		7	+ This reduces the time and effort required for manuscript preparation and submission.

Figure 8: A paragraph in the Methods section of ManubotAI. Original text is on the left and suggested revision on the right. The revision (right) contains a repeated set of sentences at the top that we removed to improve the clarity of the figure.

When revising the Methods sections of Manubot-AI (this manuscript), the model, in some cases, added novel sentences containing incorrect information. For example, for one paragraph, it included a formula (using the correct Manubot format) presumably to predict the cost of a revision run. In another paragraph (Figure 8), it introduced new sentences stating that the model was *"trained on a corpus of scientific papers from the same field as the manuscript"* and that its suggested revisions resulted in a *"modified version of the manuscript that is ready for submission."* Although these are important future directions, neither statement accurately describes the present work.

Automated assessments

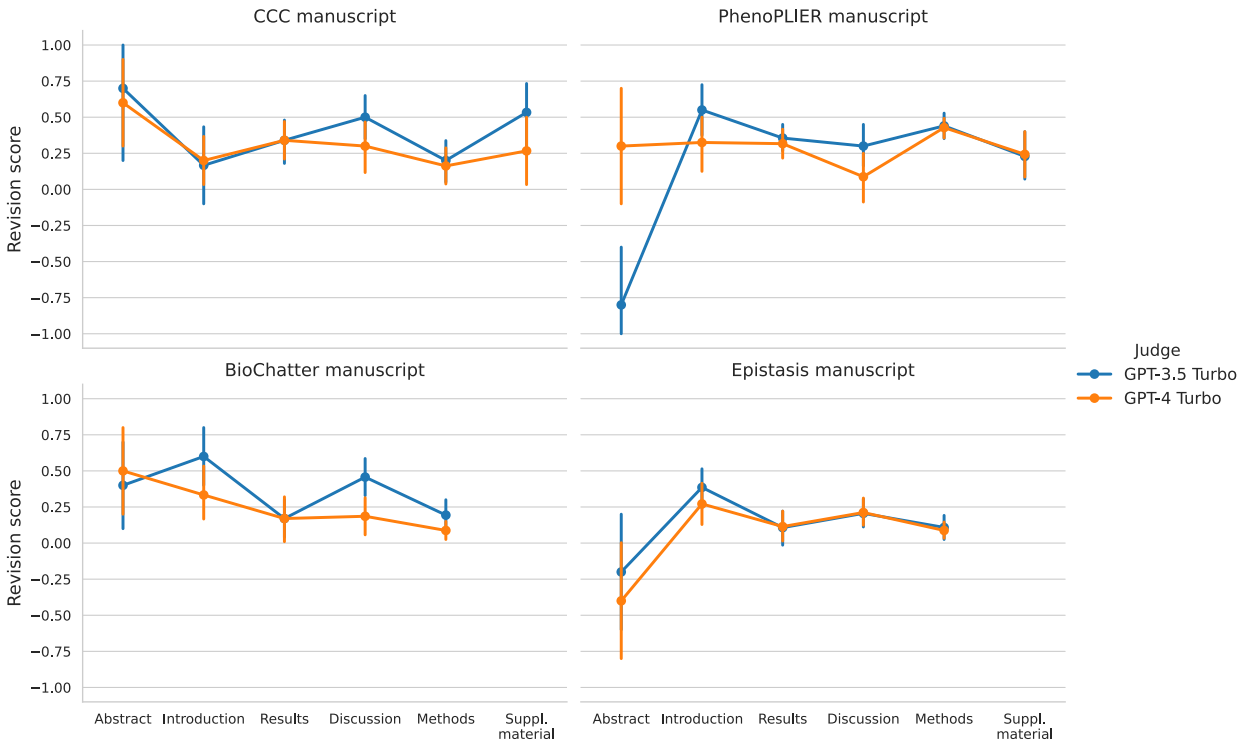


Figure 9: Automated assessment of preference over revised paragraphs. A revision score (y-axis) close to 1 indicates that the LLM acting as a judge preferred the revised paragraph over the original one, while a score of -1 indicates the opposite; a score close to zero indicates either a tie or position bias. Each point represents the average score of paragraphs from a section in one of the four manuscripts: CCC, PhenoPLIER, BioChatter and Epistasis.

