RSM 313: **Large Language Agents** *for* Business **Problem Solving**

Instructor	Mihnea Moldoveanu Desautels Professor of Integrative Thinking Professor of Economic Analysis Founder and Director, RotmanDigital, Rotman Agentic AI Hub mihnea.moldoveanu@rotman.utoronto.ca
Office	Rotman School, South Building, 4020
Course Assistant	Leticia Souto Yakabi Leticia.yakabi@rotman.utoronto.ca
Office	Rotman School, South Building, 4020

***Course Meets*:** Wednesday 3:00-5:00 p.m. in WO

Course Prerequisites: CSC108H1 and 9.0+ Credits

Background. Generative AI (GenAI) – primarily comprising Large Language Models and Large Language Agents - has come through 3 (three) generations of technological progress during the course of 24 months. Some hype surrounding GenAI has been hyperbolic and unrealistic. But the technology - accessed and deployed via platforms such as OpenAI's ChatGPT, Google's Gemini and Meta's Llama3 – has already had significant impact in both business and academic environments (Mollick, 2024; Moldoveanu, 2024; Kosslyn, 2024). By now, there is a well-defined 'GenAI skill set' employers are looking for, and forward-looking academic institutions are seeking to build into their curricula [Zamfirescu Perreira et al, 2023; Tankelevich et al, 2024; Moldoveanu, 2025 a,b,c]. The right stance is not 'what AI can do', but rather 'what I can do with AI'. *This course is meant to help you develop that skill set and stance.*

Learning Objectives: We will consider and analyze existing and emerging business applications of Large Language Agents, ranging from client narrative, interaction, comment or reaction analysis and summarization (Amazon) to cross channel optimization of sales processes (Shopify) to personal(ized) banking assistants (JP Morgan Chase, RBC) to virtual executive trainers, coaches and interview assistants (IU University). Students will learn how to program/pattern/design GenAI agents via natural language scripts to solve useful problems across business functions and roles. They will learn to analyze human tasks in terms of how augmentable and automatable they are by agents that make use of LLM's. We will together design, test and refine agents that perform computational and inferential tasks (extractive/abstractive summarization, data pre-processing, hypothesis generation and testing, predictive and causal model building, calibration and testing, etc) as well as interactive tasks of communication, dialogue and co-reasoning tasks (virtual coaches, mentors, tutors, evaluators, feedback providers, meeting summarizers/annotators, etc). They will also learn to evaluate and reason about the organizational, economic and interpersonal consequences of introducing such agents or agent architectures in an organization. By the end of the course, participants should be able to:

- *model* human tasks and skills in both interactive ('dealing with other reasoning agents') and non-interactive ('dealing with knowledge and perceptual structures that do not model you back') settings;
- *build* LLM-based agents and agent teams that together can replicate, augment or modify the tasks and workflow of an individual, team or organization;
- *evaluate* the performance of such agents in a multi-dimensional setting and –
- *refine agent architectures* in ways that improve them relative to specific performance metrics.

Participants will be able to understand the implications of applying deep tech innovation in the LLM space (new agents, new models, new protocols, new architectures), so that they (1) are not dependent on 'middleman' translators and synthesizers, and so that (2) that can themselves be reliable agents of change and transformation in their organizations. Custom LLA's (TA's, coaches, explainers) will help learners understand difficult concepts, models, methods, theories and applications.

Course Architecture and Philosophy: Learning to use GenAI by building and studying with GenAI. The course makes use of GenAI agents to help students learn how to build GenAI agents for and by themselves. It uses GenAI agents to give students feedback on quintessentially human skills – like, making credible commitments to validity, responsiveness, coherence and completeness via speech acts. So:

> *No 'what' without a 'how to' and no 'how to' without a 'why'*. We will not teach any 'how to' without 'why this works', nor will we teach any 'what is this?' without having learners understand 'why is this important?' Every concept will be taught via a use case, and every use case will rehearse a set of skills required to 'make it work' and to explain to others why it works. Which is why: *assessing know-how and know-what go hand in hand*. We evaluate learners on the basis of *what they can do with what they know*, not just on what they know. Oral Q&A sessions will be common.

Course Evaluation.

This is a course that encourages and enables *continuous learning.* This is the only way to understand what LLM's are, what they do, and how to use them to design useful agents. You will get lots of along the way feedback and will actively lead discussions. Evaluation is meant to reflect this continuous interactive feedback approach, and is structured as follows:

Class participation: being in class, on time, every time; tuning in, asking relevant questions, giving informative, valid, responsive answers, helping everyone else learn the discipline:	*20%*
In-class presentations (2 for each participant; typically you will lead off discussion of a topic or question, sign up sheet will be distributed in Week 1)	15%
Weekly write-ups (500 words max, addressing either a question to be considered or a question that you find engaging about the readings)	10%
3 in class quizzes (pen and paper or using a video (asynchronous) platform	15%
Final Project (individual, starts on week 8)	40%

Rotman Commerce Attendance Policy

Rotman Commerce students are expected to make every effort to attend each class. Infrequently, students may miss term work, e.g., quizzes, assignments, etc., due to unplanned and extenuating circumstances and must follow the Request for Special Consideration process as outlined in their course outlines. However, Rotman Commerce will not approve any Request for Special Consideration for participation marks for missed classes. Any such request will be denied.

