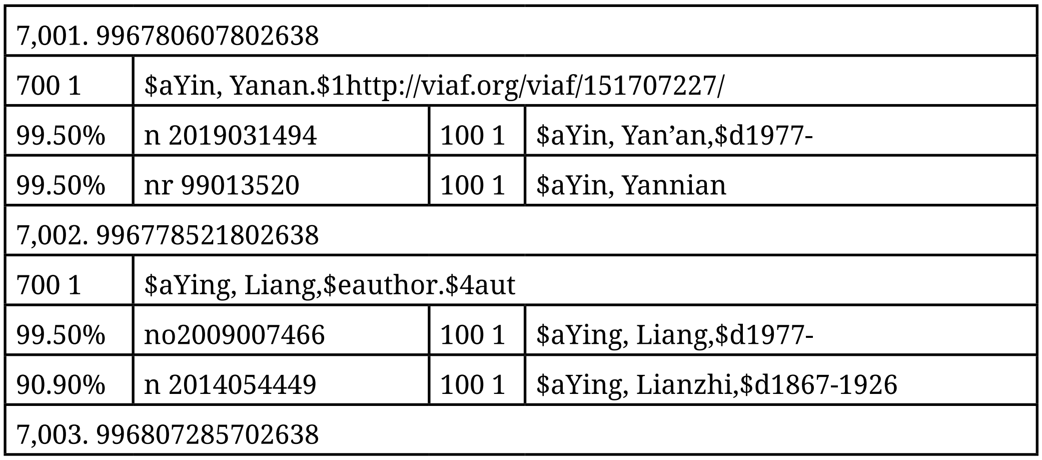
The Little Bot That Couldn't

Reflections on Effort and Insight

By Leslie A. Engelson

In 2013, our library contracted with Backstage Library Works to process all of our bibliographic records for authority control. That process resulted in reports of unmatched, partially matched, and near matches of hundreds of thousands of access points (AP). We have continued sending new bibliographic records to Backstage on a quarterly basis and receive new reports based on those records. I decided to focus on resolving the near matches first. To facilitate searching for the APs and reduce redundancy, I asked student workers to merge the main list generated from the initial record processing with the lists received every quarter with the processing of new bibliographic records.

For ease of use, I convert the reports to Excel spreadsheets and upload them into a shared Google Workspace. Each entry consists of four rows: Row 1 contains the bibliographic records number associated with the AP, row 2 contains the AP along with the field number and indicators, and rows 3 and 4 include suggestions for matches along with the percentage of how close the match might be. [See Figure 1] All of this information is helpful for the staff working to resolve these APs. Because each entry comprises four rows, it is not a simple matter of adding the new list to the bottom of the spreadsheet and sorting alphabetically by the AP. The students have to find the correct alphabetical place in the main spreadsheet to insert the four rows from the new spreadsheet for each AP, eliminating duplicates along the way. The main list has been divided across 27 spreadsheets to simplify the task; however, it is still tedious and prone to errors.



7,001. 996780607802638			
700 1	$aYin, Yanan.$1http://viaf.org/viaf/151707227/		
99.50%	n 2019031494	100 1	$aYin, Yan'an,$d1977-
99.50%	nr 99013520	100 1	$aYin, Yannian
7,002. 996778521802638			
700 1	$aYing, Liang,$eauthor.$4aut		
99.50%	no2009007466	100 1	$aYing, Liang,$d1977-
90.90%	n 2014054449	100 1	$aYing, Lianzhi,$d1867-1926
7,003. 996807285702638			

Figure 1: Snippet of spreadsheet demonstrating the structure of the data

I am always on the hunt for ways to make the workflow in the cataloging department more efficient and accurate. We often have tasks that benefit from automation due to their repetitious and routine nature. I am not a programmer so when AI tools became readily available, I was hoping the task of de-duplicating and merging spreadsheets could finally be automated utilizing these tools, and so, I started to explore.

Leslie A. Engelson is Metadata Librarian, Murray State University.

Part of that exploration included consulting with AI guru, Michael Hanegan. Co-author of *Generative AI and Libraries*, Michael has a comprehensive understanding of the capabilities of Large Language Models (LLMs) and how they can benefit and enhance the work of librarians. In discussing what the project entailed, Michael clarified that LLMs work by breaking up text into "tokens". These tokens don't necessarily correspond with words. The structure of the data in the spreadsheet (the four-row entry) provides further challenges. After testing one of the LLMs on this project with a less-than satisfactory result, Michael provided some options for me to investigate, including alternative LLMs.