The automatic assessment of paragraphs from different sections across four manuscripts is depicted in Figure 9. A revision score above zero indicates that the LLM acting as a judge preferred the revised paragraph over the original one on average, while a score below zero indicates the opposite. It can be seen that the two models used as judges, GPT-3.5 Turbo and GPT-4 Turbo, generally agreed and favored the revised paragraphs over the original ones (revision score above zero) in most cases. The only section where the original paragraphs were clearly preferred was the Abstract of the PhenoPLIER and Epistasis manuscripts. GPT-3.5 Turbo showed a preference for the original abstract of PhenoPLIER in most cases, and the model rationale (Supplementary File 5) aligns with our human assessment: the original abstract provides a more "detailed explanation" of the approaches and a "comprehensive overview of the research."

Conclusions

Our tool, the Manubot AI Editor, integrates AI-based revision models into the Manubot publishing platform. Writing academic papers can be time-consuming and challenging to comprehend, so we aimed to use technology to assist researchers in communicating their findings more effectively. Our AI-based revision workflow uses a prompt generator that creates manuscript- and section-specific instructions for the language model. Authors can easily trigger this workflow from the GitHub repository to suggest revisions that can be reviewed later. This workflow utilizes OpenAI models, generating a pull request of revisions for authors to review. We have established default parameters for these models that perform well for our use cases across different sections and manuscripts. Users also have the option to customize the revision process by selecting specific sections, adjusting the model's behavior to suit their needs and budget, and even providing custom prompts instead of using the default, section-specific ones. This can be particularly beneficial for specific use cases that do not require a complex revision. Although evaluating automatic text revision is challenging, we conducted both human and automated evaluations of the revisions generated by the AI model. We found that most paragraphs were enhanced, while in some cases the model removed important information or introduced errors. The AI model also highlighted certain paragraphs that were difficult to revise, which could pose challenges for human readers as well.

Our approach has some limitations. We found that revising abstracts proved more challenging for the model, as revisions often removed background information about the research problem. There are opportunities to improve the AI-based revisions, such as further refining prompts using few-shot learning [14], or fine-tuning the model using an additional corpus of academic writing focused on particularly challenging sections. Fine-tuning using preprint-publication pairs [15] may help to identify sections or phrases likely to be changed during peer review. Our approach processed each paragraph of the text but lacked a contextual thread between queries, which mainly affected the Results and Methods sections. Using chatbots that retain context could enable the revision of individual paragraphs while considering previously processed text. We plan to update our workflow to support this strategy. Regarding the LLM used, open and semi-open models, such as BLOOM [16], Meta's Llama 2 [17], and Mistral 7B [18], are growing in popularity and capabilities, but they lack the user-friendly OpenAI API. We used the LLM-as-a-Judge method to automatically assess the quality of revisions, which has limitations such as the self-enhancement bias where LLMs tend to favor text generated by themselves. Although our approach is based on revising human-generated text (rather than generating answers from scratch), we used two LLM judges, GPT-3.5 and GPT-4, to address this potential issue. These two models have shown limited self-enhancement bias and high alignment with human preferences [13]. In this study, we found that the automated assessments were consistent with our human evaluations. Despite these limitations, we found that models captured the main ideas

and generated a revision that often communicated the intended meaning more clearly and concisely. While our study focused on OpenAI's GPT-3 and GPT-3.5 Turbo for revisions, the Manubot AI Editor is prepared to support future models.

The use of AI-assisted tools for scientific authoring is controversial [19,20]. Questions arise concerning the originality and ownership of texts generated by these models. For example, the *Nature* journal has established that any use of these models in scientific writing must be documented [21], and the International Conference on Machine Learning (ICML) has prohibited the submission of *"papers that include text generated from a large-scale language model (LLM)"* [22], although editing tools for grammar and spelling correction are allowed. Our work, however, focuses on revising *existing* text written by a human author. Additionally, all changes made by humans and AI are tracked in the version control system, which allows for full transparency. Despite the concerns, there are also significant opportunities. Our work lays the foundation for a future in which humans and machines construct academic manuscripts together. Scientific articles need to adhere to a certain style, which can make the writing time-consuming and require a significant amount of effort to think about *how* to communicate a result or finding that has already been obtained. As machines become increasingly capable of improving scholarly text, humans can focus more on *what* to communicate to others, rather than on *how* to write it. This could lead to a more equitable and productive future for research, where scientists are only limited by their ideas and ability to conduct experiments to uncover the underlying organizing principles of ourselves and our environment.

