Metadata & Discovery Group

Augmenting cataloguers

planning an AI Agent to generate MARC21 records

Sheldon Korpet
Library Systems and Discovery Advisor, Manchester Metropolitan University Library

Nathalie Rees
Library Systems and Discovery Librarian, Manchester Metropolitan University Library

Received: 02 June 2025 | Published: 17 June 2025

ABSTRACT

This article outlines the planning and development of an AI-powered agent designed to assist with generating records at Manchester Metropolitan University's Library and Cultural Services. The team explores the use of Large Language Models (LLMs) and Retrieval Augmented Generation (RAG) to automate and enhance cataloguing processes, particularly for unique and multilingual materials in the Special Collections Museum and Manchester Poetry Library.

The article discusses technical challenges, ethical considerations, and the importance of maintaining human oversight to ensure quality and transparency. It also details the system architecture and outlines future goals such as multilingual support and automated record enhancement. This project not only aims to improve efficiency but also empowers staff through upskilling and deeper engagement with emerging AI technologies.

KEYWORDS metadata enhancement; cataloguing automation; AI; Large Language Models; Retrieval Augmented Generation

CONTACT Sheldon Korpet S.Korpet@mmu.ac.uk Manchester Metropolitan University
Nathalie Rees N.Rees@mmu.ac.uk Manchester Metropolitan University

Introduction

We are part of the Digital Library Team operating within the Library and Cultural Services Department at Manchester Metropolitan University. Operating in a team of six colleagues, we use WorldCat (OCLC) as our Library Management System supplier. Our department has never had a dedicated full-time cataloguer and there is no funding for a specialist role.

We catalogue unique and self-published items for our Special Collections Museum and multilingual Manchester Poetry Library. However, cataloguing these items is a time-intensive process and there are competing demands for our expertise.

Our aim is to build an AI-powered Agent to assist in the cataloguing process. Albada (2025) defines agents as a piece of software, utilising a Large Language Model, to undertake a task without explicit instructions to initialise each individual action. The system will use an LLM to generate MARC21 format records using RDA standards and FAST subject headings. The intent being to improve cataloguing efficiency, reduce administrative load and improve item processing times.

While we acknowledge there have been significant AI advancements since 2022, cataloguers will continue to oversee the workflow. Ultimately, this AI system will suit our library staffing context, freeing staff to undertake innovative digital initiatives, such as horizon scanning, programming, and user experience research.

Past Automation Attempts

Previous in-house attempts have been made to automate the cataloguing process with various technologies (OCR, FAST API tools, document information extraction). However, an end-to-end system has not been implemented before.

Historically, AI has been rules–based but with the introduction of Large Language Models (LLMs) like transformers, it means AI can be flexible and competently dealing with natural language input (Gonuguntla, Meghana and Prajwal et al., 2024). We have experimented with commercial Generative Artificial Intellgence (GAI) tools like Copilot and DeepSeek. However, hallucinations are not uncommon, and staff found the results were mixed. Our conclusions were similar to industry findings from OCLC (Urban, 2024) as a cataloguer needs to read all the metadata generated to spot the mistakes as well as ensure it is correct and consistently conforms to our cataloguing processes.

GAI has proven itself capable of creating basic records, especially for popular titles, which can then be improved by cataloguers. However, generating DDC or subject headings is not always accurate. Creating successful prompts to produce useful records can be more time consuming than creating the record from scratch. The LLMs can bring in incorrect book information despite being given ISBNs or links to publisher pages.

Our aim is to tackle the unique, historic and self-published items we have purchased or received as donations to our Special Collections and Poetry Library. Therefore, we are looking to create a system capable of handling these items, which may not have standard layouts.

Discussion – Concerns, Barriers and Ethics

We will prioritise the accuracy and quality of records produced, mitigating the risk of introducing poor-quality records into WorldCat. However, this conflicts with internal pressure to get items out onto the shelves quickly, particularly if there is a large acquisition order or donation. There is a constant dilemma of whether it is better to create complete records in full or get a basic record on our system so items can be

discovered quickly but has to be revisited and improved later. GAI allows us to outsource speed by instantly drafting a record, which then allows staff to focus solely on accuracy, rather than searching for details and transcription.