Due to time constraints and the challenges of delving further, the project sat on the back burner until I signed up for the Designing Custom AI Assistants for Higher Education Learning Lab through EDUCAUSE. This was a series of workshops across four days in July that utilized live sessions, online discussions, and hands-on activities to walk me through the process of creating an AI assistant specifically designed to do this task of de-duplicating and merging spreadsheets. The course facilitators, Tim Lindgren, Kyle Fidalgo, and Claire Angus, provided one-on-one feedback through every step of the process. The course also employed AI assistants specifically designed to facilitate the process of developing my own AI assistant.

While I already understood the importance of well-crafted prompts for generating the results I need, in this course I learned that saving those prompts in a shell assistant creates a dedicated LLM or "generative pre-trained transformer" (GPT) focused on solely and repeatedly performing specific processes. This functionality is available in any of the paid versions of current LLMs. However, for this course, students were provided access to Playlab (https://www.playlab.ai/), an AI platform designed to help educators develop AI tools for their classrooms and communities. In Playlab, it was easy to create the shell for my assistant which I named Deduplicator. The next step was to define the parameters and develop prompts.

After creating the shell, I used one of the course assistants called IDA (Idea Discovery Assistant) to discuss the idea I had for what I wanted Deduplicator to do. This assistant is prompted to ask questions and guide the discussion to help the user determine the scope and end results generated for the assistant they are creating. Through discussion with IDA, I realized that it would be better to simplify the task and have the assistant focus only on deduplicating the two spreadsheets. This would allow for less complicated prompts, easier troubleshooting, and would minimize processing time. IDA's questions helped me think through my goals, clarifying the key functionality of the assistant as the process of the interaction of the user - in this case student workers - with the Deduplicator assistant, along with determining the scope and constraints of the task, and the criteria that will determine the success of its efforts.

This process also adjusted my understanding of how LLMs can be approached. Instead of typing a question into Claude or ChatGPT, I now ask the LLM to interview me about a project or task to help me think through it. For instance, I recently asked ChatGPT to interview me about the pros and cons of implementing shelf-ready cataloging and processing. It responded with open-ended questions that would help someone considering this service for their library to think through not just the pros and cons, but also the risks and implications of shelf-ready cataloging and processing, including vendor considerations, staff and workflow impact, and much more.

After determining the scope, interaction process, and success criteria for Deduplicator, I next used CHAD (Conversational Helper for Assistant Design) to help me create the prompts for Deduplicator that will help guide the user (a student worker) through the process of deduplicating two spreadsheets. Since I had already thought through the process with the help of IDA, developing the prompts was straightforward. Again, CHAD interviewed me and guided my development of the prompts.

My first attempt was a great success. I chose to upload two small spreadsheets in order to simplify the process, minimize processing time, and allow for easier troubleshooting. My next effort, however, resulted in Deduplicator freezing. It is possible that the assistant was hanging up because of caching issues, improperly reset memory, or challenges stemming from the size of the spreadsheets and of temporary file storage. None of these bode well for enabling student workers to independently perform this process repeatedly on different spreadsheets in the same session. In my discussion post on this experience in the EDUCAUSE Learning Lab workshop, the facilitator provided me with some troubleshooting steps but, ultimately indicated that "…the task you are pursuing isn't the best fit for a language-based chatbot" (C. Angus).

I learned during this process that I can use an LLM for troubleshooting. The facilitator asked an LLM if my task was manageable by a custom LLM assistant or if I would be better off using a different technology, platform, or approach. In addition to stating the challenges of my task for an LLM to facilitate (token constraints), it also provided alternatives such as dedicated deduplication tools or programming such as Python.

While I was disappointed in the ultimate failure of Deduplicator, I did learn a lot about how LLMs work and how they can be utilized better. I also gained some insights that put me in a better position to utilize LLMs more effectively with expectations better aligned with their capabilities.

Some of these insights include:

- The supportive nature of the IDA assistant was nice; however, it would have been more helpful if that assistant was more realistic instead of idealistic, letting me know what the pitfalls and challenges might be for the task I was designing my assistant for. LLMs are notoriously optimistic and affirming.
- I need to be even more literal in my communication with an LLM than I usually am. People can infer better than machines and, while I always try to be clear and literal, especially when training, working with an AI assistant has made me more aware of where gaps in information might lead to an incorrect result.
- Working with an AI assistant, with its predictive and discursive nature, is more like an interactive discussion than inputting programming prompts.

I'm still on the lookout for a way to make this tedious task more efficient and accurate. I have further options to explore, so it is possible that AI might have a role to play in those.