References

1. **A history of scientific & technical periodicals: the origins and development of the scientific and technical press, 1665-1790**
   David A Kronick
   Scarecrow Press (1976)
   ISBN: 9780810808447

2. **The history of the peer-review process**
   Ray Spier
   Trends in Biotechnology (2002-08) https://doi.org/d26d8b
   DOI: 10.1016/s0167-7799(02)01985-6 · PMID: 12127284

3. **How to write a first-class paper**
   Virginia Gewin
   Nature (2018-02-28) https://doi.org/ggh63n
   DOI: 10.1038/d41586-018-02404-4

4. **Language Models are Few-Shot Learners**
   Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, … Dario Amodei
   arXiv (2020-07-24) https://arxiv.org/abs/2005.14165

5. **Using AI-generated suggestions from ChatGPT to optimize clinical decision support**
   Siru Liu, Aileen P Wright, Barron L Patterson, Jonathan P Wanderer, Robert W Turer, Scott D Nelson, Allison B McCoy, Dean F Sittig, Adam Wright
   Journal of the American Medical Informatics Association (2023-04-22) https://doi.org/gsgvw2
   DOI: 10.1093/jamia/ocad072 · PMID: 37087108 · PMCID: PMC10280357

6. **Benchmarking the symptom-checking capabilities of ChatGPT for a broad range of diseases**
   Anjun Chen, Drake O Chen, Lu Tian
   Journal of the American Medical Informatics Association (2023-12-18)
   https://doi.org/10.1093/jamia/ocad245

7. **Could AI help you to write your next paper?**
   Matthew Hutson
   Nature (2022-10-31) https://doi.org/grpm4w
   DOI: 10.1038/d41586-022-03479-w · PMID: 36316468

8. **ChatGPT and the clinical informatics board examination: the end of unproctored maintenance of certification?**
   Yaa Kumah-Crystal, Scott Mankowitz, Peter Embi, Christoph U Lehmann
   Journal of the American Medical Informatics Association (2023-06-19) https://doi.org/gs9km8
   DOI: 10.1093/jamia/ocad104 · PMID: 37335851 · PMCID: PMC10436139

9. **AI in health: keeping the human in the loop**
   Suzanne Bakken
   Journal of the American Medical Informatics Association (2023-06-20) https://doi.org/gs9km7
   DOI: 10.1093/jamia/ocad091 · PMID: 37337923 · PMCID: PMC10280340

10. **Open collaborative writing with Manubot**
    Daniel S Himmelstein, Vincent Rubinetti, David R Slochower, Dongbo Hu, Venkat S Malladi, Casey S Greene, Anthony Gitter

PLOS Computational Biology (2019-06-24) https://doi.org/c7np
DOI: 10.1371/journal.pcbi.1007128 · PMID: 31233491 · PMCID: PMC6611653

11. **An efficient not-only-linear correlation coefficient based on machine learning**
Milton Pividori, Marylyn D Ritchie, Diego H Milone, Casey S Greene
Cold Spring Harbor Laboratory (2022-06-17) https://doi.org/gqcvbw
DOI: 10.1101/2022.06.15.496326

12. **Projecting genetic associations through gene expression patterns highlights disease etiology and drug mechanisms**
Milton Pividori, Sumei Lu, Binglan Li, Chun Su, Matthew E Johnson, Wei-Qi Wei, Qiping Feng, Bahram Namjou, Krzysztof Kiryluk, Iftikhar J Kullo, …
Nature Communications (2023-09-09) https://doi.org/gspsxr
DOI: 10.1038/s41467-023-41057-4 · PMID: 37689782 · PMCID: PMC10492839

13. **Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena**
Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, EricP Xing, … Ion Stoica
arXiv (2023-10-17) https://arxiv.org/abs/2306.05685

14. **Generalizing from a Few Examples**
Yaqing Wang, Quanming Yao, James T Kwok, Lionel M Ni
ACM Computing Surveys (2020-06-12) https://doi.org/gg37m2
DOI: 10.1145/3386252

