DEV1351C
Harnessing Multimodal Data for Enhanced Splunk Analytics - Lab Exercises

Overview

This lab provides hands-on training in using Splunk, Python, and Ollama, which can run large language models (LLMs) locally.
Key learning points include:

- Setup and Configuration: installation of Ollama and LLM models, adding the LLM Parser Splunk Add-on, configuring the Add-on and environment.
- Python Scripting and LLMs: Participants will learn about Splunks Python SDK and leveraging LLMs for image-to-text conversions and explore the basics of LLMs.
- Image Processing: Techniques for extracting meaningful information from images, converting images to text using LLMs, and integrating this process within Splunk.
- Analysis Techniques: Utilize Splunk streaming commands in SPL to analyze data using LLMs.
- Real-World Scenarios: Exercises based on real-life cases, according to Buttercup.

Supplemental Materials: All participants will have access to a GitHub repository containing a custom Splunk Add-on app and sample files for practice in the labs, enhancing the learning experience. It is assumed that each participle has Splunk Enterprise 9 running locally they can modify.

Lab Exercise 1 - Getting to Know Ollama

Description

In this hands-on lab, participants will install and configure Ollama, an application for running large language models. This exercise includes:

- *Installation: Guidance on downloading and installing Ollama from Ollama.com, available for macOS, Linux, and Windows.*
- *Introduction to Ollama usage: Learn to navigate Ollama's commands, such as pull, list, and run, which are essential for managing models.*
- *Running a Model: Participants will pull the latest llama3 model, verify its installation, and execute it to engage with the model interactively.*
- *Interactive Prompts: Explore how to input multiline prompts and retrieve responses, enhancing understanding of dynamic model interaction.*
- *Troubleshooting and Feedback: Troubleshooting tips for common issues and a mechanism for feedback to facilitate continuous learning and support.*

This lab is designed to give participants a thorough introduction to using Ollama, empowering them with the skills to integrate and leverage large language models in their projects.

Steps

1. Download and install Ollama from the following location. https://ollama.com/ Ollama is available for macOS, Linux and Windows.
2. Open a command prompt or terminal.
    a. On Windows, search for “cmd” or “Command Prompt” in the Start menu.
    b. On macOS, you unzip, then move it to the applications folder.
    c. On macOS or Linux, open “Terminal”.
3. Check Installation
    a. Type the command
    `ollama --version`

b. Press Enter. This command should return the version number of Ollama installed on your system, confirming that it is installed correctly.

4. Review Usage - See Appendix for usage
   a. Type the command
      `ollama`
   b. Press Enter. This command should return the usage information of Ollama.
5. Download the “llama3” model
   a. Type the command
      `ollama pull llama3`

      Note: Pulling a model with the ollama pull command may take some time, depending on your internet speed and the size of the model.
   b. This command downloads and adds the llama3 model to your Ollama installation.
   c. This file is over 4.7GB and may take a few mins
6. List the installed model.
   a. Type the command
      `ollama list`
   b. This command should show which models are installed.
7. Try running a model
   a. Type the command
      `ollama run llama3`
   b. You should see a prompt line “>>>”
   c. Type the following at the prompt
      `/?`
   d. You should see the available commands
   e. Type the following prompt
      `What is Splunk?`
   f. Type the following prompt
      `“””`
      `What is`
      `Splunk?`
      `“””`
   g. Type the following prompt to exit ollama.
   h. `/bye`

> Note: We will need the llava LLM model for a later lab. Lets go ahead and start the download now. Ollama supports multitasking, so while it's downloading we can proceed in a new terminal.

8. Download the "llava" model
    a. Type the command `ollama pull llava`
    b. This command downloads and adds the llama3 model to your Ollama installation.
    c. This file is over 4.1GB and may take a few mins

Summary

In this engaging lab, participants learned to install and configure Ollama, an application for running large language models. They navigated essential commands like pull, list, and run, simplifying model management. A key activity was pulling and interacting with the llama3 model, including handling multiline prompts to enhance their understanding of dynamic interactions. Additionally, participants received troubleshooting tips and initiated the download of the llava model for future use.

Lab Exercise 2 - Creating a Prompt

Description

This lab teaches participants to craft effective prompts using Ollama. It covers four main areas:

- *Clarity is Key: Learn the impact of detail in prompts through practical examples.*
- *Understand the User Needs: Explore how specifying output formats, like JSON, affects responses.*
- *Refining the Prompt: Refine prompts to include all necessary details for comprehensive responses.*
- *Optimizing Future Prompts: Develop skills to generate optimized prompts for future use.*

Participants will gain proficiency in prompt engineering, crucial for leveraging LLMs in various applications.

