PROLOG SECTION

Title: RAG_Processing_GeneratingNewContentAboutComputerHistory.py [Colab Notebook using LLM]

Authors: Richard H Johnson and Anna van Raaphorst-Johnson

Copyright: 2025 by VRJ Associates, LLC

Target Publishing Date: 7 Mar 2025

Goals, Objectives:

Create a short narrative description (suitable for website posting) summarizing content in a knowledge base containing content about computer history and the professional, computer-related experiences of one of the authors.

Add the llava-generated description of the IBM 650 image to the original KB (a subset of our 7_ComputerHistory knowledge base, which was described in a prior post) and summarize the contents.

Target Audience: Some Python programming experience and a general understanding of the Generative AI (GenAI) environment would be helpful.

Humans-in-the-Loop: (e.g. stakeholder(s), project manager, programmer, tester, QA)

Dick, Anna

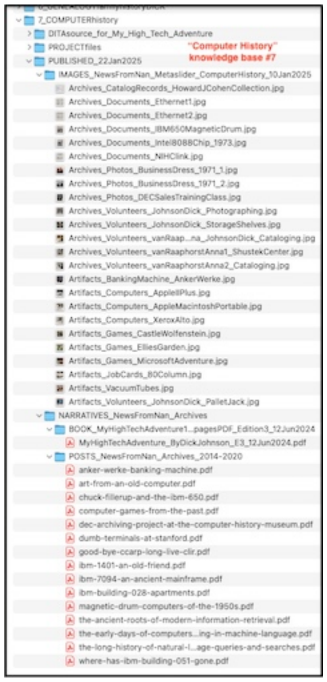
Knowledge Base Input: (e.g. formal KB, informal (other) KB, lookup on internet or elsewhere, none (rely on training knowledge))

(1) SUBSET: "7_Computer History" knowledge base (files relating to the IBM 650 computer and Dick's mentor at Service Bureau, Dr. Charles (Chuck) Fillerup). (2) ADDITION: IBM 650 document/computer generated by LLaVA running on a local computer under ollama.

BACKGROUND: Knowledge base content is stored in a temporary directory on the authors' professional website.

BACKGROUND: Items in the original "7_Computer History" Knowledge Base include 40 components in 3 categories: (1) IMAGES (24) taken by the authors as they cataloged artifacts and archives in the Computer History Museum in Mountain View, California (2) SHORT NARRATIVES (15) written in HTML and published to PDF, about computer-related artifacts and archives housed in the Computer History Museum (3) A LONGER NARRATIVE (book) that Dick had written about his experiences working in the high-technology industry.

The following image is the original knowledge base with the files of interest highlighted.


```
7_COMPUTERhistory
  DITAsource_for_My_High_Tech_Adventure
  PROJECTfiles
  PUBLISHED_22Jan2025
    IMAGES_NewsFromNan_Metaslider_ComputerHistory_10Jan2025
      Archives_CatalogRecords_HowardJCohenCollection.jpg
      Archives_Documents_Ethernet1.jpg
      Archives_Documents_Ethernet2.jpg
      Archives_Documents_IBM650MagneticDrum.jpg
      Archives_Documents_Intel8088Chip_1973.jpg
      Archives_Documents_NIHClink.jpg
      Archives_Photos_BusinessDress_1971_1.jpg
      Archives_Photos_BusinessDress_1971_2.jpg
      Archives_Photos_DECSalesTrainingClass.jpg
      Archives_Volunteers_JohnsonDick_Photographing.jpg
      Archives_Volunteers_JohnsonDick_StorageShelves.jpg
      Archives_Volunteers_vanRaap...na_JohnsonDick_Cataloging.jpg
      Archives_Volunteers_vanRaaphorstAnna1_ShustekCenter.jpg
      Archives_Volunteers_vanRaaphorstAnna2_Cataloging.jpg
      Artifacts_BankingMachine_AnkerWerke.jpg
      Artifacts_Computers_AppleIIPlus.jpg
      Artifacts_Computers_AppleMacintoshPortable.jpg
      Artifacts_Computers_XeroxAlto.jpg
      Artifacts_Games_CastleWolfenstein.jpg
      Artifacts_Games_ElliesGarden.jpg
      Artifacts_Games_MicrosoftAdventure.jpg
      Artifacts_JobCards_80Column.jpg
      Artifacts_VacuumTubes.jpg
      Artifacts_Volunteers_JohnsonDick_PalletJack.jpg
    NARRATIVES_NewsFromNan_Archives
      BOOK_MyHighTechAdventure1...pagesPDF_Edition3_12Jun2024
        MyHighTechAdventure_ByDickJohnson_E3_12Jun2024.pdf
      POSTS_NewsFromNan_Archives_2014-2020
        anker-werke-banking-machine.pdf
        art-from-an-old-computer.pdf
        chuck-fillerup-and-the-ibm-650.pdf
        computer-games-from-the-past.pdf
        dec-archiving-project-at-the-computer-history-museum.pdf
        dumb-terminals-at-stanford.pdf
        good-bye-ccarp-long-live-clir.pdf
        ibm-1401-an-old-friend.pdf
        ibm-7094-an-ancient-mainframe.pdf
        ibm-building-028-apartments.pdf
        magnetic-drum-computers-of-the-1950s.pdf
        the-ancient-roots-of-modern-information-retrieval.pdf
        the-early-days-of-computers...ing-in-machine-language.pdf
        the-long-history-of-natural-l...age-queries-and-searches.pdf
        where-has-ibm-building-051-gone.pdf
```

