Streamlining Metadata Creation: Implementing and Assessing AI Workflows to Improve Discoverability

James Mason[1], Kyla Jemison[1,*]

[1] *University of Toronto, Toronto, Ontario, Canada*

Abstract

Anthologies of art song have often posed challenges to discovery as contents notes are not always adequately transcribed, making it difficult for users to know what songs are contained in each score. Transcribing contents notes can be difficult, especially when the songs are in multiple languages. Through a practical and real-world example, this paper demonstrates the application of automation and artificial intelligence to enhance cataloguing records with improved contents notes and evaluates the results through a user-centred lens. We highlight possibilities for this evolving technology as well as the challenges that it can pose and explore the concept of a cost-benefit analysis of metadata work with the element of artificial intelligence being considered in a holistic manner.

Keywords

Cataloguing, AI, discovery, evaluation, music

1. Introduction

The University of Toronto Music Library holds the largest and most comprehensive music research collection in Canada, and as such has an extensive collection of compilations and anthologies. Many of these volumes were acquired and catalogued at times when the library did not have the resources to provide full catalogue records that included contents notes providing title and composer information for the works contained within. Frequently these contents notes were not provided or indexed in the card catalogue system, and thus they were not digitized when the library moved to an online catalogue in 1976. These collections represent an important way of exploring repertoire as they bring together works for the same instrumentation or voice, often at a particular skill level or other similar quality, but their contents are difficult to discover.

With the prevalence of keyword searching, thorough contents notes become even more important. Dougan's 2018 research indicates that music students prefer to keyword search: that is, they are not very interested in using advanced search options and instead prefer to search using keywords like "bach english suite", combining composer names and work titles in one keyword search [1]. While today's comprehensive description practices ensure that those kinds of keywords always exist in a record for a single work, we have contemporary and historical compilations that do not include complete contents notes.

Without adequate descriptions, users cannot discover compilations that contain the works

DCMI-2025 International Conference on Dublin Core and Metadata Applications

*Corresponding author.
j.mason@utoronto.ca (J. Mason); kyla.jemison@utoronto.ca (K. Jemison)
0009-0000-8143-3710 (J. Mason); 0000-0002-8795-1372 (K. Jemison)

https://doi.org/10.23106/dcmi.952526168

they are interested in. Newcomer [2] writes that one of the challenges of music discovery in library systems is “When the work sought is part of a compilation, users may not recognize the compilation title (usually displayed prominently and weighted heavily in indexing) as something that contains the work they seek.” Belford notes that “formatted contents notes (MARC 505) contain useful, detailed, and eye-readable information about titles, composers, performers, and other contributors” [3] that can help overcome this challenge. The Music Library Association’s Music discovery requirements: A guide to optimizing interfaces notes that “for most comprehensive coverage [of compilations], these transcribed titles should also be included in title keyword indexes and displayed” [4]. While that certainly would be ideal, even unformatted titles and composers for the works would have a significant impact on their discoverability.

The University of Toronto Libraries has a growing interest in leveraging AI tools to improve our workflows and output. Under this mandate, our tri-campus Cataloguing and Metadata Committee formed a working group to test out projects that we suspected could make use of AI effectively and to try out various AI tools to achieve these goals. This working group formed smaller project groups to focus on specific problems. To support our Music Library, we investigated developing a workflow that could create descriptions for the contents of compilations, looking primarily at vocal works. Our goal was to explore AI tools to determine how they could support the work we are already doing in cataloguing these materials, providing assistance for the time-consuming task of metadata creation for tables of contents.

Looking only at materials held in the M16195 class, a class of vocal compilations, there are 481 titles in our collection. Of these volumes, 288 (59%) did not have notes providing title and creator details. This includes both recent titles and older materials in our collections. 168 of the 288 (58%) titles without contents notes are held at Downsview, our off-site storage facility. As this location is not open to users, these titles were not browsable and thus users had no way to learn what they contained before requesting them.

2. Preliminary Testing

We needed to explore the options for scanning pages, Optical Character Recognition (OCR) for images, formatting text, implement file naming practices, all with a low-intervention process.