Academic Integrity and Statement on the Use of AI Tools. You are encouraged and, in many cases, required to use GenAI tools. You are strongly encouraged to get a paid subscription to OpenAI's ChatGPT ($26.99/mo) for the duration of the course (January 14-April 1). You are expected to do your own work but are not discouraged from cooperating and collaborating with other students on your project. That would run counter to the ways in which work, today, is done. What *is important* is that if you use AI or you collaborate with someone else, you reference and cite this in a way that credits the 'other agent' specifically for the work they have done.

References to Introduction

Kosslyn, S. M. (2024). *Learning to flourish in the age of AI*. Routledge.

Moldoveanu, M. C. (2024). *Soft skills: How to see, measure and build the skills that make us uniquely human*. De Gruyter. doi:10.1515/9783111055527

Moldoveanu, MC. 2025a. *An Agentic Re-Design for Higher Education*. Forthcoming in *The Agentic Re-design of Higher Education*, Seoul: Taejae University Press.

Moldoveanu, MC. 2025b*. GPT's Can Help Humans Become Better at Being Human*. Forthcoming in *The Agentic Re-design of Higher Education*, Seoul: Taejae University Press.

Moldoveanu, MC. 2025c. *Meta-Human:Working with GenAI Agents Requires a New Skill Set and New Approach to learning Design.* Forthcoming in *The Agentic Re-design of Higher Education*, Seoul: Taejae University Press.

Mollick, E. (2024). *Co-intelligence: Living and working with AI*. New York: Portfolio.

Tankelevitch, L., Kewenig, V., Simkute, A., Scott, A. E., Sarkar, A., Sellen, A., & Rintel, S. (2024). The metacognitive demands and opportunities of generative AI. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24)* (24 pp.). Association for Computing Machinery. doi:10.1145/3613904.3642902

Zamfirescu-Pereira, J. D., Wong, R. Y., Hartmann, B., & Yang, Q. (2023). Why Johnny can't prompt: How non-AI experts try (and fail) to design LLM prompts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)* (pp. 1–21). Association for Computing Machinery. doi:10.1145/3544548.3581388

Session 1. What is an agent? What is a Large Language Agent? What can LLMs do—and what can we do with LLM's do via agents?

An *agent* adds goal-directed behavior and iterative, composable interactions: planning, research, patterned reasoning and calculation, tool invocation (search, code, databases), environment memory, and policies for when and how to act. Practical agent frameworks (e.g., LangGraph) provide stateful, multi-actor workflows and control flow for complex business processes, such as document diligence, market scanning, and client-ready drafting. In enterprise settings, agents orchestrate LLM calls with retrieval (RAG), verifiers, and guardrails to reduce error and cost, while surfacing where human oversight adds value.

We will see agentic AI as a new *operational layer* for cognitive work: LLMs provide the substrate; humans impose objectives, contexts, tools, and evaluation loops. This separation lets us deploy workflows that are auditable, improvable (through logs, tests, rubrics), and safer than ad-hoc prompting. LLMs give you capabilities; the agents make them useful.

Readings

Amodei, D. 2024. *Machines of Loving Grace: Work and Meaning*. www.darioamodei.com. *Read carefully, thoughtfully. Engage. Be prepared to ask and answer questions.*

Moldoveanu, M.C. 2024. *Time to Act: Preparing Youth for Work in an AI Powered World*. My report to the UN (2024). *Background on the scale and scope of the new economic story.*

McKinsey&Co.2025. *The State of AI in 2025.*

Questions to be answered:

1. What is the agentic design skill set?
   a. What is an agent?
   b. What is a Large Language Agent?
   c. What can LLM's do?
   d. What can humans do?
   e. What can LLM's do that humans cannot?
   f. What can humans do that LLM's cannot?
   g. What do humans need to learn to become designers of LLA's?

Session 1 Jobs to be done.

I. Get to know each other's goals and purposes.
II. Canvass the range of cognitive tasks and skills that LLM's can be used to carry out.
III. Discuss the concept of an agent and agency.
IV. Discuss the layout and philosophy of the course.
V. Agree, collectively, on how we will hold one another accountable for learning.

Session 2. The Machines: Understanding the Sub-symbolic Layer.

