



NYPL & NYU

Integrated Content Management Platform to Grow and
Evolve Discoverability of Large Library Collections

Course: Information Technology Projects
CSCI-GA 3812 Fall 2024

Team

Simon Zeng sz4559@nyu.edu

Rahul Singhal rs8960@nyu.edu

Tanmay Khandelwal tk3309@nyu.edu

Fei Wang fw799@nyu.edu

Professor

Jean-Claude Franchitti jcf@cs.nyu.edu

Technical Advisor

Joanna Gilberti jlg204@nyu.edu

NYPL Supervisor

Eric Shows ericshows@nypl.org

Table of Contents

Introduction	4
Context	5
Business Problem	5
Solution	5
Technical Solution	6
Improvements in Existing Architecture	6
DigiSuite Classifier	6
Improve Classification Performance	6
Performance Evaluation	9
LLM-ATGen	10
Q&A Bot	16
Implementation Details	18
User Interface	18
Benefits and Impact	21
Challenges	21
Integration with Existing Infrastructure	21
Generating Meaningful Alt-Text Descriptions	22
Future Work	23
Contextual Enrichment	23
Dynamic Knowledge Extraction	24
Multimodal Fusion	24
Real-Time Collection Updates	24
Appendix	25
DigiSuite Classifier	25
LLM-ATGen	26
QA-Bot	26

Introduction

In today's digital era, libraries around the world are adopting cutting-edge technologies to preserve, organize and distribute their extensive collections. The New York Public Library (NYPL), the second largest public library in the United States and the fourth largest public library in the world, is no exception to this and is one of the leading pioneers in the fields.

Since 1895, NYPL has been a central institution in New York City, playing a key role in community education and cultural engagement. It currently operates 92 locations, including research centers, across the Bronx, Manhattan, and Staten Island. Across those various locations, the library serves over 17 million patrons annually and millions more online. The library holds more than 55 million items, ranging from books, e-book and DVDs to renowned research collections used by scholars worldwide.

Given the library's extensive collection of physical assets and broad community reach, the NYPL maintains a significant online presence. Its digital archives and tools offer free access to thousands of historical and cultural documents. However, managing and expanding upon such a vast array of digital resources to improve NYPL patron experience present unique challenges. These include digitizing through extracting information from documents, improving the discoverability of its collections, and assisting with inquiries about them. This has persisted as a time and labor intensive task, heavily dependent on librarians. Fortunately with technological advancements, particularly in machine learning, it has become possible to automate many of these efforts, leading to the creation of the NYPL DigiSuite platform.

Our capstone project, developed in collaboration with NYPL, modernizes, enhances and expands the DigiSuite platform. First, we refined the existing information extraction workflow through upgrading the artificial intelligence model. Secondly, we developed a question & answer (Q&A) system that facilitates direct interactions with the digital archive, allowing users to pose questions and receive specific, accurate answers. As a result, this system not only improves NYPL patron engagement but also makes the wealth of historical and cultural information more accessible to a broader audience.

Context

Business Problem

The New York Public Library (NYPL) stands as one of the largest and most influential libraries in the world, providing access to millions of books, historical documents, and digital collections. Its mission is to inspire lifelong learning, advance knowledge, and strengthen communities by ensuring that information is accessible to all. As a cornerstone of cultural preservation, NYPL's extensive digital collections include resources such as photographs, handwritten letters, and manuscripts, playing a critical role in democratizing access to cultural heritage.

With the ever-growing volume of digital content, the Library faces a pressing need to enhance its organizational and search capabilities to meet modern accessibility standards. Ensuring the accuracy of classification and metadata is crucial to reducing barriers to discovery and maximizing the impact of NYPL's digital archives. Additionally, as user expectations continue to evolve, implementing user-friendly systems like interactive Q&A tools and leveraging AI-driven classification techniques will be essential for maintaining NYPL's leadership in public knowledge accessibility and ensuring that its resources remain relevant and impactful for future generations.

Solution

The DigiSuite platform aims to address existing challenges such as inefficient search capabilities, manual metadata generation, and limited classification accuracy by implementing targeted enhancements. These solutions are designed to modernize the system and provide a more efficient and accessible platform for users.

One of the primary improvements is the development of an intelligent Q&A system powered by large language models (LLMs). This system is designed to deliver real-time, context-aware answers and offer dynamic follow-up support for users. By integrating advanced tools such as knowledge graphs and vector databases, the solution facilitates faster and more context-driven access to relevant collections.

Additionally, the project focuses on improving classification accuracy. Existing models are being upgraded to minimize false positives and ensure precise organization of digital assets. This enhancement is essential for improving user trust and efficiency when interacting with the system.

Lastly, the platform introduces automated metadata generation to replace traditional, manual, and error-prone processes. By leveraging enhanced classification models like ATGen and HWTGen, the solution aims to improve accuracy while significantly reducing manual effort. This automation not only streamlines operations but also ensures consistent and high-quality metadata production across the library's vast collections. Together, these initiatives position DigiSuite as a robust and scalable platform for addressing NYPL's evolving digital needs.

Technical Solution

Improvements in Existing Architecture

We implemented several enhancements to the existing model and architecture in DigiSuite as part of our solution.

DigiSuite Classifier

Improve Classification Performance

The primary objective of our solution with respect to the DigiSuite classifier is to reduce the number of incorrect classifications it makes. The reason for this is two-fold. In the previous iteration of DigiSuite, errors in this initial classification would result in documents being directed to the incorrect model, leading to poor data extraction from these downstream models. For example, sending an image of a handwritten letter to ATGen would yield poor alt-text generation and similarly, sending an photo of a play to HWTGen would yield little to no text extraction from the image due to the lack of text within the image.

