A publishing infrastructure for AI-assisted academic authoring

This manuscript (permalink) was automatically generated from greenelab/manubot-gpt-manuscript@47f5b02 on January 7, 2023.

Authors

- **Milton Pividori**
  0000-0002-3035-4403 · miltondp · miltondp
  Department of Genetics, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA 19104, USA

- **Casey S. Greene** ✉
  0000-0001-8713-9213 · cgreene · GreeneScientist
  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

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

Abstract

Academics often communicate through scholarly manuscripts. These manuscripts describe new advances, summarize existing literature, or argue for changes in the status quo. Writing and revising manuscripts can be a time-consuming process. Large language models are bringing new capabilities to many areas of knowledge work. We integrated the use of large language models into the Manubot publishing ecosystem. Users of Manubot can run a workflow, which will trigger a series of queries to OpenAI's language models, produce revisions, and create a timestamped set of suggested revisions. Given the amount of time that researchers put into crafting prose, we expect this advance to radically transform the type of knowledge work that academics perform.

Introduction

The manuscript pre-dates the invention of printing by thousands of years, but the practice of producing exclusively scientific journals only started roughly 350 years ago [1]. The implementation of external peer review varies by journal but for many is less than 100 years old [2]. To date, most manuscripts have been written by humans or teams of humans working together to describe scholarly advances.

Modern scholarly manuscripts often describe new advances, summarize existing literature, or argue for changes in the status quo. However, writing and revising can be a time-consuming process. Academics can sometimes be long-winded in getting to key points, making writing more impenetrable to their audience [3].

Modern computing capabilities and the widespread availability of text, images, and other data on the internet has laid the foundation for artificial intelligence (AI) models with many parameters. Large language models, in particular, are opening the floodgates to new technologies with the capability to transform how society operates [4]. The GPT-3 model, with its 175 billion parameters, has demonstrated strong performance on many tasks [5].

We developed a software publishing platform that imagines a future where authors co-write their manuscripts with the support of large language models. We used, as a base, the Manubot platform for scholarly publishing [6]. Manubot was designed as an end-to-end publishing platform for scholarly writing for both individual and large-collaborative projects. It has been used for collaborations of approximately 50 authors writing hundreds of pages of text reviewing progress during the COVID19 pandemic [7]. We developed a new workflow that parses the manuscript, uses a large language model with section-specific custom prompts to revise the manuscript, and then creates a set of suggested changes to reach the revised state. Changes are presented to the user through the GitHub interface for author review and integration into the published document.

