Does Microsoft *really* need more Money?

Evaluating Local Large Language Models For Machine-Readable Metadata Curation

Aerith Y. Netzer,

January 1, 2025

Abstract

Here, we show that large language models can be an efficient way to convert plain-text citations to BibTeX for use in machine-actionable metadata. Further, we prove that these models can be run locally, without cloud compute cost. With these tools, university-owned publishing operations can increase their operating efficiency with no effect on quality.

Background and Motivation

University-owned journal-publishing operations operate under far tighter economic constraints—direct and opportunity—and therefore must solve the same problems of corporate academic publishers with a fraction of the resources available.(*The State of U.S. Academic Libraries: Findings from the ACRL 2023 Annual Survey* 2024) (RELX 2023)One of these problems is reference metadata, i.e. machine-actionable references that are then used to count citations of articles. The act of capturing, counting, and using citations accurately allows for funding agencies, universities, and publishers to make data-driven decisions for funding allocation, allows for reviewers to validate the research of a manuscript, and allows for faster literature review.

There have been many projects aimed to converting plain-text to BibTeX using programmatic means, but are often limited to certain languages ("Makino Takaki's Page - Writings - Technical Tips - Generate BiBTeX Entry from Plain Text (.en)," n.d.) or are dependent upon external data ("Text2bib," n.d.). Large language models' ability to generalize instructions and their "understanding" of language, can do a much better job with less human intervention required, with no reference to external data.

An Example

The workflow for our university—a medium-size, elite university in the Mid-west United States—consists of receiving manuscripts from authors in a Microsoft Word file format. We then use pandoc ("Pandoc - Index," n.d.) to transform this Word document to a markdown file format, from which we can build PDF and Web versions from a single source. But due to author unwillingness to use plaintext markup formats such as LaTeX or Markdown, we must recreate the bibliography. Previously, this meant looking up each source, adding them to a Zotero("Zotero | Your Personal Research Assistant," n.d.) library, and then exporting the biblatex file for use as metadata in the web version of the article. This would allow for services such as Google Scholar and Web of Science to scrape the metadata and count citations for the cited articles. This allows for researchers conducting literature reviews to find articles easier and faster, and allows for easier cross-checking for dubious claims. The present system, can automate this labor-intensive machine-actionable metadata creation process. With the advent of Large Language Models (LLMs), we can create systems to parse out the plaintext citations in an article, pass it to a Large Language Model, and output a machine-actionable metadata citation entry.

Limitations and Concerns

This analysis is—by necessity—is limited to works that appear in the crossref API, creating a bias in the dataset against older works and academic monographs. While this limits the usefulness of this analysis to

publishers whose specialty lies within fields where citations are limited to recent works (such as the physical sciences), future work can and should include plain-text citations of historical, non-digital, and non-academic works.

Along with the rapid growth in users of Large-Language models, so have concerns over the ecological sustainability of LLM technology.(Ding and Shi 2024) (Chien et al. 2023) Most of these concerns, however, can be alleviated with the use of "small" models such as those provided by Ollama. Further, there are concerns about the validity of Large-Language models, especially concerning their propensity to hallucinate. However, in combination with validity checkers such as bibtexparser and human review, we are confident enough in this system to be used in production of our journals.("About the Journal - Bulletin of Applied Transgender Studies," n.d.) Future work in this area should include building scalable, verifiable workflows that do not necessitate human oversight.

Methodology

Data Collection

Data was collected via the CrossRef API (Bartell, n.d.). We sampled a DOI and randomly selected a plain-text citation style from the following list:

1. Chicago Author-Date
2. Elsevier-Harvard
3. Ecoscience
4. APA
5. MLA
6. IEEE
7. Council of Science Editors

Using the CrossRef API, we pulled the BibTex Citation, the Plaintext Citation, and the DOI to create a dataset for our analysis. [Table 1] presents the variables and their descriptions.