For scanning we settled on using an in-house scanner capable of black and white scans with a resolution of 300 dpi saved in a lossless TIFF format.

For the OCR we wanted as accessible an option as possible, while utilizing current AI-reliant methods. Furthermore, we needed an option that was able to work in a multilingual context. We settled on EasyOCR. It is an AI-dependent, open-source, and free software, which integrates seamlessly into a Python environment.

For the AI formatting tools we settled on ChatGPT. This technology is progressing so rapidly at this time that a comprehensive selection of tools to test wasn’t feasible. However, to help contextualize the growth and variability of AI options we worked with two versions of ChatGPT, 4 and 3.5 Turbo.

We worked in a Python environment as it allowed us to get as close to a closed workflow as possible, from file reading, OCRing, generating formatted table of contents with the ChatGPT

https://doi.org/10.23106/dcmi.952526168

API, to outputting the results into a format that could merge with records in Alma (our LSP).

3. Methodology

After identifying a sample set, we scanned the table of contents pages and stored the resulting greyscale TIFF images with the file for each image named after the item barcode.

Next, the Python script we wrote fetched the files and performed the OCR process on them. Then, the script passed the OCRed text to the LLM with this prompt that asked it to format the text like a contents note – Title / Composer – Next Title / Next Composer, with some variations on that depending on if there was more than one work by a composer in a row, or if the song came from a larger work, or if there was a translation given. In the prompt we provided both instructions and examples of how the formatted text should look for all these variations. We also asked it to ignore pagination. We developed this prompt by trial and error and the resulting prompt represents what got us our most complete and accurate output from the AI models used. The final step in the instructions asked it to write the results to a .csv file with the item barcode in one column and the LLM results in another. We performed these steps for both ChatGPT 4 and 3.5.

3.1. Human metrics

We devised a three-point Likert scale for our three criteria to allow us to objectively review each result from a cataloguer's perspective. Each result received a score of 1-3, rating the transcription, formatting, and completeness as Poor, Adequate, or Very Good.

Qualitative assessment by cataloguers was also done for each result and will be discussed later in this paper.

3.2. Automated metrics

To allow for testing, a human, trained in metadata practices, also created accurate contents notes for the sample set. We then evaluated the results of the process using five automated metrics. Each evaluates text slightly differently:

BLEU looks at an n-gram comparison of the reference text to the LLM-generated output, looking for an exact match (precision).

METEOR incorporates both precision and recall, measuring how much of the reference text appears anywhere in the generated text, and includes synonym matching, stemming, and word order.

ROUGE focuses primarily on recall, with ROUGE-1 measuring basic word overlap; ROUGE-2 capturing more contextual similarity, and ROUGE-L measuring the longest common subsequence between the reference and the generated text.

These metrics were used to analyse and visualize the results of the automated output by giving them a score [5].

https://doi.org/10.23106/dcmi.952526168

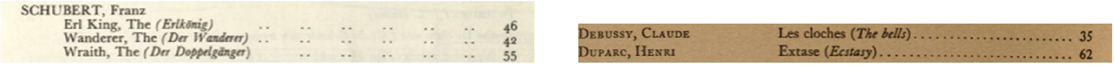
4. Analysis

4.1. Quantitative Analysis

We begin by looking at the results from the automated metrics for the output from the ChatGPT 4 model. We normalized the correction time for a cataloguer to edit the output from the LLM, the resulting "Fix" Time (seconds)—which represents the actual time spent correcting each paragraph—was divided by the word count of that paragraph. This calculation determines the average time per word in seconds, providing a standardized measure that allows for fair comparisons across paragraphs of varying lengths. There is a fair range in the time that it took to edit the output. Our goal was to improve the LLM-generated texts to meet standard cataloguing practice. Somewhat of an average between 2 to 3 seconds per word emerged; roughly 10 of the 17 titles fall into this range.