Additionally, we also saw that many documents within the NYPL collections included both photos and handwritten text, as seen in Figure 1 below. Note that due to the varying fonts available in NYPL collections as well as the inability for many general purpose image-related AI models to recognize many of them, we refer to handwritten fonts as any text existing within an image.

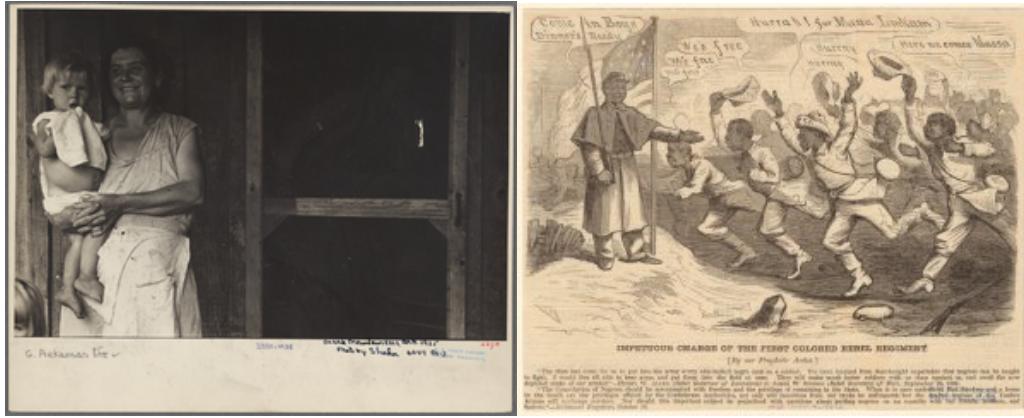


Figure 1. NYPL documents that contain both photos and handwritten text

When these documents are sent into only a single downstream model with a single binary classification, there is significant information loss on sub-components of the image.

To address these problems, we changed the DigiSuite classifier to perform window-level classification rather than the image-level classification it had previously done. By doing so, we allow for the addition of a third class label of “Both” for the entire document, which would initiate the information extraction workflow for both downstream models. The DigiSuite classifier first preprocesses images and sends them through a forward pass to generate rects (bounding box information for text regions) and confidences (the model’s confidence on each rect containing text). The rectangles generated from this algorithm are provided in Figure 2 below.



Figure 2. Algorithm-generated rectangles of handwritten text

From here, the resulting label differs based on the fulfillment of three different scenarios:

Case 1: *If there are fewer bounding boxes than `LOWER_BOX_THRESHOLD`, the document is classified as AT_Gen.* The rationale behind this is that if there are too few boxes that the model is confident contain handwritten text, then the image is primarily a photo.

Case 2: *The existing bounding boxes have their area calculated and converted into a percentage of the total area. If the uncovered percentage of the image is greater than `UNCOVERED_THRESHOLD`, the document is classified as Both.* The rationale behind this is that if there are a significant number of boxes containing handwritten text yet they do not comprise the majority of the area of the document, then the document likely contains both images/photos as well as handwritten text.

Case 3: *The existing bounding boxes have their area calculated and converted into a percentage of the total area. If the uncovered percentage of the image is less than or equal to `UNCOVERED_THRESHOLD`, the document is classified as HWT_Gen.* The rationale behind this is that if the document comprises primarily boxes containing handwritten text, then it is likely only a document containing handwritten text.

To illustrate this process, particularly for the case where both handwritten text and images are present in the document, Figure 3 is provided.

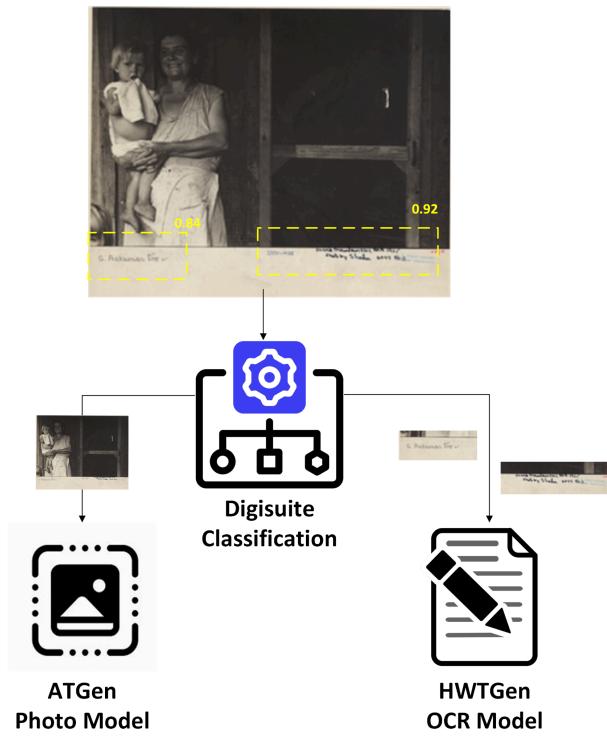


Figure 3. Information extraction workflow for documents containing both text and images

The effectiveness of this algorithm is elaborated upon in the following section: Performance Evaluation.

Performance Evaluation

Another shortcoming of the previous DigiSuite classifier was the lack of a way to evaluate its performance – it was dependent on manual and qualitative evaluation done by either developers or librarians. This makes it difficult to understand when changes to the model result in actual performance improvements.

To address this, we put together 956 documents from NYPL collections that the DigiSuite classification model was not trained on. These documents were then manually labeled as “AT_Gen,” “HWT_Gen,” or “Both.” With this test set, we were able to evaluate the original model’s performance and understand the improvements made by our window-level classification adjustments. The results are shown in Table 1.

Model	Accuracy	Notes
Original (Pre-Fall 2024)	59.5%	Performance on entire test set
Original (Pre-Fall 2024)	64.7%	Performance on examples in test set not labeled “Both”
Ours (Fall 2024)	80.7%	+21.2% from Original

Table 1. Performance of DigiSuite Classifier on Test Set

In doing this, we see that our window-level classification improved performance by 21.2%, showing an effective change in performance for the model due to the adjusted algorithm. This test set will also allow for future teams to evaluate model performance changes as a result of changes to the DigiSuite classifier.