Table 2: Citation Metadata

Variable	Description
DOI	The Digital Object Identifier of the requisite work.
BibTeX Citation	Metadata about the work in BibTeX format.
Plain Text Citation	The cited work in a given citation style.
Plain Text Citation Style	The style in which the plain text citation is given.

To fairly analyze the performance of each language model on a broad set of citation formats, each variable was randomly assigned one of the above citation styles and returned by the CrossRef content negotiation API endpoint. Using this random assignment of citation formats, we achieved a roughly even distribution of each citation style in the dataset.

Analysis

All language models were tested using the Ollama("Ollama/Ollama" 2025) toolkit using the Quest ("Quest High-Performance Computing Cluster: Information Technology - Northwestern University," n.d.) supercomputing cluster at Northwestern University, running in a singularity container.(Kurtzer, Sochat, and Bauer 2017) Testing of all models took 14 hours to complete on two NVIDIA A100 Graphical Processing Units, one node with eight cores, and 128 gigabytes of memory.[1] [2] All code was written in python using an Anaconda environment to aid in reproducible deployments of this code.

[1]The script used to create the SLURM job can be found in the GitHub repository mentioned in the "Data and Code Availability" section.

[2]Thank you to Kat Nykiel at Purdue University for her assistance in building and deploying the singularity container.

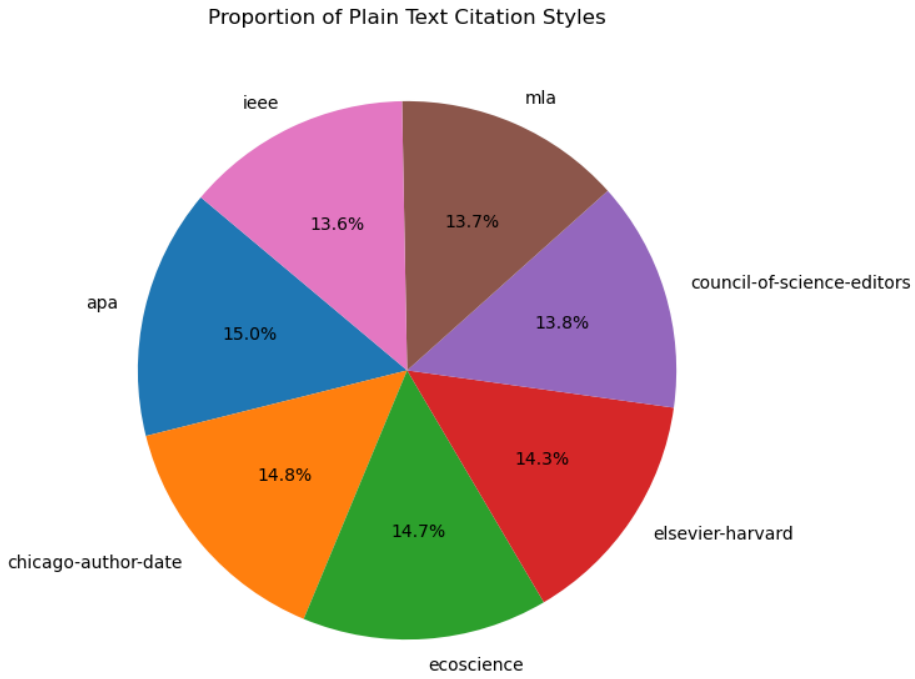

Figure 1: Pie Chart demonstrating the proportion of each citation style present in the dataset

We used the plain text citation given by the CrossRef API as a ground truth to which the model would aspire to. We prompted each model with the same text:

> You are a professional citation parser.
>
> Given the following plain text citation:
>
> {plain_text_citation}
>
> Please convert this citation into a structured BibTeX entry. Include all relevant fields such as author, title, journal, volume, pages, year, etc.
>
> Output only the BibTeX entry, and nothing else. Do not include any explanations, preambles, or additional text.