Implementing AI-based revision into the Manubot publishing ecosystem

Overview

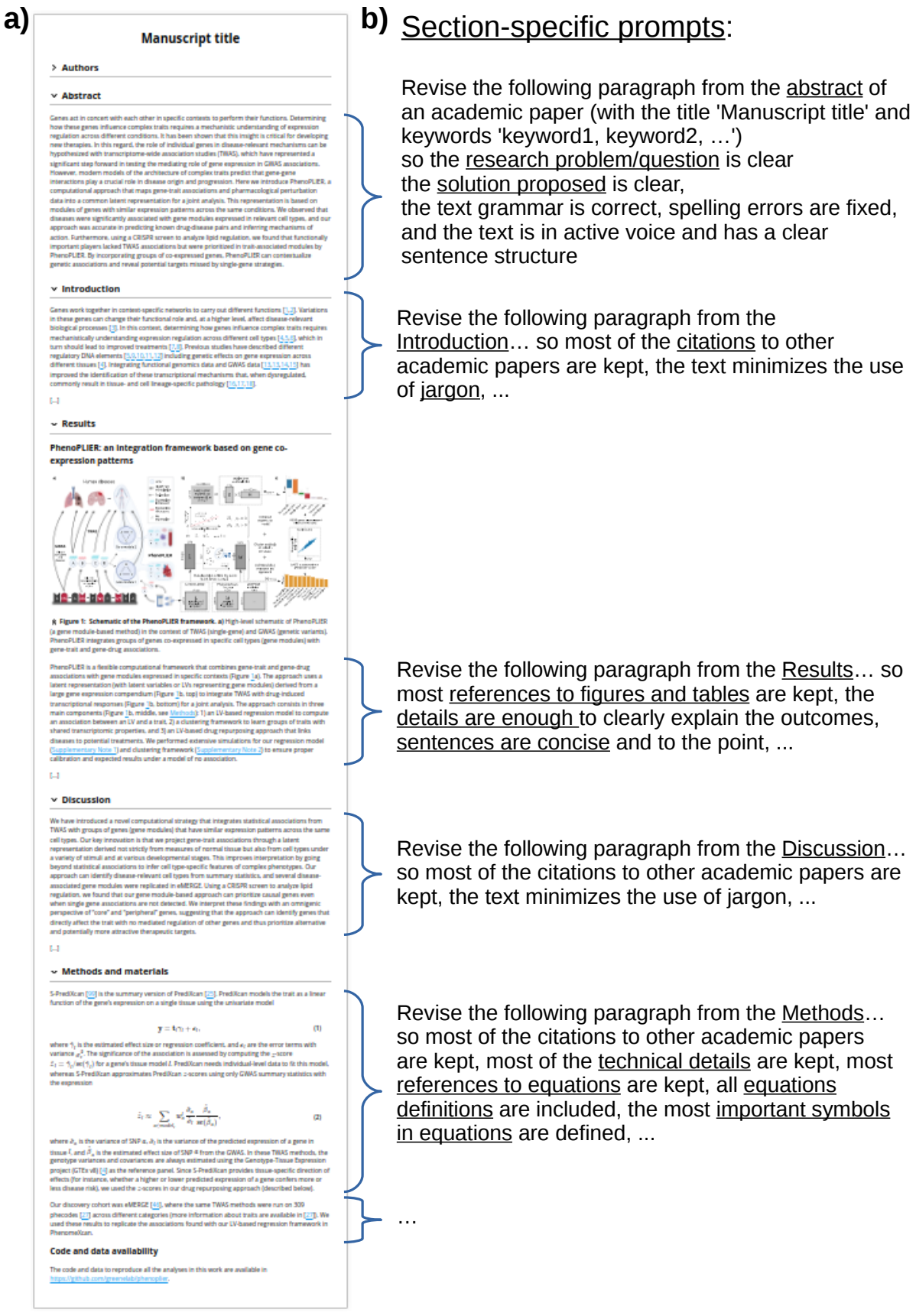

Figure 1: AI-based revision applied on a Manubot-based manuscript. a) A manuscript (written with Manubot) with different sections. **b)** Section-specific prompts used to process each paragraph. If a paragraph belongs to a non-standard section, then a default prompt will be used to perform a basic revision only.

We implemented the AI-based revision infrastructure in Manubot [6]. Manubot is a tool for collaborative writing of scientific manuscripts. It utilizes version control and a continuous integration workflow to facilitate efficient and transparent collaboration among authors. Manubot integrates with popular version control platforms such as GitHub, allowing authors to easily track changes and collaborate on writing in real time. Additionally, Manubot automates the process of generating a

formatted manuscript (such as HTML, PDF, DOCX; Figure 1a shows the HTML output), reducing the time and effort required for manuscript preparation and submission. 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.

When the user triggers the action, the manuscript is parsed by section and then by paragraph (Figure 1b), passed to the language model along with a set of custom prompts, returned, reformatted, and output. Our workflow then uses 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 attributes text to either the human user or to the AI language model, which may be important if future legal decisions alter the copyright landscape around 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 manuscript length, we implemented an workflow in GitHub Actions that can be manually triggered by the user. Our implementation allows users to tune the costs to their needs by allowing to select specific sections to be revised instead of the entire manuscript. Additionally, several model parameters can be adjusted to tune costs even further, such as the language model version (including Davinci and Curie, and potentially newly published ones), how much risk the model will take, or the "quality" of the completions. For instance, using Davinci models (the most complex and capable ones), the cost per run is under $0.50 for most manuscripts.

Implementation details

Our tools are comprised of Python scripts that perform the AI-based revision (https://github.com/greenelab/manubot-ai-editor) and a GitHub Actions workflow that integrates manuscript with Manubot. The user only needs to run the workflow by specifing the branch that will be revised and selecting the files/sections of the manuscript (optional), the language model to use (`text-davinci-003` by default) and the output branch name. As explained later, for more advanced users it is also possible change 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 to change 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. The prompt uses the manuscript title and keywords, so both have to be accurate for getting the best revision outcomes. The other key component to process a paragraph is its section. Some paragraphs are simpler to process than others. 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, where enough details about the experiments must be provided 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. Figures and tables captions, as well as paragraphs that contain only one or two sentences and less than sixty words, are not processed and 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, then the file will not be processed. If this happens, the user is still able to specify which section the file belongs to. The section could 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 will be used to instruct the model to perform only a basic revision (`minimize the use of jargon, ensure text grammar is correct spelling errors are fixed, and the text has a clear sentence structure`).

