A Few Thoughts on Artificial Intelligence in Libraries

Below are a few thoughts on the topic of artificial intelligence in libraries. They are here to provide some context for an online conference call on the same topic.

The phrase "artificial intelligence" was coined by John McCarthy in 1956 as a part of a Dartmouth Summer Research Project. When compared to the amount of time digital computers have existed (since around 1945) the phrase "artificial intelligence" has been around since almost the beginning. Thus, the concept of "artificial intelligence" is not new. In fact, it is old.

Second, the question, "What is 'artificial intelligence'?" begs the question, "What is intelligence?", and I believe we would be hard pressed to articulate a compelling answer to either question. Moreover, even if we were able to articulate a definition, why would anybody be interested in "artificial" intelligence. Such a thing sounds too much like "artificial flavorings" or "artificial sweeteners". Think saccharine.

That said, libraries are not new to artificial intelligence. In the very late '80s and early '90s a type of artificial intelligence was being explored, and such explorations were called "expert systems". Entire books were written on the topic, and more than a few library-related expert systems were developed. They centered mostly around the process of automating reference interviews or cataloging of books. As a part of fulfilling a grant sponsored by the National Library of Medicine, I wrote such a system, and it was called "Ask Eric".

I do not advocate the use of the phrase "artificial intelligence" to describe the technology of today's forum. I do not advocated it because it is more of a misnomer than anything else. Again, "What is artificial intelligence?" A better nomenclature, might be "machine learning", but then again, "What is learning?" The phrase/word "large-language model" and "generative" are, in my opinion, more accurate. Why? Because they better describe what is going on. Large-language models are very, very, very large sets of vectors. These vectors "point" to locations in n-dimensional spaces, and they are geometrically compared to each other for the purposes of denoting similarity. Using this technology -- an application of linear algebra -- a large-language model is given a word, the word is located in the n-dimensional space, and similar words are identified. In this way the model generates sentences. What we are dealing with are models of human language -- mathematical representations of things written. This not "intelligence". Instead, it is statistical analysis on a scale we have never previously seen. But alas, the phrase "artificial intelligence" is catchy.

Ironically, librarians love models; libraries overflow with models. Think of all the lists libraries produce. Each is a model. Bibliographies model the things cited in an article. Our catalogs model library holdings. An Encoded Archival Description file is a model of an archival collection. Librarians have used all of these things to describe collections and provide services against them. Why not use large-language models? Well, one reason is they do not necessarily model our collections.

But two specific applications of large-language models can be used to model library collections, sort of. The first is called "retrieval-augmented generation" (RAG), and the second is called the Model Context Protocol (MCP). In the former, sets of library-related materials are vectorized (read "indexed"). A query is garnered from a person, the query is vectorized, matching documents are identified, and the result is given as input to a large-language model for interpretation. For example, a librarian might index all of the sentences in a given book, query the book for relevant sentences, and then use a large-language model to summarize the result or address a question.

MCP servers work in a very similar manner. First the librarian creates a collection. Second, the librarian creates an application programmer interface (API) to interact with the collection. Third a MCP server is used to garner natural language commands from a person, a large-language model maps the commands to the API and submits a call to the API, the API's result is returned to the large-language model, and some sort of "answer" is presented to the person. Thus, MCP wraps an API with natural language inputs and outputs derived from large-language models. For example, MCP can garner a natural language query, convert it into SQL, get the SQL response, and interpret the result. Much easier than writing SQL by hand!

The trick to effectively implementing RAG is the initial vectorization process, not the application of the LLM. Similarly, the trick to implementing MCP is effectively implementing an API, not the LLM. You have indexed things. Right? Half the work is done. Your library overflows with APIs. Doesn't it? Again, half the work is done. Now you can put a friendly front-end on the index and/or the API, and as a bonus, you can add interpretation to the results.

Computers and library work have had a very long relationship. At the very least, think MARC, which dates from 1965. Large-language models are yet another iteration of this computer/library relationship. I believe it behoove us -- the library profession -- to explore what this technology can and can not do, and the result will inform our perceptions of it. It is a tool, and we ought to learn about this tool in order to learn how to use it effectively.

Eric Lease Morgan <emorgan@nd.edu>
July 1, 2026