The following models were prompted:

1. codegemma:2b(CodeGemma Team et al. 2024)
2. codegemma:7b(CodeGemma Team et al. 2024)
3. llama2:7b(Touvron et al. 2023)
4. llama3.3:70b(AI@Meta 2024)
5. llama3:8b(AI@Meta 2024)
6. mistral:7b("Mistral," n.d.)
7. starcoder2:3b(Li et al. 2023)
8. tinyllama(Zhang et al. 2024)

These models were chosen to represent a range of model sizes and training methods.

The following variables were saved to the output file of the model:

Variable	Description
Model	The model being tested. One of the eight models listed above.
PlainTextCitation	Maps to plain text citation field table 1.
TimeToGeneration	Time taken to generate the entry.
ActualBibTeX	BibTeX entry retrieved from Crossref.
TotalFields	The number of BibTeX fields being compared in generated and "ground truth" entries.
Matching Fields	The number of fields that have a match in both the generated and "ground truth" entries.
Percentage Match	$\frac{\text{Total Fields}}{\text{Matching Fields}}$

This generated 8 CSV files of approximately 3,000 lines each. Each row corresponds to a single DOI. These files were used for analyzing the efficiency and effectiveness of each model.

Results

Model Effectiveness

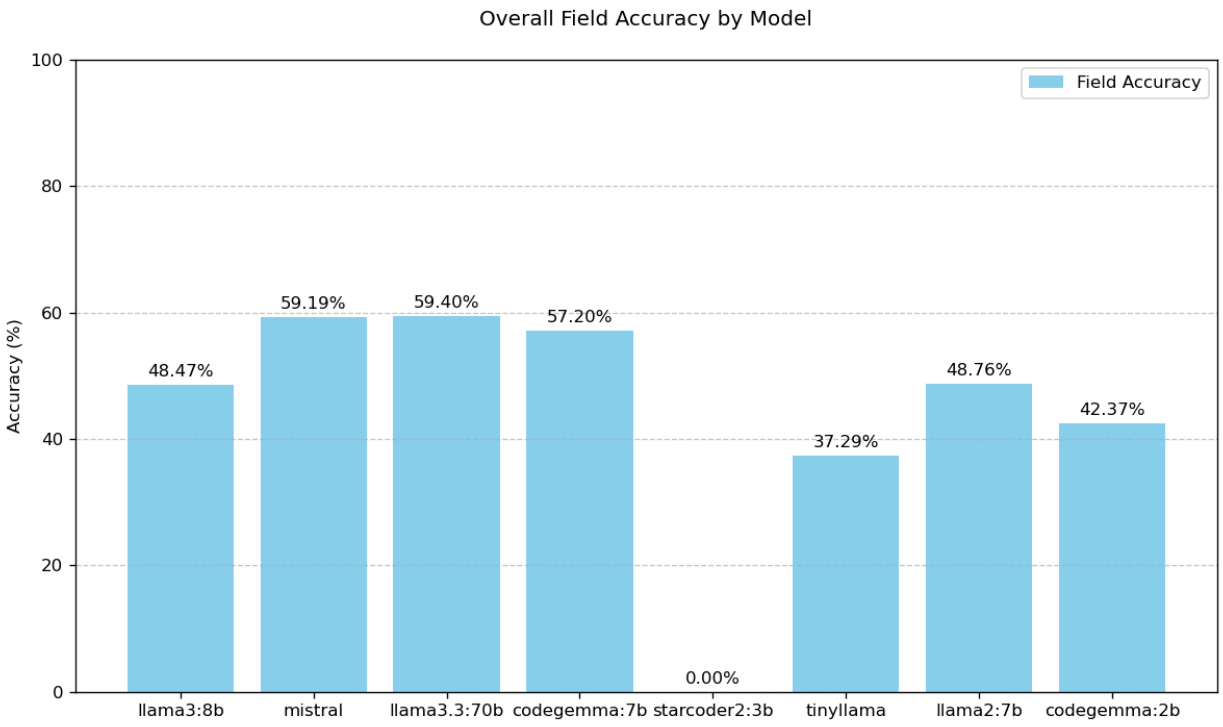

Figure 2: Per Field Accuracy and Valid BibTeX of the Model

Unsurprisingly, llama3.3:70b, the most advanced and largest model of the chosen models, performed the best. Further, starcoder2:3b failed to create any valid BibTeX entries, whereas every other model created valid BibTeX for every citation.