The following image shows the image description created by llava.

(image goes here)

RAG Processing: (e.g. model (SLM, LLM, name, version), other tools (llama-index, ollama, llava))

COMPONENTS: (1) Model: OpenAI large language model (LLM) GPT-3.5 Turbo (2) Other Tools: llama-index

This is a simple RAG chat pipeline, written in Python, that uses llama-index and runs in a Colab notebook.

ACTIONS AND RESULTS:

(1) The program includes a system prompt that provides general instructions.

(2) As it runs, the program reports its actions, using print statements.

CHAT:

(1) The model receives specific tasks and queries from a chat conversation.

(2) The model should be asked questions about the contents of the narratives in the knowledge base, to compare the contents of multiple items, and/or to generate new text drawn from all available sources.

Output: (e.g. answers to questions, summary, generated text, generated image)

Analysis, Evaluation: (e.g. met expectations? key takeaways? possible future projects?)

Double-click (or enter) to edit

KNOWLEDGE BASE SETUP SECTION

1. Set the KB variables.
2. Read in the computer history KB.
3. Read in the llava-generated image descriptions.
4. Create the relevant subset of the KB files.
5. List the files in the relevant subset.

Double-click (or enter) to edit

```
# set the knowledge base variables
KBname = "7_COMPUTERhistory_PUBLISHED_22Jan2025"
subdir = "subset"
```

```
# read in the computer history knowledge base
!rm -R {KBname}*
!wget  https://www.vrjassociates.us/KnowledgeBases/{KBname}.zip
```

```
rm: cannot remove '7_COMPUTERhistory_PUBLISHED_22Jan2025*': No such file or directory
--2025-03-08 01:41:07--  https://www.vrjassociates.us/KnowledgeBases/7_COMPUTERhistory_PUBLISHED_22Jan2025.zip
Resolving www.vrjassociates.us (www.vrjassociates.us)... 68.66.216.7
Connecting to www.vrjassociates.us (www.vrjassociates.us)|68.66.216.7|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 34515622 (33M) [application/zip]
Saving to: ‘7_COMPUTERhistory_PUBLISHED_22Jan2025.zip’

7_COMPUTERhistory_P 100%[===================>]  32.92M  7.87MB/s    in 4.8s

2025-03-08 01:41:13 (6.88 MB/s) - ‘7_COMPUTERhistory_PUBLISHED_22Jan2025.zip’ saved [34515622/34515622]
```

```
# read in the output log text file generated by LLaVa
!wget https://www.vrjassociates.us/wp-content/uploads/2025/02/llava_output_log.txt
```

```
--2025-03-08 01:41:13--  https://www.vrjassociates.us/wp-content/uploads/2025/02/llava_output_log.txt
Resolving www.vrjassociates.us (www.vrjassociates.us)... 68.66.216.7
Connecting to www.vrjassociates.us (www.vrjassociates.us)|68.66.216.7|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 16072 (16K) [text/plain]
Saving to: ‘llava_output_log.txt’

llava_output_log.tx 100%[===================>]  15.70K  76.1KB/s    in 0.2s

2025-03-08 01:41:14 (76.1 KB/s) - ‘llava_output_log.txt’ saved [16072/16072]
```

```
# expand the files in the knowledge base
!unzip -q {KBname}.zip
```

```
# list all the knowledge base files and directories
!ls -R  {KBname}
