DAY 1 – LLMS & PROMPTING - COMPLETE WORKSHOP GUIDE

Date: Day 1 of watsonx Workshop
Duration: 8 hours (4 hours theory + 4 hours labs)
Track: Core/Granite

WORKSHOP SCHEDULE

MORNING SESSION (4 HOURS) - THEORY

Time	Duration	Topic	Description
9:00 - 9:15	15 min	Welcome & Setup Check	Verify Day 0 completion
9:15 - 10:30	75 min	**1.0 LLM Concepts**	Core concepts, local vs managed, architecture
10:30 - 10:45	15 min	Break	
10:45 - 11:45	60 min	**1.2 Prompt Patterns**	Patterns, templates, best practices
11:45 - 12:00	15 min	Q&A	
12:00 - 1:00	60 min	Lunch	

AFTERNOON SESSION (4 HOURS) - LABS

Time	Duration	Topic	Description
1:00 - 1:45	45 min	**Lab 1.1**	Quickstart in both environments

LEARNING PATH

```
Day 0 (Prerequisites)
    ↓
Day 1 Morning: Theory
    → 1.0 LLM Concepts
    → 1.2 Prompt Patterns
    ↓
Day 1 Afternoon: Labs
    → Lab 1.1: Quickstart
    → Lab 1.2: Templates
    → Lab 1.3: Evaluation
    ↓
Day 2: RAG (Retrieval-Augmented Generation)
```

MATERIALS PROVIDED

THEORY DOCUMENTS

1. `llm-concepts.md` - Core LLM concepts and architecture
2. `prompt-patterns-theory.md` - Prompt engineering patterns
3. `eval-safety-theory.md` - Evaluation and safety

LAB INSTRUCTIONS

1. `lab-1-quickstart-two-envs.md` - Lab 1.1 guide
2. `lab-2-prompt-templates.md` - Lab 1.2 guide
3. `lab-3-micro-eval.md` - Lab 1.3 guide

NOTEBOOKS (TO BE CREATED BY PARTICIPANTS)

1. `ollama_quickstart.ipynb` - Ollama experiments
2. `watsonx_quickstart.ipynb` - watsonx.ai experiments
3. `prompt_patterns_ollama.ipynb` - Ollama templates
4. `prompt_patterns_watsonx.ipynb` - watsonx templates
5. `micro_evaluation.ipynb` - Evaluation framework

REFERENCE MATERIALS

- `labs-src/` - Reference RAG notebooks

LEARNING OBJECTIVES BY MODULE

1.0 LLM CONCEPTS

- ✅ Understand tokens, context windows, parameters
- ✅ Compare local vs managed deployments
- ✅ Know cost and resource considerations
- ✅ Understand accelerator architecture

1.2 PROMPT PATTERNS

- ✅ Recognize common prompt patterns
- ✅ Build reusable templates
- ✅ Apply prompt engineering principles
- ✅ Design production prompts

LAB 1.1: QUICKSTART

- ✅ Run prompts in Ollama and watsonx
- ✅ Modify parameters (temperature, max_tokens)
- ✅ Compare outputs and latency
- ✅ Connect to accelerator

LAB 1.2: TEMPLATES

PREREQUISITES CHECKLIST

Before starting Day 1, ensure:

- ✅ Day 0 completed
- ✅ `simple-ollama-environment` working
- ✅ `simple-watsonx-enviroment` working with credentials
- ✅ Jupyter accessible in both environments
- ✅ Ollama has at least one model pulled (e.g., `qwen2.5:0.5b-instruct`)
- ✅ watsonx.ai credentials verified (API key, URL, project ID)
- ✅ `watsonx-workshop` repo cloned (for accelerator reference)

KEY CONCEPTS SUMMARY

TOKENS

- Sub-units of text that LLMs process
- ~4 characters per token (English average)
- Context window = max tokens (input + output)

TEMPERATURE

- 0.0 = Deterministic, focused
- 0.7-1.0 = Balanced
- 1.5+ = Creative, unpredictable

PROMPT PATTERNS

1. **Instruction**: Direct command
2. **Few-shot**: Examples before task
3. **Chain-of-thought**: Step-by-step reasoning
4. **Style transfer**: Rewrite in different tone
5. **Summarization**: Condense content

EVALUATION SIGNALS

1. **Correctness**: Matches ground truth?