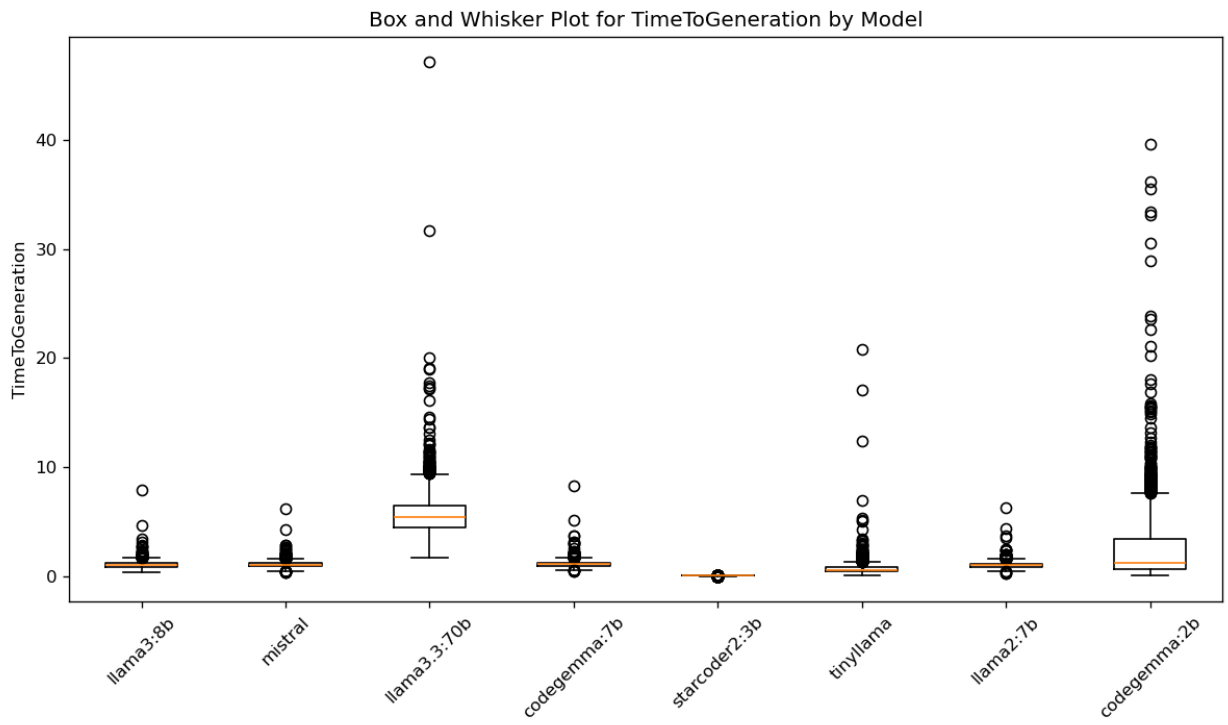

Figure 3: Model time to Generation

Model	Median Time to Generation (seconds)	Standard Deviation (seconds)
codegemma:2b	1.18	3.24
codegemma:7b	1.08	0.29
llama2:7b	0.96	0.27
llama3.3:70b	5.45	1.90
llama3:8b	1.00	0.31
mistral	1.00	0.27
starcoder2:3b	0.00	0.00
tinyllama	0.57	0.64

As every model except for starcoder2:3b created valid BibTeX perfectly, we are primarily concerned with and the accuracy of the fields. In this discussion, the *validity* of the BibTeX simply means that if the BibTeX can be parsed without errors, then the BibTeX is valid. However, a well-formed BibTeX entry can be *valid* but *incorrect*. Meaning that the entry can be parsed, but the data in the entry is wrong. But llama3.3:70b generated the most *accurate* BibTeX entries. However, we should not take this that the model was necessarily *incorrect*, but was just different from how Crossref represented the field. Mistral and Codegemma, though, are very close behind, especially with their parameter sizes (and therefore cost of compute) much lower than llama3.3:70b, it may be economical for some publishing operations to use smaller models, decreasing their cost, while keeping parity with the accuracy of the model. Trading a .2% reduction in accuracy for, on average, a 5x faster computation is an effective strategy for this use case.