Practical technical barriers to undertaking this activity include:

- staff skills and training
  - accessing funding and training resources,
  - having the time to learn,
  - keep up with advanced IT developments
  - building/configuring systems
- AI is a black box
  - transparency needs to be built into the systems
  - All models have some bias “baked in” due to how they are created (Resnik, 2024)
- making machine-readable documents
  - transcribing cataloguing processes and procedures in detail is time consuming
  - handwritten notes and non-OCR content is a challenge for machines to ingest
- I.T. infrastructure
  - having a laptop with enough computing power or access to a virtual machine
  - getting software purchase approval for pay-for-use tools and online platforms
  - installing software on institutionally managed laptops

Despite the technical barriers, by building a system ourselves we have control over the decision making and execution process. A criticism of the recent AI tools from publishers includes that the model displays overly broad sensitivity (Tay, 2025). By building a custom system, we can control the configuration to meet our requirements. Should we find the model is too focused or too broad, we can easily alter the parameters of the model.

Additionally, there is a chance OCLC may release a similar Agent to compete with the Alma offering that other institutions are currently testing. This would duplicate our efforts, but it’s something that we’d like to conceive rather than wait for the market to catch up. The main reason is because we can build our values into the Agent. We would prioritise the system's transparency and observability, allowing our cataloguers to understand how it is operating internally, thus improving usability and enabling user insights. This is something suppliers may not have time or inclination to explain.

We feel quite strongly about the ‘human in the loop’ approach. Cataloguing is not a science – anomalies arise, and decisions must be made to standardise record creation across staff and time. By having a human acting as a quality assessor who feeds back into the system, we aim to achieve two things:

- Consistency of output - catch errors at the source and ensure a level of standardisation across our catalogue.
- Training the agent – by having a human review and correcting AI generated records we are giving the system a high-quality dataset, allowing it a chance to see high quality records and ‘self-learn’ by embedding these into the vector database with domain knowledge

There are fears that AI could take work from humans. While we don’t have full-time cataloguers, we believe the role of cataloguer won’t disappear but its nature will change. We will still need to retain staff to oversee the operation of the system. Records need to be checked, new standards implemented, books physically labelled, new training documents written and processed into the system’s knowledge base. We should also monitor the functioning of the system. If anything, there is a need for change management and upskilling for cataloguers, rather than a risk of deprofessionalisation.

Even if we don’t take this system into our cataloguing workflow, it may be decided we should evaluate or purchase a similar product when it comes to market. The team will already possess the knowledge to critically evaluate technologies underpinned by LLMs, having had to think through and attempt the design process. Additionally, we also find projects like this often inspire other projects which would have otherwise never been discussed.

Technical Systems

Here we will describe two AI Agents. The first was a prototype and the second, design plans for a more advanced system with domain information but having access to more tools and memory.

Prototype build

We used RAG to give the LLM access to a domain-specific knowledge base. By allowing the LLM to access more information, it can aid it to generate an accurate and specific answer outside the areas increased in the model training data set (Mendelevitch and Bao, 2025). We used RAG to allow the agent to access specific information about how we catalogue for our library. LangChain was used as the agent framework, Chroma was used as the Vector database and Streamlit as the user interface. The system ran in the user's browser.

LangChain gave the document chatbot context, or a role to play. It instructs it to be a cataloguer, creating MARC21 format records, using RDA and FAST subject headings.

The documents in the vector database detail all our institutional guidance, as well as details about MARC and RDA which human cataloguers use (see 'Information sources').

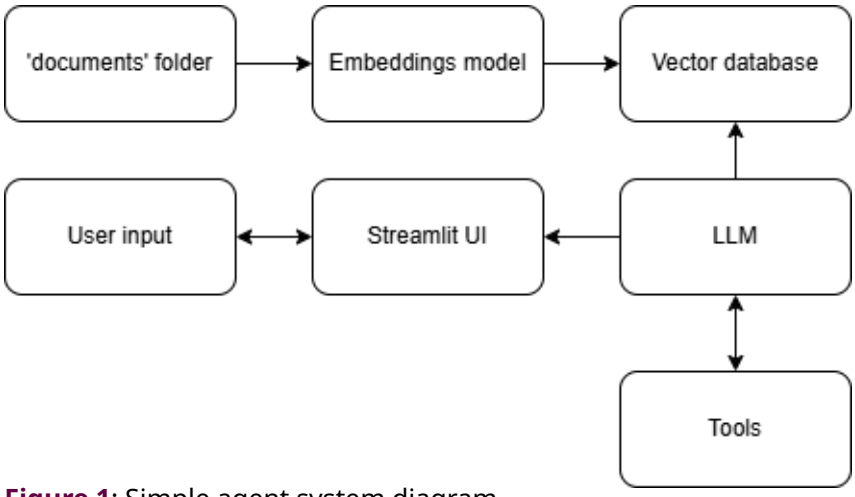

Figure 1: Simple agent system diagram

While some of the record generated was incorrect (subject headings were formatted using LCSH not FAST), the record generated showed enough improvement compared to commercial LLMs to decide to explore this technology further. We were impressed with the use of fields and noticed that the record was generating fields we often didn't have time to fill (505 for chapter titles and 520 for summary descriptions, if the information was provided).