The relationship between scores and time to correct varies. Item 7764, for example, scored very high, with a small difference in automated metrics, while the time to correct was quite low. This represents the ideal situation. Yet, a predictable pattern does not emerge. Item 1502, for example, scores relatively low with the automated metrics, yet the time to edit the record was similar to the time required to edit Item 0151 which scored much higher. Examining these three items, it is a difficult to understand why. In both Item 1502 and Item 7764 we see a multilingual text with translations formatted inside parentheses.

In Item 7764, the contents notes are represented twice on the same page, once sorted by title and once by composer. ChatGPT 4 had little problem with this and performed excellently. Looking at Item 1502, it received a mediocre score, but took on the lower end of average in time to correct. Like Item 7764 it is a multilingual document with translations (though there is substantially more English than any other language). Much of the time spent correcting Item 1502 was sorting out the placement of the article and the order of the composers' names. Missing words was also a factor in the editing process.



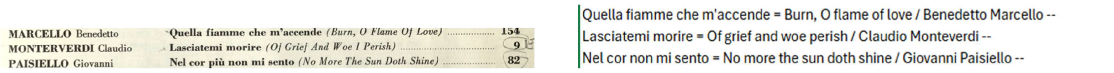
SCHUBERT, Franz

Erl King, The *(Erlkönig)*	46
Wanderer, The *(Der Wanderer)*	42
Wraith, The *(Der Doppelgänger)*	55

DEBUSSY, CLAUDE	Les cloches (*The bells*)........................	35
DUPARC, HENRI	Extase (*Ecstasy*)...........................	62

Figure 1: Examples of ChatGPT 4 output for Items 7764 and 1502, showing formatting of multilingual contents notes.

When we look at Item 0151, we see missing words and one typographical error.



MARCELLO Benedetto	Quella fiamme che m'accende *(Burn, O Flame Of Love)*	154
MONTERVERDI Claudio	Lasciatemi morire *(Of Grief And Woe I Perish)*	9
PAISIELLO Giovanni	Nel cor più non mi sento *(No More The Sun Doth Shine)*	82

Quella fiamme che m'accende = Burn, O flame of love / Benedetto Marcello --
Lasciatemi morire = Of grief and woe perish / Claudio Monteverdi --
Nel cor non mi sento = No more the sun doth shine / Giovanni Paisiello --

Figure 2: Comparison of source text and ChatGPT 4 output for Item 0151, highlighting missing words and typographical errors.

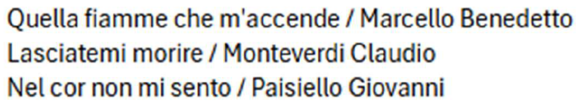
Item 0151 scored much better with the automated metrics, however. This is likely the result of the weighting of importance with the metrics and how they evaluate the text; with metrics considering sematic elements and not simply n-gram overlap, the item scored significantly higher. This suggests to the authors that automated metrics may not be able to serve the purposes of a project like this effectively enough to be able to rely upon without human oversight.

https://doi.org/10.23106/dcmi.952526168

Comparing both the ChatGPT 4 and 3.5 results allows us to understand the some of the difference between the two models.

On the low scores from the various automated metrics, the results from ChatGPT 4 model were generally higher than the low scores for the ChatGPT 3.5 Turbo model. The total scores for ChatGPT 4 were better than those for ChatGPT 3.5 Turbo 35% of the time, they scored the same 42% of the time, and worse 25.5% of the time. No clear pattern seems to be apparent, but ChatGPT 4 has a slight edge over ChatGPT 3.5 based on the automated metrics.

For Item 0151, ChatGPT 3.5 included errors in punctuation, missing words, and formatting.



Quella fiamme che m'accende / Marcello Benedetto
Lasciatemi morire / Monteverdi Claudio
Nel cor non mi sento / Paisiello Giovanni

Figure 3: ChatGPT 3.5 output for Item 0151, showing errors in punctuation, missing words, and formatting compared to ChatGPT 4.

On the low end of the automated metrics for this example (0151), the ChatGPT models were far off and the ChatGPT 4 version scored much better, but looking at the high end of the metrics they were not far off. To the human eye however, ChatGPT 4 clearly performed much better. While one of the automated metrics seems to have captured this, many did not, and without a predictable pattern.