Steps

1. Start Ollama
    a. Type the command `ollama run llama3`
2. Compare the following prompts
    a. At the prompt type `Provide information about Buttercup.`
    b. This prompt may reply with information about Buttercup, Splunks mascot. Most likely not.
    c. Update the prompt to mention your talking about Splunks Mascot `Buttercup is a horse mascot for Splunk. Provide information about Buttercup.`
3. Let's improve this prompt even more.

```
Buttercup is a horse mascot for Splunk. Provide
information about Buttercup. Detail some
characteristics about Buttercup.
```

   a. This prompt will provide a more detailed result.

4. Next let's provide a prompt that will create a properly formatted result.

```
Buttercup is a horse mascot for Splunk. Provide
information about Buttercup. Detail some
characteristics about Buttercup.  Output the results
in JSON format for easy integration.
```

   a. This prompt should contain a JSON object.

5. Now, lets ask it to reformat our prompt and include a JSON Template

```
Help me create a prompt that will respond with the
previous results with a JSON template for proper
formatting, including placeholders for JSON values.
```

   a. This will refactor our existing prompt and give a JSON Template for future use

> Note: Your results may be slightly different and OK. Review the prompt and continue.

6. Now we can run this LLM optimized prompt ensuring consistent results.

```
“””
Create a JSON response for Buttercup, the horse mascot
of Splunk. The JSON should include the following
characteristics:
* Name (string): [Buttercup]
* Species (string): [Horse]
* Personality (array of strings):
    + [Friendly]
    + [Approachable]
    + [Enthusiastic]
* Appearance (object):
    + Color (string): [Yellow]
    + Mane (string): [White]
```


```
    + Tail (string): [White]
* Symbolism (array of strings):
    + [Joy of working with data]
    + [Importance of having fun while doing it]

**JSON Template:**

```json
{
  "name": "${Buttercup}",
  "species": "${Horse}",
  "personality": [
    "${Friendly}",
    "${Approachable}",
    "${Enthusiastic}"
  ],
  "appearance": {
    "color": "${Yellow}",
    "mane": "${White}",
    "tail": "${White}"
  },
  "symbolism": [
    "${Joy of working with data}",
    "${Importance of having fun while doing it}"
  ]
}
```

Placeholders:

* `${Buttercup}`: Replace with the actual value for Buttercup's name.
* `${Horse}`: Replace with the actual value for the species (which is Horse, in this case).
```

```
* `${Friendly}`, `${Approachable}`, and `${Enthusiastic}`: Replace with the actual values for personality traits.
* `${Yellow}`, `${White}`: Replace with the actual values for appearance color and mane/tail.
* `${Joy of working with data}` and `${Importance of having fun while doing it}`: Replace with the actual values for symbolism.
"""
```

7. Close out Ollama.
    a. Run the following command to stop Ollama
    `/bye`

Summary

In this lab, participants learned to create effective prompts for LLMs, focusing on clarity, understanding user needs, refining details, and optimizing future prompts. They practiced starting Ollama, comparing different prompts for the mascot Buttercup, refining these prompts for detail and format, and ultimately creating a JSON template for consistent and comprehensive responses. This lab equips participants with essential prompt engineering skills to leverage large language models effectively in various applications.

Lab Exercise 3 - Can it Splunk?

Description
This lab demonstrates how Large Language Models (LLMs) can be utilized to streamline and enhance various Splunk functions, making administration tasks more efficient.

- *Writing Regular Expressions: Participants will use LLMs to create regular expressions for parsing and extracting data from complex log files.*

- *Writing SPL Queries: Participants will learn how to use LLMs to generate complex SPL queries for specific analytical needs, such as monitoring network traffic or user activities.*

By the end of this lab, participants will be proficient in leveraging LLMs to enhance their Splunk environment, reducing the complexity and time spent on routine tasks.

Sample Data

2022-07-15 10:42:03 INFO User called from 321-456-7890 regarding account issues.

2022-07-15 10:45:21 ERROR Connection attempt from 800.555.1234 was blocked.

2022-07-15 10:49:58 DEBUG Incoming call at 2121234567 recorded in the system."system."issues.