LLM-ATGen

Another shortcoming of the previous DigiSuite information extraction workflow is the ATGen model. ATGen is the model that was responsible for generated descriptions given an input image. However, in the previous iteration, ATGen generated output that was generic and vague, as seen in Figure 4 below. By missing critical details in the photos of documents, the existing descriptions generated by ATGen would negatively impact downstream knowledge base searches, counteracting the original purpose of the platform in enhancing customer engagement with NYPL collections.



a man standing in front of a screen



a group of people standing next to each other

Figure 4. Vague descriptions generated by previous ATGen

To address this shortcoming, we changed the backend model powering ATGen with a more powerful and modern large multimodal model called LLaVa-Next¹. As shown in Figure 5, this model consists of two components: a pre-trained CLIP Vision Transformer (ViT) and a pre-trained Llama2 model.

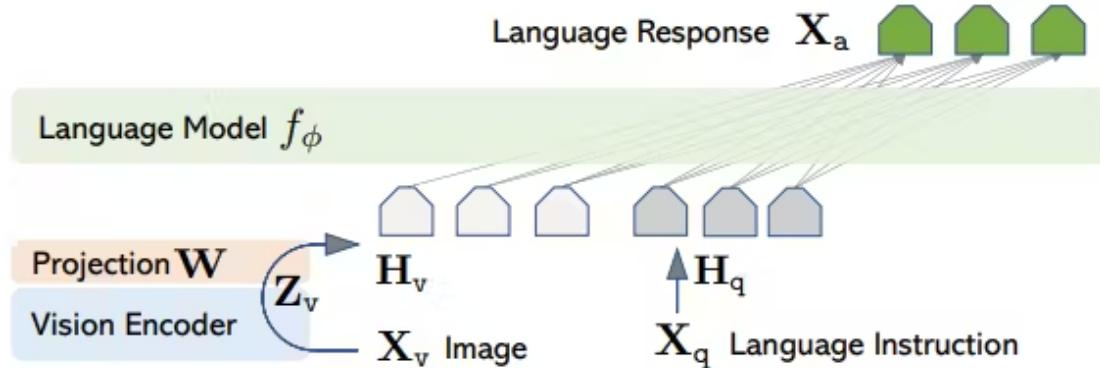


Figure 5. Architecture of LLaVa-Next model

The CLIP ViT², developed by OpenAI, is a vision encoder component of the CLIP model that processes images and extracts semantic features from those images. It uses a transformer-based architecture to interpret images as sequences of patches, enabling a rich representation of visual content. It is pre-trained on a large-scale dataset (~400 million) of image-text pairs curated by OpenAI, allowing it to generalize well to a wide variety of tasks.

¹ LLaVa-Next details and original research paper found here: <https://llava-vl.github.io/blog/2024-01-30-llava-next/>

² CLIP ViT details and original research paper found here: https://cdn.openai.com/papers/Learning_Transferable_Visual_Models_From_Natural_Language.pdf

Llama2³ is a large language model developed by Meta that is capable of understanding and generating human-like text. It is pre-trained on a large corpus of publicly available text datasets (such as books, scientific articles, websites), allowing for it to have a deep understanding of language structure and reasoning and the surrounding world, allowing for it to interpret and generate natural language.

Combined, the CLIP ViT is able to capture complex visual semantics and contextual information and convert it into vectors. Llama2 then takes these vectors and a thorough understanding of the world and language to generate text. To further refine the text that it generates to more closely align that of descriptive alt-text, we then fine-tune LLaVa on a subset of the LAION-COCO dataset⁴ (~3400 image-description pairs).

We opt for a subset of the LAION-COCO dataset for multiple reasons. First, because the LAION-COCO dataset is known for its thorough descriptions of images, we can fine-tune LLaVa on it to generate equally descriptive descriptions. Secondly, we opt for a subset due to requiring to preprocess the images to ensure that they are still linked to existing images (some links direct to images that are no longer being hosted) and the computational time necessary for the model to fine-tune. For a complex task like this, we find that excessive fine-tuning led to decreased performance (possibly due to overtraining and the loss of generalized performance) and excessive train time. Finally, we can use the LAION-COCO dataset to evaluate the performance of this model over time – in this case, we chase the minimization of the loss on the validation set but future works can leverage other metrics to directly compare the generated descriptions and true descriptions, such as BLEU score.

The results of this fine-tuned LLaVa-Next model are displayed below in Figure 6. The descriptions provided are far more descriptive than the previous iteration of the ATGen, allowing for more meaningful knowledge to be available for downstream knowledge base searches. One noteworthy observation of the multimodal model is its ability to identify some handwritten text in the images (as shown in the first image of Figure 6, where the text on the whiteboard is attempted to be recognized). Although current multimodal models don't seem to have robust enough capabilities to accurately and reliably recognize this text, we expect this to be an area of improvement in future multimodal models.

³ Llama2 details and original research paper found here: <https://ai.meta.com/research/publications/llama-2-open-foundation-and-fine-tuned-chat-models/>

⁴ LAION dataset details found here: <https://laion.ai/blog/laion-5b/>



Prompt: Answer the questions. What is shown in this image?

Answer: The image shows a man standing in front of a chalkboard with the word "Fodor's" written on it. He is dressed in a suit and tie, and appears to be presenting or explaining something related to the chalkboard. The setting suggests that this could be a classroom or lecture hall.



Prompt: Answer the questions. What is shown in this image?

Answer: The image shows a woman holding a baby. They are standing in front of a wooden door. The woman is smiling and appears to be happy. The baby is also smiling and seems to be enjoying the moment. The photo has a vintage look, suggesting it was taken some time ago. The woman is wearing a dress, and the baby is wrapped in a blanket. The wooden door behind them adds a rustic touch to the image.

