Class 16: Model Selection

- (Local) Text Model Selection
- App Design for LLM Context
- Remainder of Semester Agenda

Copyright 2026, Kyle J. Harms

Review: Website vs. App Design Patterns

Website Design Patterns	Mobile App Design Patterns
Logo in top left corner	Logo for the app's icon
Navigation bar at the top of the page (scrolls away)	Navigation panel **fixed** to the bottom of the screen
Content is centered with wide margins	Content fills the entire screen
Headings are used to organize content	Headings are rarely used; content is organized with cards, lists, etc.
Links are used to navigate to other pages or sections	Buttons are used to perform actions or navigate to other screens
Focuses on information/explanation	Focused on the functionality and the user's tasks

(Local) Text LLM Models

Model Selection

When selecting a model for your application, consider the following factors:

- **Model Size:** Larger models generally have better performance but require more computational resources.
- **Training Data:** The quality and diversity of the training data can impact the model's performance and the types of responses it generates.
- **Inference Speed:** The time it takes for the model to generate a response can be important for user experience, especially in interactive applications.
- **Specialization:** Some models are trained for specific tasks or domains, which can improve performance in those areas.
- **Context Length:** The amount of context the model can consider when generating responses can affect its ability to provide relevant and coherent answers.

WebLLM Models for Tasks

Model	Possible Use Cases (Your mileage may vary)
Llama 3.2 / 3.1	Instruction-following, summarization (class default: Llama 3.2 1B)
Qwen 3 / 3.5	Multilingual, general-purpose; newest small models
Qwen 2.5 Coder / Math	Code generation; math problems
Phi 3.5 / 4 mini	Reasoning and coding tasks (3.5 vision also accepts images)
Gemma 2 / 3 / 4	Question answering, educational agents (Gemma 2 JPN: Japanese)
Mistral / Ministral 3	Translation, general NLP (Ministral: instruct and reasoning)
SmolLM2 / TinyLlama	Real-time performance, low-resource environments
OLMo 2	Fully open (weights + training data)
DeepSeek-R1 Distill	Step-by-step reasoning (slow: long “thinking” output)
Hermes 2 / 3	Function calling, structured (JSON) output

WebLLM Models Variants

Model	Parameters	Quantization	Precision	GPU Memory
Llama 3.2 / 3.1	1B, 3B, 8B	q0/4	f16/32	0.9 - 5.0 GB
Qwen 3 / 3.5	0.6B, 0.8B, 1.7B, 2B, 4B, 8B, 9B	q0/4	f16/32	1.4 - 6.4 GB
Qwen 2.5 (Coder, Math)	0.5B, 1.5B, 3B, 7B	q0/4	f16/32	0.9 - 5.1 GB
Phi 3.5 / 4 mini	3.8B, 4.2B (vision)	q4	f16/32	3.4 - 4.5 GB
Gemma 2 / 3 / 4	1B, 2B, E2B, 9B	q4	f16/32	0.7 - 6.4 GB
Mistral / Ministral 3	3B, 7B	q4	f16/32	2.9 - 4.6 GB
SmolLM2	135M, 360M, 1.7B	q0/4	f16/32	0.4 - 1.8 GB
TinyLlama	1.1B	q4	f16/32	0.7 GB
OLMo 2	1B, 7B	q4	f16/32	1.8 - 6.5 GB
DeepSeek-R1 Distill	7B, 8B	q4	f16/32	5.0 - 5.1 GB
Hermes 2 / 3	3B, 7B, 8B	q4	f16/32	2.3 - 5.0 GB

Gotcha: Tiny Models Don't "Know" All That Much

The **maximum** size model you can reasonably run in the browser is ~3-4B parameters (~2-4 GB of GPU memory).

If you need the model to **specialize** in a particular domain:

- Provide **context** to the model about your domain.
- **Fine-tune** the model on your specific data.
- Use a **specialized model** (e.g., Qwen 2.5 Coder, Qwen 2.5 Math, etc.)

LLM Context (For Apps)

Context Window

The **context window** is the maximum number of tokens a model can “see” at once.

Everything must fit inside the window:

- System prompt
- Chat history (**every** previous message)
- Context you provide (data, documents, etc.)
- User’s prompt
- **And** the model’s response

WebLLM: 4,096 tokens for all models (~3,000 English words)

Activity: Game Night

For a tiny LLM, extra context can significantly improve the quality of the model's responses. Clever app design can help select the context to provide to the model.

Game Night is an app that recommends board games for your group.

Its catalog (`games.json`) has 1,000 games (~30,000 tokens).

With a peer, design an app that collects the information needed to filter the catalog (to provide context to the model).

Providing Context

Filter your app's data:

```
const games = [
  { name: "Catan", kind: "strategy", minPlayers: 3, maxPlayers: 4, minutes: 90 },
  { name: "Ticket to Ride", kind: "family", minPlayers: 2, maxPlayers: 5, minutes: 60 },
  { name: "Codenames", kind: "party", minPlayers: 2, maxPlayers: 8, minutes: 15 },
  { name: "Pandemic", kind: "cooperative", minPlayers: 2, maxPlayers: 4, minutes: 45 },
];
const matches = games.filter(
  (game) => game.minPlayers <= 4 && game.maxPlayers >= 4 && game.minutes <= 30
);
```

Providing Context

Then, include only the matches in the prompt:

```
const messages = [
  {
    role: "system",
    content: "You are a board game expert. Only recommend games from this list:\n" +
      JSON.stringify(matches),
  },
  { role: "user", content: userPrompt },
];
```

Retrieval Augmented Generation (RAG)

You just did (simple) **RAG**:

1. **Retrieve** the data relevant to the user's task. (filter)
2. **Augment** the prompt with that data. (context)
3. **Generate** a response with the LLM.

Thursday: Retrieval when you **can't** simply filter your data.

Agenda

Last Half of the Semester

Tentative Schedule

- **Week 9:** Retrieval Augmented Generation (Homework 5)
- Week 10: Homework 5 + Project Preparation
- Week 11: Oral Exam II + Project, Sprint 1
- Week 12: Project, Sprint 2
- Week 13: Project, Sprint 3
- Week 14: Project, Sprint 4 (Thanksgiving Break)
- Week 15: Project, Sprint 4 + Presentation
- Week 16 + Finals: Final Oral Exam

Oral Exam II

Topic: Prompt Engineering

Format: Nearly the same as Exam I

Details during class 10/29.

Project

Working in groups of 3-4, prototype an app that uses a local LLM to perform a task.

INFO 5440: Provide server-side support for your app via a REST API.

Details next week.

Activity: Project Teams

Gather and find your team members for the project. **Exchange contact information** and **create a group in Canvas**.

INFO 4340: Groups of **3** people. (One group of **4** people)
(8 groups total: A-H)

INFO 5440: Groups of **3** people. (One group of **4** people)
(6 groups total: M-R)

Studio: Homework 4

The remainder of class is studio for Homework 4.

What's Next

Homework 4 due **today**

Thursday: Retrieval Augmented Generation (RAG), Start Homework 5