FIX - Convert buttercup theme and image (ocr) with regex
Steps

1. Type the command
   `ollama run llama3`
2. At the prompt type
   `"""`


```
Write a regular expression that will match on the
phone number. It should match all three versions of
the phone number (321-456-7890, 800.555.1234,
2121234567).

Sample data:
"2022-07-15 10:42:03 INFO User called from
321-456-7890 regarding account issues.
2022-07-15 10:45:21 ERROR Connection attempt from
800.555.1234 was blocked.
2022-07-15 10:49:58 DEBUG Incoming call at 2121234567
recorded in the system."system."issues."
"""
```

3. Open the website regex101 at https://regex101.com/
    a. Copy the “Sample Data” above into the “TEST STRING” window
    b. Copy the response from step 2 into the “REGULAR EXPRESSION” window
    c. Verify that each of the phone numbers are matched.

> Note: If not all phone numbers are matched. Follow up with an additional prompt like:
> ”It didn't match on the phone number 2121234567. Try again.”

4. Close out Ollama.
    a. Run the following command to stop Ollama
    `/bye`

Lab Exercise 4 - A picture is worth a thousand words.

Description

This lab teaches participants to use the LLava: Large Language and Vision Assistant multimodal modal running in Ollama for processing and describing images, refining prompts to return results in JSON format. Key activities include:.

- *Download and run the 'llava' Large Language and Vision Assistant model.*
- *Process sample data images.*
- *Refine prompts to generate detailed JSON responses for image descriptions and specific details.*

By the end of this lab, participants will be proficient with processing images and refining prompts to generate detailed JSON responses.

Sample Images

		
buttercup_go.jpg	buttercup_id.jpg	weather_map.jpg

Steps

1. Type the command
   `ollama run llava:latest`
2. At the prompt type
   `./sample_data/buttercup_go.jpg`
3. The response should mention the following, “Added Image ‘./sample_data/buttercup_go.jpg”, then provide a description of the image.
4. FIX
5. Lets refine our prompt to return JSON.
   `./buttercup_go.jpg  Your response should be in JSON`
6. The reply should be in JSON format.
7.
8. At the prompt type
   `./sample_data/buttercup_id.jpg`
9. The response should mention the following, “Added Image ‘./sample_data/buttercup_id.jpg”, then provide a description of the image.
10. Lets refine our prompt to return JSON.
    ```
    ./buttercup_id.jpg  What are the details in this image? Only reply with information in the image. I only need a name and address.
    Your response should be in JSON
    ```
11. The response should mention the following, “Added Image ‘./sample_data/buttercup_id.jpg”, then provide a description of the i
12. At the prompt type
    `./sample_data/weather_map.jpg`
13. The response should mention the following, “Added Image ‘./sample_data/weather_map.jpg”, then provide a description of the image.
14. Lets refine our prompt to return JSON.
    ```
    ./weather_map.jpg what cities are on the map?  What are their temperatures?
    Response should be in JSON format
    ```
15. The response should mention the following, “Added Image ‘./sample_data/weather_map.jpg”, then provide details on the cities and temperatures.

Summary

In this lab, participants learned to use the LLava (Large Language and Vision Assistant) multimodal model running in Ollama for processing and describing images, refining prompts to return results in JSON format. Key activities included downloading and running the llava model, processing sample images (buttercup_go.jpg, buttercup_id.jpg, and weather_map.jpg), and refining prompts to generate detailed JSON responses for image descriptions and specific details. By the end of the lab, participants will be proficient in processing images and crafting prompts to produce structured JSON outputs.

Lab Exercise 5 - What's happening?

Description

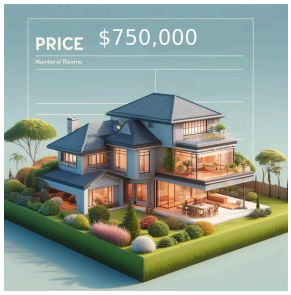
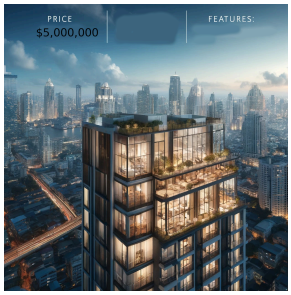
This lab demonstrates how we can leverage Splunk modular inputs to call and pass data to LLMs. This provides a way to automate processing of multimodal inputs.

- *Walk through installing the LLM_Parser Splunk Add-on.*
- *Configure the LLM_parse app to process sample data.*
- *Review processed data and evaluate the data*

By the end of this lab, participants will be proficient in leveraging LLMs to enhance their Splunk environment, reducing the complexity and time spent on routine tasks.