Figure 6. Results of the fine-tuned LLaVa-Next on Input Images

We integrate the LLM-ATGen (without context) into the DigiSuite platform and rerun on select collections (NYPL_CC and NYPL_FPC) to generate the following results in Figure 7 and Figure 8.

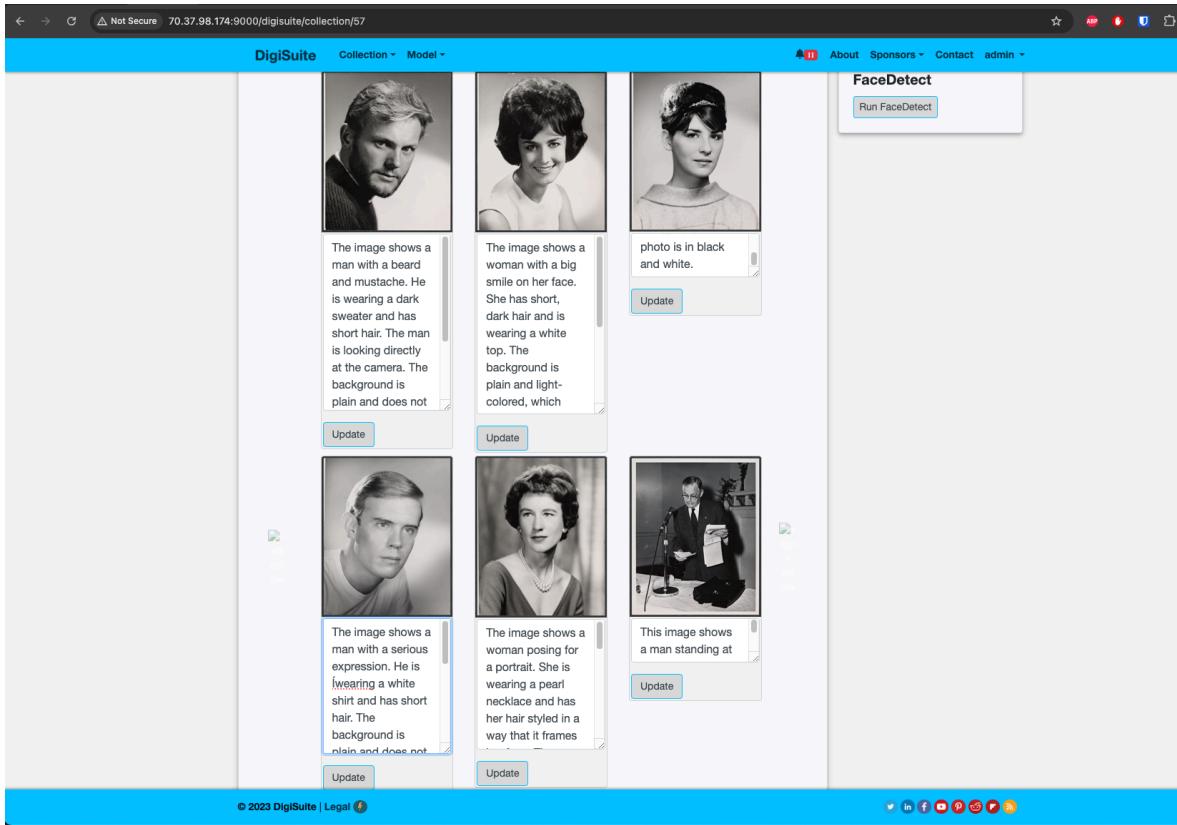


Figure 7. LLM-ATGen results integrated into DigiSuite platform - NYPL_FPC collection