Per-Field Accuracy

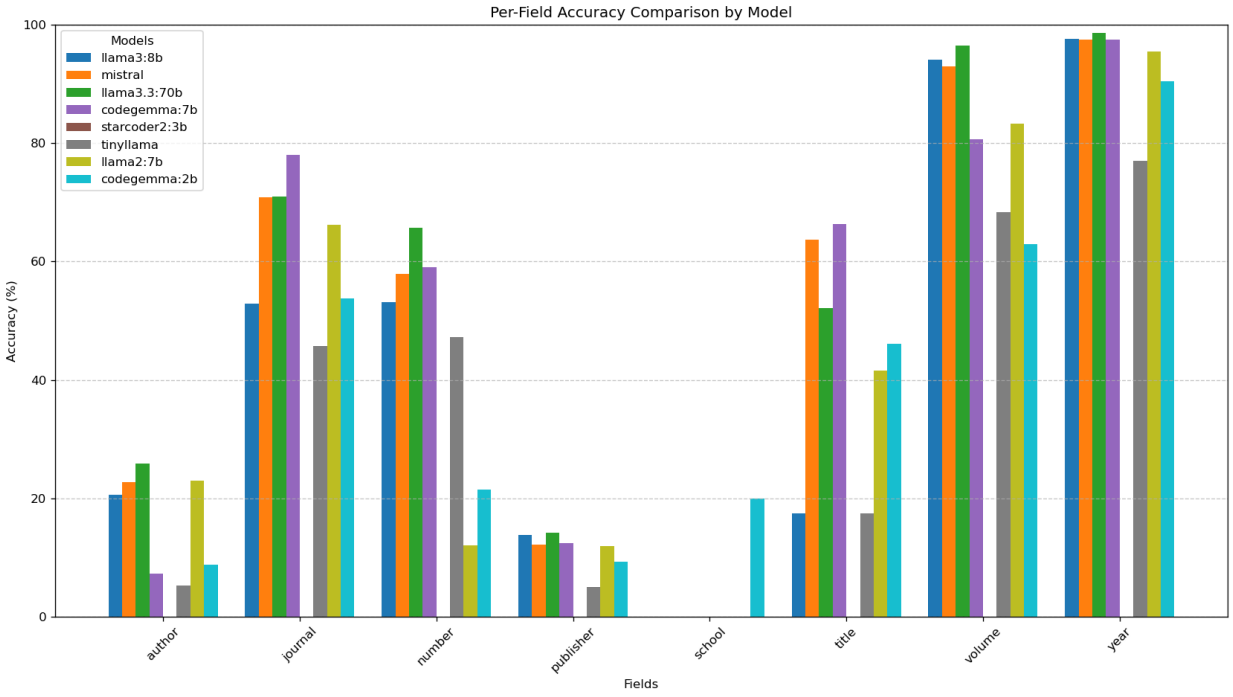

Figure 4: Per-field accuracy by model

All models were very accurate in producing volume, year, and journal entries in BibTeX, while author, publisher, and school were the least accurate fields. This is because there is greater freedom and flexibility in how these fields are entered, and thus a correct and valid generated BibTeX need not be exactly the same as Crossref's representation of the same data. Future work should include creating a validator to identify equivalent author, publisher, and school names.

For example, consider the following BibTeX entries:

> {National Academy of Sciences, The}
>
> {The National Academy of Sciences}

While these entries refer to the same entity, they cannot be identified as the same programmatically, and are thus penalized as "inaccurate." Thus, the results in Figure 2 should be interpreted as the models' accuracy when using Crossref as the metric of accuracy. This analysis is useful because it shows which models are better-suited for this task, rather than the concluding 50% of the fields to be incorrect.