```

```
7_COMPUTERhistory_PUBLISHED_22Jan2025:
IMAGES_NewsFromNan_Metaslider_ComputerHistory_10Jan2025  NARRATIVES_NewsFromNan_Archives

7_COMPUTERhistory_PUBLISHED_22Jan2025/IMAGES_NewsFromNan_Metaslider_ComputerHistory_10Jan2025:
Archives_CatalogRecords_HowardJCohenCollection.jpg
Archives_Documents_Ethernet1.jpg
Archives_Documents_Ethernet2.jpg
Archives_Documents_IBM650MagneticDrum.jpg
Archives_Documents_Intel8088Chip_1973.jpg
Archives_Documents_NIHClink.jpg
Archives_Photos_BusinessDress_1971_1.jpg
Archives_Photos_BusinessDress_1971_2.jpg
Archives_Photos_DECSalesTrainingClass.jpg
Archives_Volunteers_JohnsonDick_Photographing.jpg
Archives_Volunteers_JohnsonDick_StorageShelves.jpg
Archives_Volunteers_vanRaaphorstAnna1_ShustekCenter.jpg
Archives_Volunteers_vanRaaphorstAnna2_Cataloging.jpg
Archives_Volunteers_vanRaaphorstAnna_JohnsonDick_Cataloging.jpg
Artifacts_BankingMachine_AnkerWerke.jpg
Artifacts_Computers_AppleIIPlus.jpg
Artifacts_Computers_AppleMacintoshPortable.jpg
Artifacts_Computers_XeroxAlto.jpg
Artifacts_Games_CastleWolfenstein.jpg
Artifacts_Games_ElliesGarden.jpg
Artifacts_Games_MicrosoftAdventure.jpg
Artifacts_JobCards_80Column.jpg
Artifacts_VacuumTubes.jpg
Artifacts_Volunteers_JohnsonDick_PalletJack.jpg

7_COMPUTERhistory_PUBLISHED_22Jan2025/NARRATIVES_NewsFromNan_Archives:
BOOK_MyHighTechAdventure1960-2024_11chapters_90pagesPDF_Edition3_12Jun2024
POSTS_NewsFromNan_Archives_2014-2020

7_COMPUTERhistory_PUBLISHED_22Jan2025/NARRATIVES_NewsFromNan_Archives/BOOK_MyHighTechAdventure1960-2024_11chapters_90pagesPD
MyHighTechAdventure_ByDickJohnson_E3_12Jun2024.pdf

7_COMPUTERhistory_PUBLISHED_22Jan2025/NARRATIVES_NewsFromNan_Archives/POSTS_NewsFromNan_Archives_2014-2020:
anker-werke-banking-machine.pdf
art-from-an-old-computer.pdf
chuck-fillerup-and-the-ibm-650.pdf
computer-games-from-the-past.pdf
dec-archiving-project-at-the-computer-history-museum.pdf
dumb-terminals-at-stanford.pdf
good-bye-ccarp-long-live-clir.pdf
ibm-1401-an-old-friend.pdf
ibm-7094-an-ancient-mainframe.pdf
ibm-building-028-apartments.pdf
magnetic-drum-computers-of-the-1950s.pdf
the-ancient-roots-of-modern-information-retrieval.pdf
the-early-days-of-computers-coding-in-machine-language.pdf
the-long-history-of-natural-language-queries-and-searches.pdf
where-has-ibm-building-051-gone.pdf
```

```
# create the relevant subset of the KB files

!rm -rf {subdir}
!mkdir {subdir}

pattern = "650"
!find . | grep {pattern} | xargs -I XX cp XX -t subset
pattern = "MyHighTechAdventure_ByDickJohnson_E3_12Jun2024"
!find . | grep {pattern} | xargs -I XX cp XX -t subset
pattern = "magnetic-drum-computers"
!find . | grep {pattern} | xargs -I XX cp XX -t subset


# add the log with the image descriptions to the subset of files
!cp llava_output_log.txt {subdir}


# list the files in the document subset
!ls -R {subdir}
```

```
subset:
Archives_Documents_IBM650MagneticDrum.jpg  magnetic-drum-computers-of-the-1950s.pdf
chuck-fillerup-and-the-ibm-650.pdf         MyHighTechAdventure_ByDickJohnson_E3_12Jun2024.pdf
llava_output_log.txt
```