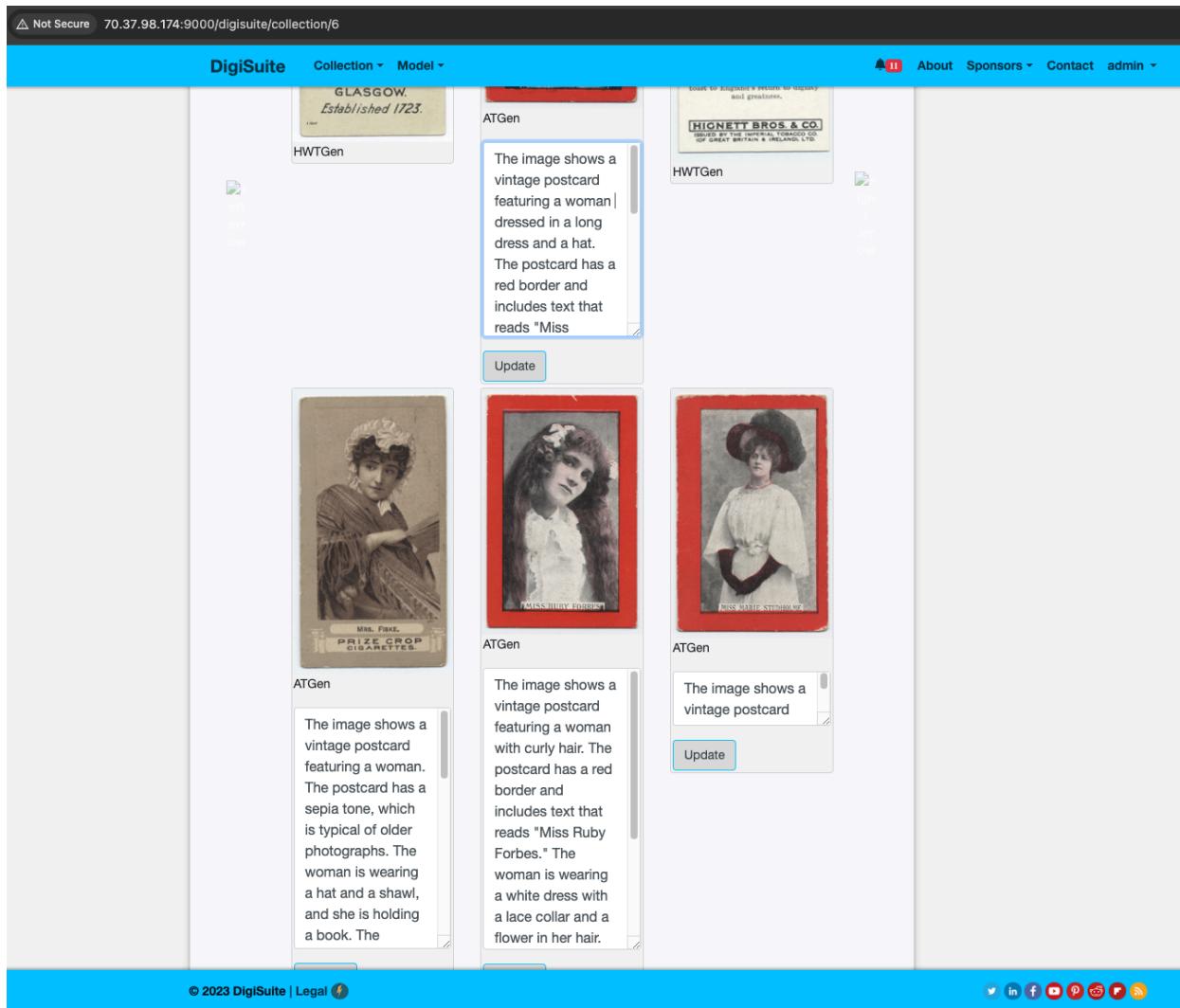


Figure 8. LLM-ATGen results integrated into DigiSuite platform - NYPL_CC collection

Q&A Bot

As part of our effort to modernize NYPL's digital infrastructure, we developed a Q&A Bot that allows users to pose questions about collections and receive contextually relevant answers. The high-level architecture for this component is illustrated in **Figure 9** below.

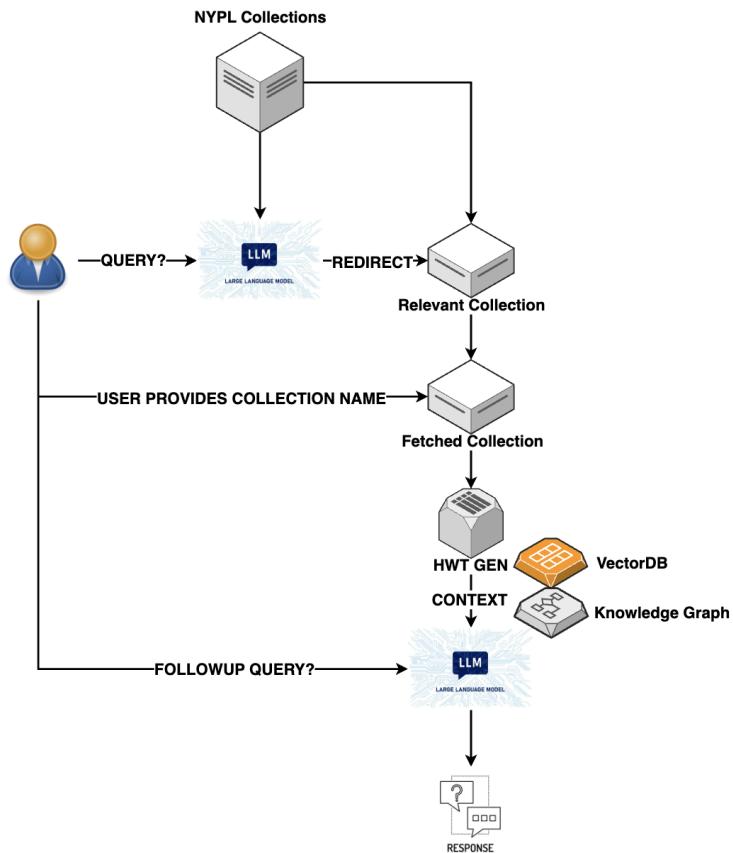


Figure 9. Q&A Bot Architecture

User Query

A user initiates a session by typing a question—such as “Which collection contains the personal letters of John Smith?” or “What is the significance of the building shown in this photo?” In essence, the Q&A Bot aims to serve both casual inquiries (e.g., general knowledge lookups) and more scholarly research questions (e.g., historically contextual inquiries about manuscripts).

Large Language Model (LLM) Interaction

The user query is first routed to a large language model, which serves two key purposes:

- **Understanding:** The LLM interprets the user's intent and potentially reformulates the question to be clearer. For example, if the user says, “Are there any letters about the design of the Cadillac Place Building here?” the LLM will distill that into a more explicit request, such as “Show me all letters referencing the Cadillac Place Building, specifically its design or architecture.”

- **Initial Retrieval:** The LLM can identify, from high-level metadata, which collections are likely relevant to the question. For instance, it might identify that “NYPL_FPC” or “NYPL_CC” contain relevant documents.

Redirect to Relevant Collection

Depending on the user query and the LLM’s classification, the system can redirect the search into one or more relevant collections. In many cases, the user might provide a specific **collection name** they wish to explore further (“I want to see documents in the NYPL_FPC collection”). In other scenarios, the LLM infers which collection is most appropriate based on the text or keywords.

Fetching the Collection

Once a relevant collection is identified, that collection’s documents—often pre-processed by DigiSuite classifiers—are fetched for deeper analysis. The Q&A Bot architecture leverages:

- **Vector Databases (VectorDB):** Documents have been embedded into a vector space (via sentence-transformers or similar LLM-based embeddings). This enables semantic matching rather than just keyword matching.
- **Knowledge Graph:** If present, structured metadata or relationships between entities (e.g., authors, publication dates, locations) can be consulted to help refine the answers.