Large Language Models (LLMs) are neural networks trained to predict the next token (sub-word) in a sequence on the basis of latent features that encode the co-distributional patterns of words-in-context. Their architecture—the Transformer—replaced recurrent and convolutional networks for language by using self-attention to model long-range dependencies efficiently (Vaswani et al., 2017). The "sub-symbolic layer" refers to distributed representations—vector encodings of words, phrases, tasks—on which the model learns statistical regularities. This enables impressive emergent behaviors (few-shot learning, tool use, instruction following), but also limitations (hallucinations, brittle reasoning without structure). The original GPT-3 paper crystallized the modern view of LLMs as general-purpose learners through in-context learning (ICL), where the model conditions on a handful of examples to solve new tasks without updating weights (Brown et al., 2020).

A *Transformer* layer computes attention-weighted mixtures of token embeddings; stacking layers creates hierarchical abstractions, while residual connections and normalization stabilize training (Vaswani et al., 2017). This architecture underwrites the "foundation model" paradigm—train once at scale, adapt intensively.

Readings (to be posted)

Brown, T. B., Mann, B., Ryder, N., et al. (2020). *Language models are few-shot learners*. An important paper that shows how LLM's 'learn from examples'.

Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). *Attention is all you need*. The architecture design paper. Read by yourself first. Then, read the short summary of LLM mechanics below (high level). Then read again, by yourself or with the use of the *Explainer* agent I will share with you. Be prepared to answer questions.

Wolfram, S. 2023. What is ChatGPT Doing? And Why Does it Work? From stephenwolfram.com. This is a very useful tutorial, which Stephen has also turned into a book.

Questions to be answered.

1. What is the self-attention mechanism?
2. What is positional encoding?
3. What are the functions of the various components of a Transformer architecture?
4. What is the basis for 'transfer learning'?
5. What do token probabilities come from?
6. What is a 'model'?
7. What is a 'language model'?
8. What is a 'foundational model'?

Jobs to be done.

I. Understand ‘the machine’ at several different levels.
II. Understand the different phases of training (pre-training, training, agentic design, evaluation)
III. Relate the mechanics to observed behaviors of LLM’s.

Brief, High level Explanation of How LLM’s ‘Work’ (Based on Brandenburger and Provost, 2023)

What an AI system like ChatGPT includes

A system such as ChatGPT is not just a single model. It typically includes:

- one or more large language models (LLMs),
- a dialog-management mechanism for interacting with users,
- and potentially additional components (e.g., web search or other features beyond “classic” ChatGPT).

The core predictive mechanism: “giant” nonlinear multinomial logistic regression

At heart, an LLM can be viewed as having estimated the coefficients of an extremely large nonlinear multinomial logistic regression from its training corpus.

- The model begins as an equation with many free parameters (the coefficients).
- A machine-learning procedure sets/fits these coefficients to useful values.
- “Giant” means **billions** of coefficients, even **over a trillion**.
- Once trained, the equation is used for prediction:
  given an input sequence of words, it outputs a **probability distribution over the next word**.

Tokens (not words) and what “features” and “targets” mean here

In practice, LLMs operate on **tokens**, not words.

- The token set is **smaller than the set of all words** (e.g., in English).
- In principle tokens could be as small as **26 letters plus additional symbols**, but that is too small to work well.
- Real token sets are much larger and include:
  - many whole words,
  - pieces of words (e.g., common syllables),
  - letters and punctuation,
  - and components useful for other text applications, such as **computer programming**.

- These tokens are the values the regression "features" (input variables, the *x*-variables) can take.
- The target (the *y*-variable) is the **probability of each possible output token**.
- The training corpus need not be only text: it could include things like **GitHub code** or a **database of images**. For simplicity the explanation speaks in "words," but it intends tokens representing text chunks or other objects.

How text is generated: the continuation protocol

When queried, the system calls the regression equation to produce a probability distribution over all possible continuation words/tokens given the input, then chooses a continuation according to a protocol.

- Example protocol:
    1. select the **n** continuation words with highest estimated probability (for some integer **n**),
    2. then **sample** one word from that subset, e.g., using those probabilities.
- To generate the second word, feed the original input plus the first generated word back into the equation; generate the next word the same way.
- Iterate until a stopping point (e.g., selecting a "finish" token).
- The resulting output text (a sequence of words/tokens) is returned to the user; this whole sequence is what we see as ChatGPT's output.

Where training data come from

Training data (used to determine the regression coefficients) can be collected differently across LLMs. As the authors understand it for OpenAI's GPT models:

- data collection essentially starts from the broad corpus of what has been written and digitized, obtained largely by **crawling the web**,
- with some subcorpora more heavily weighted; for example, more authoritative sources such as **Wikipedia** are significantly **overweighted**.

Supervised learning via next-token "labels," plus post-training

LLMs are trained using **supervised machine learning**, meaning the training data are labeled so a "correct" response is encoded.