RAG PROCESSING SETUP SECTION

1. Install the llama-index package.
2. Import the classes used by the script.
3. Set the OpenAI key.

```
# install the llama-index package
!pip install llama-index --upgrade --quiet
```

```
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.6/1.6 MB 41.6 MB/s eta 0:00:00
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 40.4/40.4 kB 3.1 MB/s eta 0:00:00
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 261.7/261.7 kB 16.5 MB/s eta 0:00:00
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 302.0/302.0 kB 18.0 MB/s eta 0:00:00
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.2/1.2 MB 46.3 MB/s eta 0:00:00
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 50.9/50.9 kB 3.9 MB/s eta 0:00:00
```

```
# import the necessary classes from the llama_index package
from llama_index.core import (
    VectorStoreIndex,
    SimpleDirectoryReader,
    StorageContext,
    load_index_from_storage,
    Settings,
)

from llama_index.llms.openai import OpenAI

# import package to display the LLM response in markdown
from IPython.display import display, Markdown
```

```
# set up the OpenAI key
from google.colab import userdata
import os

open_ai_key = userdata.get('open_ai_key')

os.environ["OPENAI_API_KEY"] = open_ai_key
```

FUNCTION DEFINITION SECTION

Definition: chat_with_model_loop() - the main chat loop

```
# function to handle user chat queries in a loop
def chat_with_model_loop():
    print()
    print("Chatbot: Hello! How can I assist you today?")
    print("  Type 'exit' to end the conversation")
    print("  Type 'restart' to start a new conversation")

    while True: # the main chat loop
        print()
        # read the user query from the console
        user_input = input("You: ")
        print()

        print("Your query was:")
        display(Markdown(user_input))
        print()

        # stop the chat if the user entered "exit"
        if user_input.lower() == 'exit':
            print("Bot: Goodbye! Have a great day!")
            print()
            break

        # restart the conversation if the user entered "restart"
        if user_input.lower() == 'restart':
            print("Bot: restarting the chat")
            chat_engine.reset()
            continue

        # Send the query to the language model and get back a response.
        #
        # In summary, chat_engine.chat() performs the following functions:
        #
        #  1) It uses RAG to fetch relevant context from the knowledge base
        #     ensuring responses are based on indexed data rather than
        #     external sources
        #  2) The chat engine retrieves text from the knowledge base and
        #     uses it as context sent to the model for generating reponses

        response = chat_engine.chat(user_input)

        # display the response generated by the model
        # display the bot response in a readable format
        print("The response is:")
        print()
        display(Markdown(response.response))
```