Contextual Answer Generation

Within the Q&A Bot, a dedicated text generation workflow (HWTGen or a comparable pipeline) can further extract or transform text from the documents. Then, the LLM is invoked again, this time with the relevant chunks or data from the VectorDB/Knowledge Graph. This step ensures that **answers are grounded in the actual library content** rather than “hallucinated” by the model.

- **Context Injection:** The retrieved snippets—either alt-text from images (ATGen or LLM-ATGen) or extracted handwritten text (HWTGen)—are fed into the prompt. This provides crucial factual grounding for the final response.
- **Follow-Up Queries:** If the user asks a clarification or a subsequent question (e.g., “What is the date on that letter?”), the system keeps track of the conversation’s context. The LLM then re-checks the relevant subsets of data to address the follow-up in context.

Response Delivery

Finally, the system compiles the relevant information and returns a concise, coherent answer to the user. For instance:

“The Cadillac Place Building was designed by architect Albert Kahn. A handwritten letter found in NYPL_FPC details his involvement and is dated July 28, 1915.”

Implementation Details

1. Vector Store Creation

- **FAISS or Equivalent:** We parse PDF documents, or images with OCR, and chunk them into segments of ~1,000 characters. Each chunk is embedded using a Transformer-based embedding model (e.g., [sentence-transformers/all-MiniLM-L6-v2](#)) and stored in a vector index, enabling fast similarity lookups at query time.

2. Conversation Handling

- **LangChain ConversationalRetrievalChain:** We employ a conversation chain that keeps track of user messages and model replies. When a new user query arrives, the model:
 1. Rewrites the query (if necessary) to a self-contained question using a specialized prompt template.
 2. Searches the vector index for relevant context chunks, factoring in the user's conversation history and references to prior content.
 3. Generates a final answer that cites the sources of the retrieved documents (e.g., "Source: [letters.pdf](#)").

3. Workflow Logic

- **Collection Confirmation:** If the user's question includes "Which collection has X?," the system can propose a collection, then wait for the user to confirm or cancel.
- **Multi-Turn Interactivity:** By preserving conversation state, the bot supports follow-up queries without losing context—particularly useful when referencing old letters or alt-text from images.

4. Alt-Text Integration

- Documents that contain images are processed by **LLM-ATGen** (the updated alt-text generator) or older alt-text data. If the user's question requires analyzing the imagery (e.g., "Who is the person in this photograph?"), the Q&A Bot includes alt-text or recognized text in its conversation context, enabling more accurate responses.

User Interface

To make the Q&A Bot readily accessible within DigiSuite, we placed it as a dedicated menu item under **Tools** in the top-level navigation bar (see **Figure 10**). By selecting **QA Bot** from this dropdown, users are taken to a chat interface where they can pose natural-language questions about the library's digitized documents and receive context-specific answers.



Figure 10. DigiSuite platform's main interface, highlighting the new "QA Bot" option

This placement was chosen for two key reasons:

1. **Seamless Access:** Locating the Q&A Bot alongside other core DigiSuite functionalities (Connect, New, Update, Annotate, etc.) ensures that users can quickly switch between document management tasks and question-answering sessions.
2. **Consistent User Experience:** By keeping the Q&A Bot within the same design framework (navigation bar, color scheme, and layout), we maintain a cohesive look and feel, minimizing the need to learn a separate workflow.

Below is an example of the Q&A Bot's user interface. When the user asks, "Who was the designer of the theme building?" the system identifies a relevant collection—"NEW YORK WORLD'S FAIR 1939 AND 1940 INCORPORATED RECORDS"—and offers to search within it. Upon confirmation, the bot returns an answer grounded in the documents' contents: "The designer of the Theme Building at the New York World's Fair 1939 was Mr. Henry Dreyfuss."