Looking at Item 8364, we see an agreement regarding a large split between the low score and the high score. There were two outliers in this case. BLEU scored it poorly. This is likely as it focuses on a uni-gram comparison between the output and the reference text, not considering any semantic nuances. The highest score, which was also an outlier, came from the ROUGE-L metric which measures the longest strings of n-grams.

Exploring the time it took to edit the results of both models, ChatGPT 3.5 took longer to edit 24% of the time, while 59% of the time it was quicker. Although ChatGPT 4 regularly scored higher with the cataloguers, a qualitative look at this correlation suggests that ChatGPT 4 errors generally have less of an impact on discoverability even though fixing them manually took longer.

A comparison of the cataloguer scores with the automated metrics (both averaged) shows a fair range of discrepancy. Cataloguer scores were normalized to better compare with the automated scores. The scores were normalized using the formula:

$$\text{Normalized Value} = \frac{\text{Value} - \text{Min}}{\text{Max} - \text{Min}}$$

This formula is commonly used in data normalization to make values from different ranges comparable. It rescales the values so that they range from 0 to 1 and are comparable to the scale used by the automated metric. We see a fair bit of difference between the two, and a clear pattern doesn't emerge. The discrepancy between cataloguer scores and the automated scores is larger with the ChatGPT 3.5 model results.

Focussing on the ChatGPT 4 results, we can explore the various automated metrics, and the cataloguer scores presented, as a linear regression. This linear regression allows for predictability between metrics to present itself. If one metric says "this", then we can confidently expect or predict the other metric would say "that". Looking at results from linear regression analysis shows a fair amount of variation, with the line of regression being about .6 which demonstrates a low to moderate relationship. Furthermore, that moderate relationship isn't clearly predictable. Sometimes variation is in excess and other times the opposite. This suggests it may be difficult to find a "good enough" measurement from these scores that could predict what a cataloguer may think and therefore allow us to avoid human oversight in some aspects of the workflow.

Looking more closely into the linear regressions, the Pearson Correlation Coefficient measures the strength and direction of a linear relationship between two numerical variables. It ranges from -1 to 1, where +1 indicates a perfect positive correlation, -1 indicates a perfect negative correlation, and 0 means no linear relationship.

We used this metric to quantify and visualize the relationship between normalized cataloguer scores and automated metrics. The average (mean) of the data was 0.24884129714027803, which shows a fairly weak correlation between the cataloguer scores and the automated metrics (1, or −1 would show the strongest coloration). Looking at the linear regression for the normalized cataloguer scores and automated metrics scores shows a coefficient slope of 0.15390261. The coefficient average suggests there is a positive relationship, the cataloguer scores go up, then the automated metrics scores tend to as well. However, the relationship is small, and likely not enough to allow for meaningful predictability. The coefficient slope shows that the relationship is there, but the data scatters so it does not form a meaningful line.

An analysis of the scores given by each of the automated metrics as well as the cataloguer's score shows chaotic and unpredictable results and does not indicate a metric that could help us evaluate generated text for this project.

4.2. Qualitative Analysis

During the Likert scale analysis, we reviewed each result and made notes on the errors and issues each result displayed. This review highlighted some common issues that had been flagged across multiple records and allowed us to evaluate their significance.

Many of the issues were related to formatting and including all relevant information. For ChatGPT 3.5 in particular, when a composer's name was given as Last Name, First Name, it would usually appear in the results as Last Name First Name. This model also struggled with translations, frequently omitting them, and did not include any of the larger works the songs were part of (operas, musicals, song cycles, etc.). It also omitted the names of any lyricists given for two out of three items. ChatGPT 4 behaved similarly for larger works and lyricists. It performed better but not perfectly regarding translations and composers' names, though it often formatted translations in parentheses, instead of following an equal sign.

