Open Discussion #1

NICE
MAY 2025

Theme

From LLMs to LCMs: Managing Concepts Assets

NexComm 2025 & DigitalWorld 2025

Open Discussion #1

NICE
MAY 2025

Ignitor

Prof. Dr. Petre Dini, IARIA, USA/EU

Driver

Prof. Dr. Nitin Agarwal, University of Arkansas - Little Rock & University of California, Berkeley, USA

Themes

Analogy: Syslog messages, Eye diopter, Nature Shapes Brain Captures,… (*context-dependent*)

(running for 100% is context-dependent, resource optimization, including water & energy)

LLMs: Stable/adopted vocabulary

(synonyms, idioms, …)

(intensive process, repetitive, hardly reproducible, trillions of parameters, overfitting, underfitting, input balancing, hallucinations, …)

DeepSeek: clustering (optimization), rounding meaning (similarity, sufficient approximation)

LCMs: ad hoc concepts, open range, uncontrolled definitions, concept management, definition conflicts, evolution control

LLMs over LCMs

Here's the big picture:

sql Copy Edit

```
User ↔ LLM ↔ LCM ↔ External Knowledge / Tools
```

- **LLM** handles: natural language understanding, interaction, generalization
- **LCM** handles: structured concept grounding, validation, abstraction, explanation
- **Goal**: use LCM to "filter", validate, or enrich the LLM's decisions/actions

LLMs vs LCMs

NICE
MAY 2025

Feature / Aspect	LLM (Large Language Model)	LCM (Large Concept Model)
🔍 Primary Objective	Model linguistic patterns and generate coherent text	Model, abstract, and reason with concepts and relationships
📦 Core Representation	Tokens, word embeddings, attention distributions	Concepts, relations, hierarchies, logical rules
🗄️ Training Data	Text corpora (web, books, conversations, code)	Ontologies, knowledge graphs, structured data, task schemas
🧠 Learning Style	Pattern learning via self-supervised next-token predictions	Symbolic abstraction, graph expansion, property induction
🧩 Compositionality	Implicit, via transformer layers and positional embeddings	Explicit, via concept composition (e.g., part-of, is-a, causes)
🔄 Reasoning	Mostly statistical (e.g., analogies, surface inferences)	Symbolic or hybrid reasoning (deductive, abductive, causal)
🧠 Contextuality	Strong in recent context windows (e.g., 128k tokens)	Strong in structured semantic context and background knowledge
🔬 Interpretability	Often opaque (“black-box”)	More explainable (conceptual graphs, rule sets, ontologies)
🤖 Use Cases	Text generation, dialogue, summarization, translation	Planning, abstraction, categorization, analogical problem solving
🔧 Tool Integration	Prompt-based, external tools via plugins	Programmatic interfaces, rule engines, structured APIs
🧠 Generalization	Strong zero-shot/few-shot generalization in language tasks	Strong generalization in conceptual domains, especially with schema
🛠️ Limitations	Hallucinations, shallow semantics, weak long-term abstraction	Brittle logic, data sparsity, poor natural language fluency

Inquiries

Experience reports

Concept management (validation, repositories, etc.)

Concept mismatching (conflict resolution, mediation, etc.)

Funneling LLMs into LCMs, (… then: to coding, testing, validating)

How to adopt such models for real systems

Themes

Analogy: Syslog messages, Eye diopter, Nature Shapes Brain Surgery,… (*context-dependent*)

(running for 100% is context-dependent, resource optimization, including water & energy)

LLMs: Stable/adopted vocabulary

(synonyms, idioms, …)

(intensive process, repetitive, hardly reproducible, trillions of parameters, overfitting, underfitting, input balancing, hallucinations, …)

DeepSeek: clustering (optimization), rounding meaning (similarity, sufficient approximation)

LCMs: ad hoc concepts, open range, uncontrolled definitions, concept management, definition conflicts, evolution control

THE STAGE IS YOURS

Back-up slides

Concept Mapping Layer (Bridge)

Converts LLM-parsed input into conceptual tokens using:

Schema mapping (e.g., concepts from ConceptNet, DBpedia)

Ontology alignment (e.g., OWL-based matching)

Embedding clustering or symbolic tagging

```
text                                                    Copy    Edit

"Explain gravity on the Moon."
↓
["Gravity", "Moon", "Mass", "Acceleration"]
```

LCM Core Layer

This is the reasoning brain, composed of:

Knowledge Graph: concepts + relations

Reasoning Engine: for inference (e.g., description logic, rule engines)

Conceptual Simulation: mechanisms for analogical or causal reasoning

Could be built with:

Component Technologies	
> Ontologies	OWL, Protégé
> Graph DBs	Neo4j, RDF/SPARQL
> Reasoning	Pellet, HermiT, Rule Engines
> Learning	Meta-Concept Embeddings (Vec2Graph)
> Inference	CLIPS, MiniKanren, ASP

Concept Validation & Augmentation Layer

Verifies LLM claims vs. LCM facts

```
text                                                    Copy    Edit

LLM: "Gravity on the Moon is due to the same mass as Earth."
LCM: ❌ Flag mismatch → Earth mass ≠ Moon mass → suggest correction
```

Identifies contradictions, gaps, or vagueness

Augments missing relationships or re-contextualizes sub-concepts

LLM Feedback Loop

Receives corrected, enriched, or validated representations
Reformulates responses in natural language
Can cite sources or provide structured explanations

Step-by-Step

- ☑ Identify the domain of your assistant (e.g., legal, medical, science)
- ☑ Build/obtain a concept graph or ontology (e.g., SNOMED CT for medical)
- ☑ Enable LLM to detect or tag concepts in user input via tools (e.g., spaCy, custom prompt chains)
- ☑ Use a rule engine or graph inference library to simulate reasoning
- ☑ Construct a loop where LLM can call, query, or validate through LCM and refine the output
- ☑ Add conflict resolution / trust scoring from LCM to gate hallucinations

END