15. **Examining linguistic shifts between preprints and publications**
David N Nicholson, Vincent Rubinetti, Dongbo Hu, Marvin Thielk, Lawrence E Hunter, Casey S Greene
PLOS Biology (2022-02-01) https://doi.org/gqqzn2
DOI: 10.1371/journal.pbio.3001470 · PMID: 35104289 · PMCID: PMC8806061

16. **BLOOM: A 176B-Parameter Open-Access Multilingual Language Model**
BigScience Workshop, :, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, … Thomas Wolf
arXiv (2023-06-28) https://arxiv.org/abs/2211.05100

17. **Llama 2: Open Foundation and Fine-Tuned Chat Models**
Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, … Thomas Scialom
arXiv (2023-07-20) https://arxiv.org/abs/2307.09288

18. **Mistral 7B**
Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, … William El Sayed
arXiv (2023-10-11) https://arxiv.org/abs/2310.06825

19. **Abstracts written by ChatGPT fool scientists**
Holly Else
Nature (2023-01-12) https://doi.org/js2g
DOI: 10.1038/d41586-023-00056-7 · PMID: 36635510

20. **ChatGPT listed as author on research papers: many scientists disapprove**
Chris Stokel-Walker
Nature (2023-01-18) https://doi.org/grn72b
DOI: 10.1038/d41586-023-00107-z · PMID: 36653617

21. **Tools such as ChatGPT threaten transparent science; here are our ground rules for their use**
Nature
(2023-01-24) https://doi.org/grpm2s
DOI: 10.1038/d41586-023-00191-1 · PMID: 36694020

22. **ICML 2023** https://icml.cc/Conferences/2023/llm-policy

Supplementary Material

Installation and use

The Manubot AI Editor is part of the standard Manubot template manuscript, referred to as rootstock, and is available at https://github.com/manubot/rootstock. Users wishing to use the workflow only need to follow the standard procedures to install Manubot. The section "AI-assisted authoring," found in the file `USAGE.md` of the rootstock repository, explains how to enable the tool. Afterward, the workflow (named `ai-revision`) will be available and ready to use under the Actions tab of the user's manuscript repository.

Implementation details

To run the workflow, the user must specify the branch that will be revised, select the files/sections of the manuscript (optional), specify the language model to use, provide an optional custom prompt (section-specific prompts are used by default), and provide the output branch name. For more advanced users, it is also possible to modify most of the tool's behavior or the language model parameters.

When the workflow is triggered, it downloads the manuscript by cloning the specified branch. It revises all of the manuscript files, or only some of them if the user specifies a subset. Next, each paragraph in the file is read and submitted to the OpenAI API for revision. If the request is successful, the tool will write the revised paragraph in place of the original one, using one sentence per line (which is the recommended format for the input text). If the request fails, the tool might try again (up to five times by default) if it is a common error (such as "server overloaded") or a model-specific error that requires changing some of its parameters. If the error cannot be handled or the maximum number of retries is reached, the original paragraph is written instead, with an HTML comment at the top explaining the cause of the error. This allows the user to debug the problem and attempt to fix it if desired.

As shown in Figure 1b, each API request comprises a prompt (the instructions given to the model) and the paragraph to be revised. Unless the user specifies a custom prompt, the tool will use a section-specific prompt generator that incorporates the manuscript title and keywords. Therefore, both must be accurate to obtain the best revision outcomes. The other key component to process a paragraph is its section. For instance, the abstract is a set of sentences with no citations, whereas a paragraph from the Introduction section has several references to other scientific papers. A paragraph in the Results section has fewer citations but many references to figures or tables and must provide enough details about the experiments to understand and interpret the outcomes. The Methods section is more dependent on the type of paper, but in general, it has to provide technical details and sometimes mathematical formulas and equations. Therefore, we designed section-specific prompts, which we found led to the most useful suggestions. Figure and table captions, as well as paragraphs that contain only one or two sentences and fewer than sixty words, are not processed and are copied directly to the output file.