Information sources

When deciding what to include within the RAG document store for the agent, we considered what a new member of the cataloguing team would need to know. Not just how to catalogue but also what our organisational practices and policies were. We provided the embeddings model with our policy and guidance documents, information about the DDC system, MARC21 fields, RDA documents, FAST Subject Headings and a few example records for specific collections.

Planning the new AI Agent

This is often done in the form of a 'Design Document' as best practice (Ubl, 2020). This helps share the idea, make transparent the system design plan, capture institutional knowledge and think through issues.

Design philosophy

There are five levels of complexity in Agents (Huang, 2024). Our first attempt was Level 3 – with the RAG system LLM having access to domain knowledge through the knowledgebase and basic memory and a simple web search tool. This time we will aim for Level 4 – the agent will have a cataloguing knowledge base still, but we will also build short term memory for individual conversations and a long-term memory

database. The system will be able to 'learn' in a simplistic way by saving any records generated as examples for the future.

Goals and non-goals

These aims help define the scope of the system and outline clearly how the system functions.

The system will not:

- be able to transcribe information with 100% accuracy
- be able to guarantee an accurate, complete record is generated
- be able to operate without human oversight

The system will:

- draft records almost instantly, ready for human review and correction
- have access to expert-identified and curated domain knowledge
- use tools to suggest a shelf mark
- use tools to suggest FAST subject headings
- have observability built in so threads and decision making can be reviewed by a human
- use a database for conversational persistency (long term memory)
- learn from record generation process by integrating records into its knowledgebase

System description

This agent will be given a Context on start-up (this defines its purpose):

> "You are a cataloguer for an academic library, special collections museum and poetry library. You create MARC21 format records using RDA and use FAST subject headings."

It will also have access to tools and built-in memory. LangGraph works by connecting nodes; information flows through these, allowing the process to iterate somewhat. Each agent can follow the conversation as a thread, and each thread is observable using LangSmith (another Python Library). That allows the logic process to become somewhat transparent to the user if further investigation is required.

The system will use a Streamlit user interface (UI) where cataloguers can input natural language queries alongside scanned images of book materials (covers, bibliographic information, author pages etc). The images submitted to the UI will be processed via a Vision Language Model (VLM). This will extract text from the images and show the user the transcription before records are generated. Cataloguers will be able to review the transcription of this material produced by a VLM, view and re-submit

records with errors. The system will be designed to work locally on our machines and we'll utilise GitHub to hold our code repository.