PROCESSING INITIALIZATION SECTION

1. Set the location of knowledge base (KB) ("Computer History").
2. Create the chunked documents using the KB as information source.
3. Create a vector store index from the chunked documents.
4. Set a system prompt for the chat.
5. Create a chat query engine.

```
# set the location of the knowledge base
datadir = subdir
print ("Setting the location of the local knowledge base to:",datadir)
```

```
Setting the location of the local knowledge base to: subset
```

```
# load the knowledge base files, allowing specific filetypes to be scanned,
# and split them into annotated chunks (called "documents")
documents = SimpleDirectoryReader(datadir,required_exts=[".txt",".pdf"]).load_data()
print ("Loading knowledge base files and chunking them into documents")
```

```
Loading knowledge base files and chunking them into documents
```

```
print(len(documents),"chunks created")
```

```
96 chunks created
```

```
# create a vector store index from the documents to be used in a similarity search
# between the user query and the document chunks in the vector store.
index = VectorStoreIndex.from_documents(documents)
print ("Creating a vector store index from the chunked documents")
```

```
Creating a vector store index from the chunked documents
```

```
# set a system prompt for the chat
system_prompt_text = """
You are an archivist who answers research questions from researchers. You need
help in locating archived images and documents that will help you do that task.
Sometimes you ask the model to generate text on your behalf.

Don't make up an answer; if you don't know the answer to a question,
just say that you don't know. Provide the response in Markdown format.

"""
print ("Creating a system prompt")
```

```
Creating a system prompt
```

```
# create a RAG chat engine
#
# creates a chat engine from an index
#
# retrieves relevant data
# from the index to provide context for the model responses
#
# setting chat_mode = "context" restricts the model to using
# only the RAG context to generate a response
#
#
# set the value of the LLM model we want to use and also temperature
language_model_name = "gpt-3.5-turbo"
llm = OpenAI(model=language_model_name, temperature = 0.7)

chat_engine = index.as_chat_engine(
    llm=llm,
    chat_mode="context",
    system_prompt=system_prompt_text)
print ("Creating a chat engine")
```

```
Creating a chat engine
```

PROCESSING SECTION

Starts a chat session. In this program, the expectation is that the user will ask the model to create a summary, based on what is in the KB and the user's specific requirements (e.g. length of response). The user can refine the request based on the model's initial response. The entire dialog is maintained as content until the user types "restart," at which point it starts a completely new session.

```
print("Start a conversation with",language_model_name)

# Start the chat RAG processing with the SLM
chat_with_model_loop()
```

```
Start a conversation with gpt-3.5-turbo

Chatbot: Hello! How can I assist you today?
  Type 'exit' to end the conversation
  Type 'restart' to start a new conversation

You: Please summarize the input into a post-size document that I could put on a website. Include a little background on Dick

Your query was:
```

Please summarize the input into a post-size document that I could put on a website. Include a little background on Dick Johnson and his career history, something about his relationship with Dr. Charles Fillerup, and Dr. Fillerup's experiences with the IBM 650. Also include a short description of the IBM 650 itself.

```
The response is:
```

Post Summary:

Background on Dick Johnson:

Dick Johnson worked at IBM Service Bureau Corp after graduating from UCLA in 1965. His mentor at the time was Dr. Charles Fillerup, who played a significant role in Johnson's programming education and career development. Johnson and Fillerup collaborated on various projects, and Fillerup shared valuable insights and stories about programming challenges, particularly related to the IBM 650.

Dr. Charles Fillerup and His Experiences with the IBM 650:

Dr. Charles Fillerup, a mentor to Dick Johnson, faced challenges programming the IBM 650 due to its unique memory structure—a rotating drum instead of core memory. Fillerup's experiences highlighted the importance of efficient program execution to minimize waiting time for drum rotations. His expertise and stories about programming intricacies provided valuable lessons to Johnson and others in the field.

Description of the IBM 650:

The IBM 650, introduced in 1953, was a landmark computer that used a magnetic drum for memory storage instead of core memory. This design choice posed programming challenges, as efficient code execution was crucial to optimize performance and reduce waiting time for drum rotations. The IBM 650's presence in the Computer History Museum's collection serves as a reminder of the evolution of computing technology and the pioneering work of individuals like Dr. Charles Fillerup.

For more detailed information, you can refer to the full article on Chuck Fillerup and the IBM 650 by Dick Johnson on the News from Nan website.

```
You: exit

Your query was:
```

exit

```
Bot: Goodbye! Have a great day!
```

EXIT SECTION

```
# Exit the script
print("Exiting")
```

```
Exiting
```