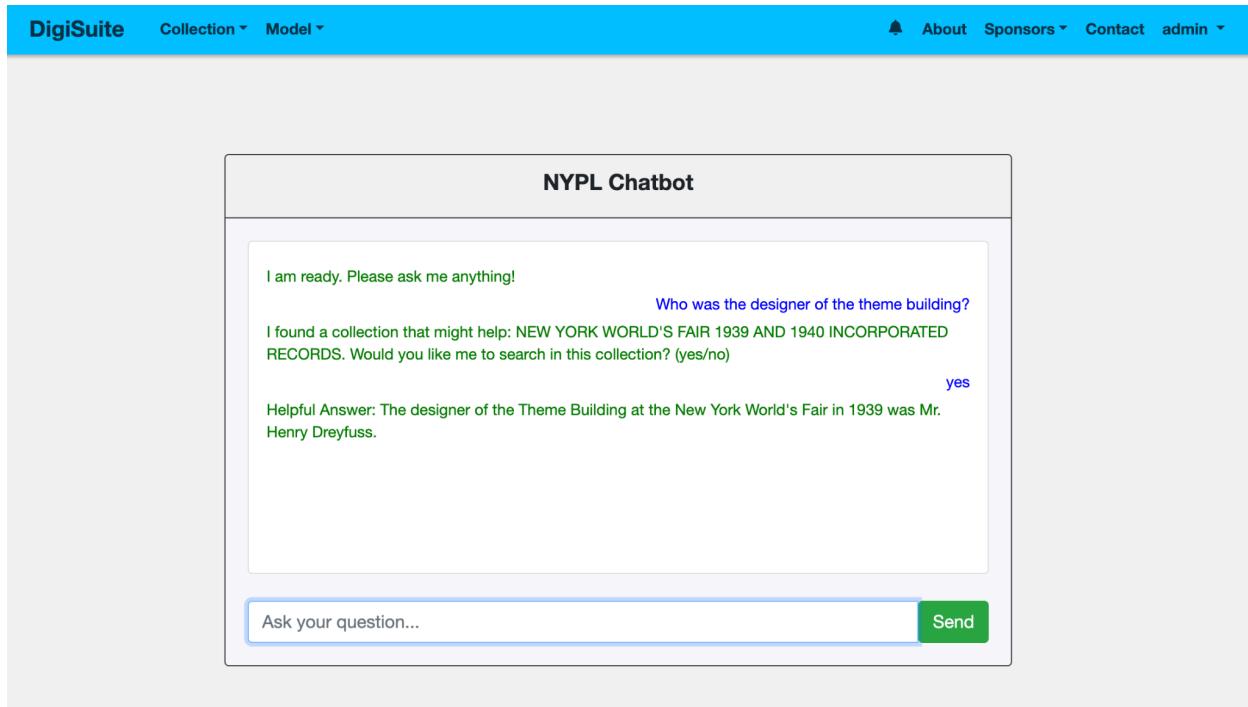


Figure 11. Q&A Digitsuite Interface

This interaction demonstrates the Q&A Bot's workflow in action:

1. **User Query:** The user requests specific information.
2. **Collection Suggestion:** Based on the query, the system locates a relevant collection, in this case the World's Fair records.
3. **Confirmation:** The bot asks if the user wants to proceed with that collection, a step that both clarifies user intent and reduces irrelevant searches.
4. **Response Generation:** After confirmation, the bot retrieves and synthesizes information from the selected records to deliver a direct, contextually accurate answer.

By prompting the user to confirm or reject collections, the Q&A Bot keeps patrons in control of the search process, ensuring they explore the most relevant datasets while minimizing extraneous results. This design also reflects a conversational paradigm that can be extended for follow-up queries—such as "Were there any other notable designers at the 1939 World's Fair?"—without losing context.

Benefits and Impact

- **Improved Discoverability:** Patrons can ask natural language questions and be directed to the most relevant documents, rather than manually searching vast catalogs.
- **User-Centric Interaction:** Using an LLM-driven conversation lowers the barrier to entry for exploring archives, appealing to non-expert researchers or casual library users.
- **Scalable Architecture:** By leveraging VectorDB and knowledge graphs, the approach scales as more collections are digitized and integrated into DigiSuite.

This Q&A Bot thus represents a key step in **delivering interactive, on-demand insights** into NYPL's digital collections. Over time, as the alt-text generation and classification models improve, the Q&A Bot's accuracy and depth of answers will likewise become more robust, providing a richer research experience and enhancing the overall discoverability of the library's vast holdings.

Challenges

Integration with Existing Infrastructure

A significant challenge faced with this project was the integration of our new features with the existing infrastructure. The previous code was deployed on NYPL servers but had various unused subdirectories (containing outdated code from previous years) and unsuccessfully integrated code, such as the FaceDetect capabilities, which did not appear to work as expected during our testing. This proved to be problematic especially when integrating the model enhancements (which required understanding where previous models were being used). We suggest that future teams spend ample time in the beginning of the semester to familiarize and properly set-up development environments to ensure success in this area.

Additionally, although we were looking to enhance the output of the ATGen model's alt-text descriptions, the development efforts had to go in parallel with the development of the Q&A bot, meaning that the Q&A bot had to use existing alt-text descriptions and extracted handwritten text. Many existing descriptions proved to be too vague to provide meaningful contextual

information for the Q&A bot to use in answering questions. For example, our Q&A bot can be queried to find connections between entities in the extracted data (such as “who was the architect of the Cadillac Place Building”). However, the existing ATGen descriptions were too vague (such as “a man standing in front of a screen”), leading to ineffective entity recognition within the text and the resulting poor query performance. As such, the Q&A bot was primarily developed and tested using a few selected HWTGen-generated PDFs of letters. With improved ATGen results, we expect future teams to need to ensure that the Q&A bot integrates properly and performs accordingly

Generating Meaningful Alt-Text Descriptions

Generating meaningful alt-text descriptions is a relatively new development in the natural language processing world. Significant advancements have only been made as recently as 2022-2023 with the maturation of CLIP and the advent of BLIP and BLIP-2. These multimodal models have allowed for AI to better recognize objects and interpret contextual nuances, allowing for more advanced descriptions beyond basic object tagging. As a result, the code to generate such models was scant and difficult to replicate results.

We tried various other multimodal models (such as CLIP and BLIP) and image-caption datasets for fine-tuning, such as the COCO dataset (released in 2015) and various subsets of the LAION dataset (released in 2021) yielding limited results. Other multimodal models, when transferred to the context of generating descriptions for NYPL collections, yielded very poor results. We hypothesize that this is due to the dataset that they were trained on being vastly different from the collections that are contained within the NYPL dataset, which tend to be more historical in nature. Fine-tuning on other datasets also led to similar issues and results.

In addition, generating alt-text descriptions for historical documents is a very domain-specific task that very few other tasks and datasets can come close to replicating. As a result, we believe that the strongest results will only come when a historical image-caption dataset becomes available (or if NYPL librarians are able to provide descriptions for a sizable dataset for training). We advise this as a direction for future research and work, as it will yield the most promising results for the DigiSuite information extraction workflow.

Future Work

Contextual Enrichment

To improve the quality and relevance of alt-text generation, future work will focus on integrating pre-existing collection metadata into the text generation pipeline. This metadata includes collection names, time periods, and thematic information, which will provide additional context to create more descriptive and precise alt-text. By embedding these contextual details, the system aims to enhance the user experience, particularly for users who rely heavily on alt-text for understanding digital collections.

This can be especially relevant with the new LLM-ATGen model, where the addition of the LLM allows for in-context learning and additional prompt engineering. As a demonstration of the capabilities of this model, we show in Figure 12 that by altering the prompt provided to the model (without any additional fine-tuning or retraining), we can leverage the LLM's knowledge of the world to elicit more descriptive responses.