FIX - command line ollama, then data input, then Splunk search

Sample Images

			
house_barn.jpg	house_suburbs.jpg	house_city.jpg	house_stable.jpg

Steps

1. Download the Splunk app from https://github.com/rbarber68/DEV1351C
    a. You can use git to clone the files locally
    `git clone https://github.com/rbarber68/DEV1351C.git`

2. Move the llm_parser directory into the Splunk app folder
    a. The Splunk app folder is located on your local machine ($SPLUNK_HOME/etc/apps/)
    b. You can copy the folder using the following
    `cd DEV1351C`
    `cp -R llm_parser $SPLUNK_HOME/etc/apps`
3. FIX Only support Zips
4. Install Splunk SDK Python
    a. Using a terminal or cmd window, use pip to install the SDK
    `pip install splunk-sdk --target=`$SPLUNK_HOME`/etc/apps/llm_parser/lib`
    b. Optionally install from SDK manually
    c. Download the Splunk SDK from https://github.com/splunk/splunk-sdk-python
    d. Download the .zip or using git clone the repo locally.
        i. Git clone https://github.com/splunk/splunk-sdk-python.git
    e. Copy the splunklib folder into the llm_parser/lib folder (step 2b)
    f. It should look like the following
    $SPLUNK_HOME/etc/apps/llm_parser/lib/splunklib
5. You should restart Splunk
    `$SPLUNK_HOME/bin/splunk restart`
6. Let's review the sample images to create a good prompt
7. Type the command
    `ollama run llava:latest`
8. At the prompt type
    `./house_suburbs.png`
9. The response should mention the following, "Added Image './house_suburbs.png", then provide a description of the image.
10. Lets refine our prompt to return JSON.
    `./house_suburbs.png Respond in JSON. How much is this house?`
11. Go into Splunk, click on "Settings", in the top right
12. Under DATA, click on "Data Inputs"
13. Scroll thru the list of "Local inputs", click on "LLM Parser".
14. Click on the "New" Green Button in the top right.

15. FIX $Splunk_home
16. Enter the following:

```
Name:House Prices
Monitoring Folder:
$SPLUNK_HOME/etc/apps/llm_parser/appserver/static/images/ho
use*.png
Model Provider:      ollama
Model Location:      http://localhost:11434/api/generate
Model Name:          llava
Prompt:              Respond in JSON. How much is this
house?
```

    a. Click “Next”
    b. Click “Start Searching” button
    c.
17. From a Splunk search run the following

```
index=main sourcetype=llm_parser
```

18. The response should mention the following, “Added Image ‘./sample_data/buttercup.games.jpg”, then provide a description of the image.
19. Lets refine our prompt to return JSON
20.

Lab Exercise 6 - SPL Integration

This lab demonstrates how we can leverage Splunks streaming commands to call LLMs using SPL. We will look how SPL can pass arguments to LLMs. This provides a way to interactively call LLMs from SPL.

- *Walk thru the llm_prompt python code.*
- *Review how to pass arguments to the LLM*
- *Review returned values from the LLM.*

By the end of this lab, participants will be proficient in leveraging LLMs to enhance their Splunk environment, reducing the complexity and time spent on routine tasks.

Steps

1. This lab assumes that you've installed the LLM_Parser Splunk Add-on from Lab 5.
2. Verify the SPL is installed.
    a. Click on "Apps" dropdown, on the top left of a Search
    b. Verify the app "LLM Parser" is in the list of apps
3. In a new search window type the following

```
| llmprompt model=llama3 prompt="what is Splunk's mascot Buttercup favorite food?"
```

    a. The result should be a single line with a response.
    b. Review the response
4. Review the parameters
    a. Model: The name of the ollama model you wish to call "llama3"
    b. Prompt: The prompt info to run.
5. Next will wll look at SPL Streaming command that will allow use to pipe results into the LLM for processing.
6. In a new search window type the following

```
index=_internal log_level
| head 1
| llmstream evalfieldname=log_level prompt="translate the following into Imaginary horse Language.  Only
```

```
reply with horse speak" model=llama3
fieldname=llm_response_horse
| llmstream evalfieldname=log_level prompt="translate
the following into Spanish.  Only reply with the
Spanish translation" model=llama3
fieldname=llm_response_spanish
```

a. Review the response

Appendix

Ollama usage info

```
$ ollama
Usage:
  ollama [flags]
  ollama [command]

Available Commands:
  serve       Start ollama
  create      Create a model from a Modelfile
  show        Show information for a model
  run         Run a model
  pull        Pull a model from a registry
  push        Push a model to a registry
  list        List models
  ps          List running models
  cp          Copy a model
  rm          Remove a model
  help        Help about any command

Flags:
  -h, --help      help for ollama
  -v, --version   Show version information

Use "ollama [command] --help" for more information about a command.
```

Summary

In this lab, we delved into … We discovered how to …. These exercises actualize the activities our security customers conduct in their real-world environments, giving us …

Happy Splunking!