INSTRUCTOR NOTES

MORNING SESSION TIPS

- **LLM Concepts**: Use diagrams for architecture
- **Prompt Patterns**: Live demo with watsonx Prompt Lab
- Keep theory interactive with questions
- Relate concepts to students' use cases

AFTERNOON SESSION TIPS

- **Lab 1.1**: Ensure all students complete before moving on
- **Lab 1.2**: Encourage creativity in template design
- **Lab 1.3**: Form small groups for evaluation discussions
- Circulate during labs to answer questions
- Have backup notebooks ready for students with issues

COMMON ISSUES

1. **Ollama not running**: Check Docker container or service
2. **watsonx 401 errors**: Verify credentials in `.env`
3. **Rate limits**: Remind students to pace requests
4. **Python environment**: Ensure correct kernel selected

SUCCESS CRITERIA

By end of Day 1, students should be able to:

1. **Explain** how LLMs work at a high level
2. **Compare** local and managed LLM deployments
3. **Write** effective prompts for different tasks
4. **Build** reusable prompt templates in Python
5. **Evaluate** LLM outputs systematically
6. **Run** notebooks in both Ollama and watsonx environments
7. **Understand** how LLMs fit into the accelerator architecture

HOMEWORK (OPTIONAL)

1. **Expand test set**: Add 10 more diverse prompts to Lab 1.3
2. **Try different models**:
   - Ollama: `llama3.2:3b`, `qwen2.5:1.5b`
   - watsonx: Try different Granite variants
3. **Advanced prompting**: Implement a multi-turn conversation pattern
4. **Read ahead**: Review Day 2 materials on RAG concepts

CONNECTIONS TO FUTURE DAYS

DAY 2 (RAG)

- Today’s prompts → prompts with retrieved context
- Single LLM call → retrieval + LLM pipeline
- Manual evaluation → automated RAG metrics (retrieval quality, answer quality)

DAY 3 (AGENTS & ORCHESTRATION)

- Static prompts → dynamic tool-calling prompts
- Single-turn → multi-turn conversations
- Basic evaluation → production monitoring

RESOURCES

DOCUMENTATION

- IBM Granite Models
- watsonx.ai Docs
- Ollama Docs

PROMPT ENGINEERING

- OpenAI Prompt Guide
- Granite Prompting Guide

COMMUNITY

- IBM Granite GitHub
- watsonx Community

FEEDBACK & QUESTIONS

DURING WORKSHOP

- Use chat/Slack for quick questions
- Raise hand for blocking issues
- Share interesting findings with the group

AFTER WORKSHOP

- Complete feedback survey
- Share lab solutions with peers
- Join community discussions

DAY 1 COMPLETION CHECKLIST

Theory: - ✅ Attended 1.0 LLM Concepts session - ✅ Attended 1.2 Prompt Patterns session - ✅ Attended 1.3 Evaluation & Safety session

Labs: - ✅ Completed Lab 1.1 (Quickstart) - ✅ Completed Lab 1.2 (Templates) - ✅ Completed Lab 1.3 (Evaluation)

Deliverables: - ✅ Working notebooks in both environments - ✅ Prompt templates created - ✅ Evaluation results CSV generated

Understanding: - ✅ Can explain LLM concepts - ✅ Can write effective prompts - ✅ Can evaluate LLM outputs - ✅ Ready for Day 2 (RAG)

NEXT: DAY 2 PREVIEW

Tomorrow we'll: 1. Add **retrieval** to our LLM calls (RAG) 2. Integrate with the **accelerator** codebase 3. Build a **production-ready** RAG service 4. Learn about **vector databases** and **embeddings**

Prepare by: - Reviewing today's materials - Ensuring accelerator code is accessible - Thinking about documents you'd like to use for RAG

Congratulations on completing Day 1! 🎉

You’ve built a strong foundation in LLM fundamentals and prompt engineering. Tomorrow, we’ll take it to the next level with RAG.