Conclusion

For university-owned publishers, small, locally-available LLMs are capable of producing well-formed BibTeX. These models can be used to create machine-actionable citation metadata, automating a step in the publishing process. As of the publication of this paper, 7 billion parameter models, especially mistral, are capable of running on the latest laptops, and provide good performance at the least cost.

Data and Code Availability

The author strives to adhere to the FAIR guiding principles. Code and data used for this analysis is available on GitHub.("Aerithnetzer/Biblatex-Transformer," n.d.)

Bibliography

"About the Journal - Bulletin of Applied Transgender Studies." n.d. https://bulletin.appliedtransstudies.org/about/. Accessed March 8, 2024.

"Aerithnetzer/Biblatex-Transformer." n.d. https://github.com/aerithnetzer/biblatex-transformer. Accessed January 1, 2025.

AI@Meta. 2024. "Llama 3 Model Card."

Bartell, Amanda. n.d. "REST API." Website. *Crossref*. https://www.crossref.org/documentation/retrieve-metadata/rest-api/. Accessed December 31, 2024.

Chien, Andrew A, Liuzixuan Lin, Hai Nguyen, Varsha Rao, Tristan Sharma, and Rajini Wijayawardana. 2023. "Reducing the Carbon Impact of Generative AI Inference (Today and in 2035)." In *Proceedings of the 2nd Workshop on Sustainable Computer Systems*, 1–7. Boston MA USA: ACM. https://doi.org/10.1145/3604930.3605705.

CodeGemma Team, Ale Jakse Hartman, Andrea Hu, Christopher A. Choquette-Choo, Heri Zhao, Jane Fine, Jeffrey Hui, et al. 2024. "CodeGemma: Open Code Models Based on Gemma."

Ding, Yi, and Tianyao Shi. 2024. "Sustainable LLM Serving: Environmental Implications, Challenges, and Opportunities : Invited Paper." In *2024 IEEE 15th International Green and Sustainable Computing Conference (IGSC)*, 37–38. https://doi.org/10.1109/IGSC64514.2024.00016.

Kurtzer, Gregory M., Vanessa Sochat, and Michael W. Bauer. 2017. "Singularity: Scientific Containers for Mobility of Compute." *PLOS ONE* 12 (5): e0177459. https://doi.org/10.1371/journal.pone.0177459.

Li, Raymond, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, et al. 2023. "StarCoder: May the Source Be with You!" https://arxiv.org/abs/2305.06161.

"Makino Takaki's Page - Writings - Technical Tips - Generate BiBTeX Entry from Plain Text (.en)." n.d. https://www.snowelm.com/~t/doc/tips/makebib.en.html. Accessed March 9, 2024.

"Mistral." n.d. https://ollama.com/mistral. Accessed January 1, 2025.

"Ollama/Ollama." 2025. Ollama.

"Pandoc - Index." n.d. https://pandoc.org/. Accessed March 8, 2024.

"Quest High-Performance Computing Cluster: Information Technology - Northwestern University." n.d. https://www.it.northwestern.edu/departments/it-services-support/research/computing/quest/. Accessed January 1, 2025.

RELX. 2023. "Market Segments." RELX.

"Text2bib." n.d. https://text2bib.economics.utoronto.ca/index.php/index. Accessed March 9, 2024.

The State of U.S. Academic Libraries: Findings from the ACRL 2023 Annual Survey. 2024. Chicago: Association of College & Research Libraries.

Touvron, Hugo, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, et al. 2023. "Llama 2: Open Foundation and Fine-Tuned Chat Models." arXiv. https://doi.org/10.48550/arXiv.2307.09288.

Zhang, Peiyuan, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. "TinyLlama: An Open-Source Small Language Model." https://arxiv.org/abs/2401.02385.

"Zotero | Your Personal Research Assistant." n.d. https://www.zotero.org/. Accessed December 31, 2024.