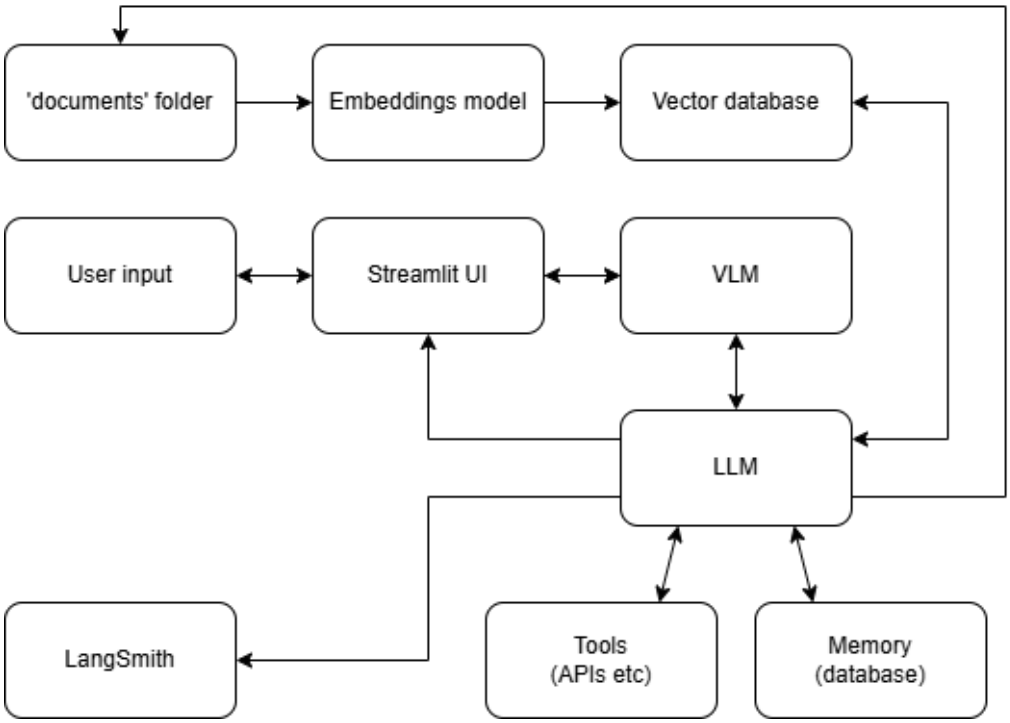

Figure 2: Advanced agent system diagram

The knowledge base will be created by a preprocessing script. An embeddings model will be utilised to make a Chroma database. This is a vector database with numeric representation of the domain documents provided to it. This allows the LLM to access the knowledge when generating a response to the cataloguer's inputs.

We plan to make use of OCLC APIs – Fast Search, Metadata. These will allow the agent to access data outside its models to choose subject heading information as well as check the system for existing records. Should the API connection fail, the system will alert the cataloguer and proceed to generate the record. The system will have clear and helpful error messages so the (likely non-programming) cataloguer will be able to understand what failed and feedback to the developer.

After the record has been generated, a copy will be saved locally as a .txt file. It will then be possible to run the preprocessing script before the next cataloguing session and give the agent an expanding bank of human quality assured cataloguing examples. As we are not finetuning a model, we can swap these records at will – should we want to create only a certain type of record e.g. we're working on a zine project, it should be possible to focus the agent on a specific subject to catalogue.

Framework

LangGraph is a Python framework that expands possibilities for more complex systems. It works by connecting nodes. Information flows through these, allowing the process to iterate somewhat. Functions include tool calling (allowing the LLM to grab data from APIs), built-in memory (both within the current conversation or thread and between sessions), and the use of an interrupt function (which stops the chatbot to allow the human to intervene and increase output quality).

Each agent is able to follow the conversation as a thread, and each thread is observable using LangSmith (another Python Library). That allows the logic process to become somewhat transparent to the user if further investigation is required. This is necessary as the developer gives the LLM access to tools – it doesn't specify when and which tools it should call or use.

The reasons for choosing LangGraph include:

- LangChain, the technology we previously used for RAG, is depreciating some of its functions to LangGraph.
- LangGraph is also popular, and this means there are plenty of online courses and community support.
- The documentation is comprehensive and well-defined.
- Additionally, it's built to utilise LangSmith, which is a platform which allows us to observe the application and evaluate performance.
- These are industry standard tools, so while unnecessary for a small project hosted remotely from our local machines, it's good practice and will provide a project to enable skills acquisition in the team.
- The framework is scalable, and it will give us options in the future should we wish to expand into multi-system agents (Level 5 - Agentic AI – see Huang, 2024).

Model choices

This is a difficult task as there are many model options and many evaluation tests. The key thing is to choose a model trained and capable of doing the task you are asking it to do (Kirmer, 2024).

It is possible to use commercial models e.g. Anthropic's Claude Sonnet model or ChatGPT 4.x. However, we would prefer to choose an open-source model as it's free. One tool which can be useful for identifying high performing LLM models are Hugging Face Leaderboards (Hugging Face, 2025).

We will likely need three models for the system:

- a Vision Language Model (VLM) - to transcribe text from item scans
- an LLM to create the record and "converse" with the user in natural language
- an Embeddings model to be used with the document store and create the vector database (domain specific knowledge which will be accessible to the LLM)

Testing Record Generation

Assessing the effectiveness of the system can be done by providing a human with the item to be catalogued and comparing the record quality (presence of fields, relevant subject headings etc) of both. The system will be able to produce a record faster every time, so what we will be directly comparing/measuring is how accurate the transcription is, how well the record represents the item, and how much detail the records have.

The system can also be compared directly to other commercial LLM services without domain knowledge via RAG. This should demonstrate how having access to process documents and specialist knowledge changes generative record production.