Prompt: Answer the questions. What is shown in this image?

Answer: The image shows a painting of a black swan standing on a rock in a body of water, surrounded by reeds and other plants. There are also birds flying in the background. The painting is detailed and realistic, capturing the beauty of the swan and its natural habitat. The artist has skillfully used shading and color to create a sense of depth and realism in the painting. The painting is likely a representation of the natural world, showcasing the beauty and diversity of wildlife.

Adjusted Prompt: Answer the questions. The title of this is 'Bernicla nigricans, the Black Brant' Can you describe the image and what you know about it?

Answer: The image is a vintage illustration of a black brant, a type of bird, standing on a rock near a body of water. The bird is depicted with a long neck and a black plumage. In the background, there are other birds flying in the sky and a landscape with trees and a mountain. The illustration is detailed and realistic, capturing the bird's features and

the surrounding environment. The title 'Bernicla nigricans, the Black Brant' indicates that the bird is a black brant, a species of bird known for its black plumage. The illustration is likely from a book or a publication related to ornithology or natural history. The presence of other birds in the background suggests that the black brant is in its natural habitat, which is typically coastal areas. The detailed depiction of the bird and its surroundings reflects the artist's attention to detail.

Figure 12. Results of providing NYPL document title as context to LLM-ATGen

By adding the name of the document into the prompt fed to the model, we are able to extract more meaningful descriptions based on the LLM's understanding of the world. The new description is now able to properly recognize the species of animal in the photo, the type of the environment in the background (based off it's understanding of the species) and the characteristics of the species. Metadata such as the document name, collection name, collection time period and author are all details that are currently available through NYPL APIs and we foresee that successful integration of those details with the LLM-ATGen prompts will yield even more meaningful and detailed descriptions.

Dynamic Knowledge Extraction

Establishing a feedback loop will be a key area of development. This will involve dynamically incorporating new insights, such as handwritten text extracted from tools like HWTGen, into the text generation process. These insights will enable the system to re-evaluate and regenerate alt-text as fresh data becomes available. This dynamic integration ensures that the generated alt-text remains accurate, up-to-date, and contextually relevant over time.

Multimodal Fusion

Future efforts will also explore combining multiple data inputs, including images, extracted handwritten text, and metadata, into a multimodal model pipeline. This approach will allow the system to leverage diverse data types for richer, more context-aware alt-text generation. By fusing these modalities, the system can better account for the complexities of each collection and deliver a more holistic descriptive output.

Real-Time Collection Updates

Automation will play a central role in enabling real-time updates to the collection. A pipeline will be developed to dynamically detect new collections and automatically update the system with

the latest data. This capability will ensure that all models are consistently aligned with the most current information, reducing manual intervention and improving efficiency.

Appendix

This appendix describes the changes and location of changes made to make the above work possible. Note that all of these changes exist in the ITPFA23 branch at the time of writing.

DigiSuite Classifier

The new window-level classification logic is contained within the `DigiSuiteScripts/DigiSuiteClassifier/east/EastModel.py` file, specifically in the `classify` function. This file will need to exist within the HPC directory of the individual whose credentials are being used.

For the test set evaluation, the relevant files are contained within the following directory:
/robothon/atgen/DigiSuite/DigiSuiteClassifier/evaluation

- `labels.csv`
 - This is a csv containing the filename and the manual label provided for the image. Note that if photos had some text around the margin but is not legible due to low quality, the image is labeled as ATGen. If the document includes any legible text in the image, it would be HWTGen and if it also included an image, then it would be both.
- `test_images`
 - Folder of all images that were used in the test set.
- `evaluate_digisuite_classifier.ipynb`
 - The jupyter notebook that's used for evaluation. We copy and paste the logic for classification into this notebook (and transfer the weights to colab) rather than reinstantiating through the model files provided. This method proved to be much faster and easier to perform evaluation.

NYPL_FPC and NYPL_CC are the two collections for which this new classification has been performed.

LLM-ATGen

The new code for the LLM-ATGen is found in the following directory:

`/robothon/atgen/DigiSuite/ATGen/LLM-ATGen`

- `nypl_llm_atgen_inference.ipynb`
 - notebook showing how inference was performed
- `nypl_llm_atgen.tar.gz`
 - compressed directory of all fine-tuned model weights and architecture
- `nypl_llm_atgen`
 - directory of all fine-tuned model weights and architecture
- `finetuning`
 - all files used to initialize and finetune LLaVa-Next:
 - `llm_atgen_functions.py`
 - helper functions for the initialization of the model, creation of dataset and finetuning.
 - `filtered_laion_coco_subset3464.zip`
 - zip containing all the images and captions used for fine-tuning

The inference logic has been integrated into the existing ATGen code by altering the `get_captions` function with a `get_llm_captions` function.

QA-Bot

Location:

/robothon/atgen/DigiSuite/DigiSuiteClassifier/Chatbot_NYPL.py

This file contains the core logic for interfacing with NYPL's collections via the QA Bot. It handles user input, invokes the retrieval pipeline, and returns contextually relevant answers.

ChatBot_NYPL.py

Main script that defines the conversation flow, manages user queries, interacts with vector databases or knowledge graphs, and generates final responses. It also handles special prompts such as confirming a relevant collection and follows up with the user based on their answers.

templates/

Directory holding HTML templates for rendering the chatbot UI or other front-end components. These templates are often referenced by Django views to provide consistent, styled pages.

admin.py, apps.py, forms.py, models.py, tests.py, urls.py, views.py

Standard Django files supporting the overall web application structure. These modules handle database models, form submissions, URL routing, and view logic. The QA Bot endpoints typically appear in `urls.py` and `views.py`, which route user requests to the chatbot logic.