There were several common typographical issues between the two models as well. Both omitted the word "più" when it appeared in song titles. They both struggled with the word O in song titles, sometimes omitting it but more often replacing it with a 0 (zero). Both models occasionally omitted other words as well – more in the ChatGPT 3.5 results than in the ChatGPT 4 results. In some results we also noticed hallucinations, where the model has made up information that was not given in the source materials. One particularly bad hallucination was due to illegibility of the printed source material, but other hallucinations did appear in the results for cleanly scanned sources.

https://doi.org/10.23106/dcmi.952526168

In the quantitative analysis we noted that the cataloguers generally ranked the ChatGPT 4 results higher. Looking at this from a qualitative perspective, we can see that generally the newer model made fewer errors that would impede discoverability – it was not great at grouping works by the same composer together or formatting translations, for example – but it did well at including the full text of almost all the songs correctly, along with the composer's surname at least. This keeps the relevant information that someone might be searching for in close proximity, thus making it quite likely that a keyword search for title and composer will return the compilation as a strong match for the song.

5. Conclusion

For a trained library technician, transcribing the contents notes took a little under 4 hours. For that same technician to correct the results from ChatGPT 4 took about 2 ¾ hours.

Several of the issues flagged in our qualitative analysis could easily be remediated on a large scale. Predictable patterns of errors, especially around formatting and mistaken letters, could be fixed with a find and replace, for example. Other problems could be addressed by adapting our prompt to encourage it to pay extra attention to certain words or include more examples of where to find the larger work information in a table of contents. Providing mass fixes for these repeated issues would shorten the correction times required, thus improving this workflow.

The qualitative analysis also demonstrated that we could likely accept results that aid discoverability yet don't conform with standard practice. When composers' names are formatted out of order, it poses almost no problem for keyword searching as most composers are primarily referred to by their surname alone. While having multiple songs by one composer listed individually so the composer's name shows up many times is not ideal from a cataloguer's perspective, it does not impact the discoverability of those songs. In an ideal record, if the information is available, we would like to include the lyricist or poet as a creator of the work, but most users are not searching for works by the lyricist. We would also likely have to set standards for the materials scanned to be used in a process like this; when the text is too difficult for even a human to read, the AI models perform particularly badly, so it is necessary to ensure that the original materials are legible.

In our research, automated metrics were not able to be relied upon to meaningfully predict the quality level of output, therefore human oversight would be necessary even if perfection was not the goal.

Early on in this process we dropped our Cyrillic-text examples as they did not achieve good results with this process. The OCR process did not perform well when combined with multiple Latin alphabet languages but performed much better when it was just Cyrillic and English. For future work, we are interested in using this process on Cyrillic materials because they often contain long tables of contents and as many music cataloguers are not fluent in Cyrillic languages, they take a very long time to type and transliterate. However, this would have been a much more complicated process – installing the script package and teaching the AI model to use ALA transliteration rules, as well as everything else it was doing – we chose to omit this issue for now.

We are looking to begin experimentation with the Alma AI Metadata Assistant. This tool

https://doi.org/10.23106/dcmi.952526168

works within the Alma LSP's Metadata Editor and creates certain MARC fields based on images of the resource. Demonstrations have shown how it works with multiple languages and scripts, though we have not seen any examples where it creates contents notes for musical anthologies, which are often in many different languages in the same volume. We also look forward to hearing about other AI tools that might provide similar services, as the AI landscape is constantly changing and new resources are created regularly.

References

[1] Dougan, Kristin. "The 'Black box': how students use a single search box to search for music materials." *Information Technology and Libraries* 37.4 (2018): 81-106.

[2] Newcomer, Nara L. "Discovery." *Notes (Music Library Association)* 80.2 (2023): 249-258.

[3] Belford, Rebecca. "Evaluating library discovery tools through a music lens." *Library Resources & Technical Services* 58.1 (2014): 49-72.

[4] Newcomer, Nara L., Rebecca Belford, Deb Kulczak, Kimmy Szeto, Jennifer Matthews, and Misti Shaw. "Music discovery requirements: a guide to optimizing interfaces." *Notes (Music Library Association)* 69.3 (2013): 494-524.

[5] Garg, Muskan, Sandeep Kumar, Abdul Khader Jilani Saudagar, eds. *Natural Language Processing and Information Retrieval*, 1st. ed., CRC Press Unlimited, Boca Raton, 2023.

https://doi.org/10.23106/dcmi.952526168