Properties of language models

Our AI-based revision workflow uses text completion to process each paragraph, either using the completion endpoint or the new edits endpoint (which is currently in beta). We tested our tool using Davinci and Curie models, including `text-davinci-003`, `text-davinci-edit-001` and `text-curie-001`. Davinci models are the most powerful GPT-3 model, whereas Curie ones are less capable but faster and less expensive. Although the edits endpoints would be the ideal interface for our task, it is still in beta. Therefore, we mainly focused on the completion endpoint. All models can be fine-tuned using different parameters [8], 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, and the maximum number of tokens to generate for the completion (parameter `max_tokens`). For instance, the context length of Davinci models is 4,000, and 2,048 for Curie [9]. For this reason, it is still not possible to use the entire manuscript as input, not even entire sections. Therefore, our AI-assisted revision software process each paragraph of the manuscript with section-specific prompts, as shown in Figure 1b. The advantage of this approach is the ability to process large manuscripts by processing small chunks of text. The main issue, however, is that the language model processes only a single paragraph from a section, potentially losing important context to produce a better output. Nonetheless, we find that the model still produces high-quality output (see Results). Additionally, since the goal of our tool is to revise a paragraph, by default we set the maximum number of tokens (parameter `max_tokens`) as twice the estimated number of tokens in the paragraph (one token approximately represents four characters, as indicated in [10]). The tool automatically adjusts this parameter and performs the request again if a related error is returned by the API. The user can 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: they will generate a different revision for the same input paragraph each time. This behavior can be changed by using the "sampling temperature" or "nucleus sampling" parameters (we use `temperature=0.5` by default). Although we selected default values that worked well across multiple manuscripts, these parameters can be changed by the user if necessary to make the model more deterministic. The user can also instruct the model to generate, for each paragraph, several completions and select the one with the highest log probability per token, what can improve the quality of the revision. Our proof-of-concept implementation generates only one completion (parameter `best_of=1`) to avoid potentially high costs for the user. Additionally, our workflow allows to process either the entire manuscript or individual sections. This allows to control costs more effectively while focusing on a single piece of text in which the user can run the tool several times and pick the prefered revised text.

Results

We used this infrastructure to revise an existing manuscript as well as to author a new one. We back-ported the changes in Manubot to a manuscript describing the Clustermatch Correlation Coefficient (CCC) [11]. The CCC was designed to capture both linear and non-linear relationships between variables. The CCC manuscript describes its use, in particular with gene expression data.

The abstract of the CCC manuscript before revision had a Flesh-Kincaid readability score of X and a grade level of Y. > PREVIOUS_VERSION

After suggested revisions, the readability score was X and the grade level was Y and read as follows: > NEW_VERSION

The full manuscript before AI-based revision is available at [link], and the revised version is available at [new_link]. We noticed that the model has difficulty with the Manubot citation style, which may lead to some references becoming incorrect. This pipeline is not fully automated: authors will need to review changes and verify the output.

We also used this framework in the context of authoring a new manuscript that described a publishing infrastructure that implemented large language models to suggest revisions. The abstract before revisions had a Flesh-Kincaid readability score of X and a grade level of Y and read as follows: > Academics often communicate through scholarly manuscripts. > These manuscripts describe new advances, summarize existing literature, or argue for changes in the status quo. > Writing and revising manuscripts can be a time-consuming process. > Large language models are bringing new capabilities to many areas of knowledge work. > We integrated the use of large language models into the Manubot publishing ecosystem. > Users of Manubot can run a workflow, which will trigger a series of queries to OpenAI's language models, produce revisions, and create a timestamped set of suggested revisions. > Given the amount of time that researchers put into crafting prose, we expect this advance to radically transform the type of knowledge work that academics perform.

After suggested revisions, abstract had a Flesh-Kincaid readability score of X and a grade level of Y and read as follows: > NEW_VERSION

Conclusions

We implemented AI-based models into publishing infrastructure. While most manuscripts have been written by humans, the process is time consuming and academic writing can be difficult to parse. We sought to develop a technology that academics could use to make their writing more understandable without changing the fundamental meaning. This work lays the foundation for a future where academic manuscripts are constructed by a process that incorporates both human and machine authors.

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. **Understanding the Capabilities, Limitations, and Societal Impact of Large Language Models**
Alex Tamkin, Miles Brundage, Jack Clark, Deep Ganguli
arXiv (2021-02-05) https://arxiv.org/abs/2102.02503

5. **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

6. **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

7. **An Open-Publishing Response to the COVID-19 Infodemic.**
Halie M Rando, Simina M Boca, Lucy D'Agostino McGowan, Daniel S Himmelstein, Michael P Robson, Vincent Rubinetti, Ryan Velazquez, Casey S Greene, Anthony Gitter
ArXiv (2021-09-17) https://www.ncbi.nlm.nih.gov/pubmed/34545336
PMID: 34545336 · PMCID: PMC8452106

8. **OpenAI API** https://beta.openai.com

9. **OpenAI API** https://beta.openai.com

10. **OpenAI API** https://beta.openai.com

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