NML Language Model Lab

Hands-On LLM Prompting

—— ✂ **Page 1** ✂ ——

Mad-Lib Warm-Up

A short, playful tour of language-model concepts that you can reuse in research workflows

Term-Bank (for the Mad-Lib below)

`API · Batch Prompting · Model · Prompt · Prompt Engineering · Role Assignment · System Message · Temperature · Top-p · Token · Training Data`

Mad-Lib: Anatomy of a Prompt

> A(n) [1] begins with a [2] that tells the [3] which role to play.

> By lowering [4] or [5], you shrink the model's creative range.

> Behind the scenes, the request travels through an [6] to the chosen [7], which slices your words into [8] and consults its [9] to predict continuations.

> Gym class heroes speed things up with [10].

Mad-Lib: Corresponding Answers

1.
2.
3.
4.
5.
6.
7.
8.
9.
10.

—— ✂ **Page 2** ✂ ——

AI-LLM Glossary (Spoiler-Free)

API: The doorway software uses to talk to a model – like a menu filled with rules instead of dishes.

Batch Prompting: Sending many related prompts at once to compare outputs

Prompt: The overt instruction you write to models

Prompt Engineering: Refining that instruction for better outcomes

Training Data: Massive text and/or image datasets that serve as the knowledge base for AI models

Role Assignment: Stating *who* or *what* the model should pretend to be (e.g., "You are a skeptical archivist")

System Message: A backstage note that quietly enforces rules and persona

Temperature: A knob that raises or lowers output spontaneity

Top-p: A filter that limits randomness to the most-likely words

Token: A tiny chunk of text the model actually processes (often sub-word)

LLM (Large Language Model): A huge neural network that predicts the next word and adapts to new tasks from a prompt alone

Embedding: A dense numeric vector that places text in "semantic space" so closeness ≈ similarity

Transfer Learning: Re-using a pre-trained model's knowledge to jump-start a new task

Fine-Tuning: Continuing training on your own labelled examples to specialize the model

RAG (Retrieval-Augmented Generation): Fetches facts first and feeds them to a language model so it remains grounded in real source material

RLHF: "Reinforcement Learning from Human Feedback" – humans rate outputs so the model prefers helpful ones

OpenRouter Sandbox Guide

Scanning opens the provisioned OpenRouter chat.

API Provider: Fields & Defaults

1. Go to "Settings"

2. Click "Model Provider" then "Add Custom Provider"
3. **Name**: `openrouter`
4. **API Host**: https://openrouter.ai/api/v1/chat/completions
5. **API Key**:

```
┌──────────────────────────┐
│   ▒▒  QR-CODE HERE  ▒▒   │
└──────────────────────────┘
```

6. **Model**: Choose or paste an ID (see p. 4)
7. **Max Tokens in Context**: Optional cap. Prevents runaway length and saves credits
8. **Temperature / Top-p**: 0.2–1.0 sliders (lower values → crisp summaries; higher → creative drafts)
9. Click **Save**

Prompting Exercises

Task 1: Document Completion Task

1. Paste a paragraph from a primary source
2. Prompt:

```
Complete the incomplete paragraph in the following excerpt from a primary
source:
[insert-excerpt]
```

3. Repeat with Temperature 0.2 vs 0.8 and note changes

Task 2: Document Understanding

1. Find the image of a historical document
2. Enter one of these models:
   - `openai/gpt-4o`
   - `anthropic/claude-3-sonnet`
   - `google/gemini-pro-vision`

    - `mistralai/pixtral-12b`
3. Edit the system message for the task:

```
You are an advanced OCR processing tool  for parsing and transcribing historical materials with mixed media, multiple formats, and/or challenging handwriting.
```

3. Paste the next "seed prompt" and attach your image:

```
Transcribe the attached image of the document with alt-text for mixed media or embedded images, filling as Dublin Core fields where present and grounded in the input file, which the user will provide.
```

Ready-to-Try Model IDs

`openai/gpt-4o` : Flagship large language model from OpenAI

`anthropic/claude-3-sonnet` : Flagship large language model from Anthropic

`google/gemini-pro-vision` : Vision language model for extracting images + text together from Google

`mistralai/pixtral-large-2411` : Vision language model built on top of `mistral-large-2411` from MistralAI

`mistralai/pixtral-12b` : Fast, lightweight vision language model for documenting understanding from Mistral AI