The section of a paragraph is automatically inferred from the file name using a simple strategy, such as if "introduction" or "methods" is part of the file name. If the tool fails to infer a section from the file, the user can still specify to which section the file belongs. The section can be a standard one (abstract, introduction, results, methods, or discussion) for which a specific prompt is used (Figure 1b), or a non-standard one for which a default prompt is used to instruct the model to perform basic revision. This includes *"minimizing the use of jargon, ensuring text grammar is correct, fixing spelling errors, and making sure the text has a clear sentence structure."*

Properties of language models

The Manubot AI Editor uses the Chat Completions API to process each paragraph. We have tested our tool using the Davinci (`text-davinci-003`, based on the initial GPT-3 models) and GPT-3.5 Turbo models (`gpt-3.5-turbo`). All models can be adjusted using different parameters (refer to OpenAI - API Reference), and the most important ones can be easily adjusted using our tool.

Language models for text completion have a context length that indicates the limit of tokens they can process (tokens are common character sequences in text). This limit includes the size of the prompt and the paragraph, as well as the maximum number of tokens to generate for the completion (parameter `max_tokens`). To ensure we never exceed this context length, our AI-assisted revision software processes each paragraph of the manuscript with section-specific prompts, as shown in Figure 1b. This approach allows us to process large manuscripts by breaking them into smaller chunks of text. However, since the language model only processes a single paragraph from a section, it can potentially lose the context needed to produce a better output. Nonetheless, we find that the model still produces high-quality revisions (see Results). Additionally, the maximum number of tokens (parameter `max_tokens`) is twice the estimated number of tokens in the paragraph (one token approximately represents four characters, see OpenAI - Tokenizer). The tool automatically adjusts this parameter and performs the request again if a related error is returned by the API. The user can also force the tool to either use a fixed value for `max_tokens` for all paragraphs or change the fraction of maximum tokens based on the estimated paragraph size (two by default).

The language models used are stochastic, meaning they generate a different revision for the same input paragraph each time. This behavior can be adjusted by using the "sampling temperature" or "nucleus sampling" parameters (we use `temperature=0.5` by default). Although we selected default values that work well across multiple manuscripts, these parameters can be changed to make the model more deterministic. The user can also instruct the model to generate several completions and select the one with the highest log probability per token, which can improve the quality of the revision. Our implementation generates only one completion (parameter `best_of=1`) to avoid potentially high costs for the user. Additionally, our workflow allows the user to process either the entire manuscript or individual sections. This provides more cost-effective control while focusing on a single piece of text, wherein the user can run the tool several times and pick the preferred revised text.

Prompt engineering

We extensively tested our tool, including prompts, using a unit testing framework. Our unit tests cover the general processing of the manuscript content (such as splitting by paragraphs), the generation of custom prompts using the manuscript metadata, and writing back the text suggestions (ensuring that the original style is preserved as much as possible to minimize the number of changes). More importantly, they also cover some basic quality measures of the revised text. This latter set of unit tests was used during our prompt engineering work, and they ensure that section-specific prompts yield revisions with a minimum set of quality measures. For instance, we wrote unit tests to check that revised Abstracts consist of a single paragraph, start with a capital letter, end with a period, and that no citations to other articles are included. For the Introduction section, we check that a certain percentage of citations are kept, which also attempts to give the model some flexibility to remove text deemed unnecessary. We found that adding the instruction *"most of the citations to other academic papers are kept"* to the prompt was enough to achieve this with the most capable model. We also wrote unit tests to ensure the models returned citations in the correct Manubot/Markdown format (e.g., `[@doi:...]` or `[@arxiv:...]`), and found that no changes to the prompt were needed for this (i.e., the model automatically detected the correct format in most cases). For the Results section, we included tests with short inline formulas in LaTeX (e.g., `$\gamma_l$`) and references to figures,

tables, equations, or other sections (e.g., `Figure @id` or `Equation (@id)`) and found that, in the majority of cases, the most capable model was able to correctly keep them with the right format. For the Methods section, in addition to the aforementioned tests, we also evaluated the ability of models to use the correct format for the definition of numbered, multiline equations, and found that the most capable model succeeded in most cases. For this particular case, we needed to modify our prompt to explicitly mention the correct format of multiline equations (see prompt for Methods in Figure 1).

We also included tests where the model is expected to fail in generating a revision (for instance, when the input paragraph is too long for the model's context length). In these cases, we ensure that the tool returns a proper error message. We ran our unit tests across all models under evaluation.