Future Development

We see potential for development in four areas:

- Collaboration - we're discussing with other libraries whether projects like this can be made into collaborative, open-source efforts.
- Technology - We would like to explore using this technology to automatically enhance current records, such as adding more subject headings to aid discovery as well as evaluating its ability to catalogue in other languages for our Poetry Library.
- Skills and Training – projects like this allow a framework for staff to develop, upskill, engage in interesting and challenging work.
- Efficiency – reducing administrative tasks, like typing and transcribing, and creating time efficiencies.

Glossary

Agent - a piece of software, utilising a Large Language Model, to undertake a task, make decisions without humans. More advanced ones have memory and call tools.

Hallucination - mistakes the Generative AI model makes when creating an answer. Often a result of poor training data, incorrect assumptions, or biases (Google, 2024)

Retrieval Augmented Generation (RAG) - the improvement of LLM outputs by allowing them to access domain specific information.

LangGraph - a Python library. It is prewritten code which can create the framework for generative AI agents without coding everything from scratch (Taulli, 2024).

Large Language Models (LLM) - Machine Learning models, often with billions of parameters. They are trained on massive datasets and often generate language or perform natural language processing.

Machine Learning – computers learn and identify complex patterns from data without being explicitly programmed by humans.

Python – a versatile programming language used mostly for AI, Machine Learning, automation.

Vision Language Models (VLM) - Large Language Models which can process images, extract key data like colours and shapes, then make observations available to other models via mathematical representations.

References

Albada, Michael (2025) *Building Applications with Agents*. O'Reilly. Available at: https://learning.oreilly.com/library/view/building-applications-with/9781098176495/ [Accessed: 20 May 2025]

Gonuguntla, H., Meghana, K., Prajwal, T.S., Koundinya, N.V.S.S., Sai, K.N. and Jain, C. (2024) 'Evaluating Modern Information Extraction Techniques for Complex Document Structures', *International Conference on Electronic, Computer and Energy Technologies (ICECET 2024)*, Sydney, 25-27 July 2024. Available at: https://doi.org/10.1109/ICECET61485.2024.10698618 [Accessed: 30 April 2025]

Google (2024) *What are AI hallucinations?* Available at: https://cloud.google.com/discover/what-are-ai-hallucinations [Accessed: 28 May 2025]

Huang, Yu (2024) *Levels of AI Agents: from Rules to Large Language Models*. Available at: https://doi.org/10.48550/arXiv.2405.06643 [Accessed: 28 May 2025]

Hugging Face (2025) *Leaderboards and Evaluations*. Available at: https://huggingface.co/docs/leaderboards/en/index [Accessed: 28 May 2025]

Kirmer, Stephanie (2024) *Choosing and Implementing Hugging Face Models*. Available at: https://www.stephaniekirmer.com/writing/choosingandimplementinghuggingfacemodels [Accessed: 28 May 2025]

Langchain (2025) *Langgraph*. Available at: https://langchain-ai.github.io/langgraph/ [Accessed: 19 May 2025]

Mendelevitch, Ofer and Bao, Forrest (2025) 'Rag vs. fine-tuning' in O. Mendelevitch and F. Bao *Hands-On RAG for Production*. Available at: https://learning.oreilly.com/library/view/hands-on-rag-for/9798341621701 [Accessed: 20 May 2025]

Resnik, Philip (2024) *Large Language Models are Biased Because They Are Large Language Models*. Available at: https://doi.org/10.48550/arXiv.2406.13138 [Accessed: 23 May 2025]

Taulli, Tom (2024) *What is LangGraph?* O'Reilly. Available at: https://learning.oreilly.com/videos/what-is-langgraph/0642572071776/ [Accessed: 20 May 2025]

Tay, Aaron (2025) 'Deep Dive into Three AI Academic Search Tools', *Katina*, (20 May). Available at: https://doi.org/10.1146/katina-052025-2 [Accessed: 20 May 2025]

Ubl, Malte (2020) 'Design Docs at Google', *Industrial Empathy*, 6 July. Available at: https://www.industrialempathy.com/posts/design-docs-at-google/ [Accessed: 28 May 2025]

Urban, Richard (2024) 'Keeping up with next-generation metadata in archives and special collections', *Hanging Together*, 17 December. Available at: https://hangingtogether.org/keeping-up-with-next-generation-metadata-in-archives-and-special-collections/ [Accessed: 8 May 2025]