- Classic analogy: images labeled "cat" vs "non-cat."
- For LLMs:
    - an input instance is a **contiguous sequence of text** drawn from the training corpus,
    - the label is the **actual next word** following that sequence.

    - This is why “correct” is in quotes: it is correct *relative to what was written in the source data*—the next word that appears there is de facto correct.

- This backward-looking correctness differs from the real target use: producing a new chat response to user input. The next word in the training corpus is not necessarily the next word in the *best* target response.
  Nevertheless, massive pretraining on prior text provides very effective **proxy labeling**.

- In “ChatGPT”:
    - **P = pretrained** (referring to this pretraining),
    - **G = generative** (it generates text),
    - **T = transformer** (the neural network architecture used; more on this below).

- LLMs can also be **post-trained** to be better at chat than a purely pre-trained LLM; ChatGPT uses such a post-trained LLM.
    - One post-training example: a human tester generates multiple outputs to an input and selects the best-sounding output.

- Key recap: LLM training works well because massive models need massive labeled data, and for next-word prediction the labels are naturally available—the next word in the text.

Why neural networks can do this, and the overfitting/complexity issue

Why can the chosen computer architecture—a **neural network**—estimate the logistic-regression mapping?

- The core mathematical rationale is the **Universal Approximation Theorem** (in various versions): functions in a very large family can be approximated arbitrarily well by the input–output mapping of a suitably chosen neural network.

- But these theorems also warn about **overfitting**, so **complexity control** is crucial in large neural network design:
    - too much complexity fits training data too well and harms performance on new inputs (overfitting),
    - too little complexity is also suboptimal – large prediction errors, undifferentiated across input classes.

  A way to limit overfitting is the use of **massive amounts of labeled training data**, which LLMs have.

Session 3. From Query to Instruction Set.

A 'plain old' LLM query is what one unthinkingly types into a ChatGPT or Gemini chat window. 'GIGO' (garbage in garbage out) almost always applies.

A *well-structured query* is specific about task, audience, constraints, and output format.

Effective *instruction sets* (prompts) are meant to design and constrain the behavior of an LLM for a use case. They require precise specification of a **role** (persona), **objectives**, **workflow instructions**, **context**, **examples**, and evaluation **criteria**. Clarity, delimiters, and testable checks are practices that robustly improve outputs. Personas scaffolds nudge style and stance, but interact with model biases and can induce motivated reasoning if over-indexed on identity.

We will treat prompts as *programs*: version them, add tests, monitor failures. As an example, DSPy formalizes prompts as parameterized modules that can be compiled/optimized to maximize a metric—useful when you need reproducible improvements over guess-and-check engineering.

Readings

Moldoveanu, MC. 2025. *Rudiments for Agentic Instruction Set Design.* Excerpt from Meta-Human.

Anthropic. 2024. *Building Effective Agents*. Read closely, and be aware that the guidance provided is not easily implementable. There is much that lies between high level blueprints and fine grained agentic design.

Anthropic. 2025. *Effective context engineering for AI agents.*

Questions to be answered

1. What kind of semantics should we adopt when interacting with LLM's – and, why?
    a. What different kinds of semantics are there?
    b. What are they good and bad for?
    c. Why are distributional semantics not enough for the kind of work we need to trust AI to do for us?
2. What are the components of a useful i

Jobs to be done

I. Understand the function and uses of a *persona*.
II. Critique personas and improve them
III. Understand the function and uses of *objectives*.
IV. Critique objectives and improve them.
V. Critically parse sample agentic instruction set designs.

Session 4 — Instruction Set Engineering.

Reliable instructions are *unambiguous*, *testable*, and *bounded by checks*. Three approaches to patterning reasoning processes in LLM's have emerged as important. We will discuss and unpack:

> **Chain-of-Thought (CoT)** asks models to show intermediate reasoning; it can perform arithmetic and commonsense tasks (Wei et al., 2022). Self-Consistency further improves CoT by sampling multiple reasoning paths and majority-voting the answer (Wang et al., 2022). Note some providers advise *not* to request verbatim CoT in production to avoid leaking sensitive reasoning; instead, instruct "think step by step, then provide final answer only" or verify with hidden chains.
>
> **Tree-of-Thoughts (ToT)** explores multiple branches with look-ahead, scoring partial solutions before expansion—useful for planning and optimization (Yao et al., 2023).
>
> **Refiners/Self-improvement** (Self-Refine; Reflexion) iterate: draft → critique → revise, optionally using external feedback or dynamic memory (Madaan et al., 2023; Shinn et al., 2023).

Instruction sets can also be embedded in *actor–critic* training (e.g., PPO) or adversarial red-teaming. PPO stabilizes policy updates via a clipped objective and underlies many alignment pipelines (Schulman et al., 2017). In practice, you can *simulate* actor–critic in prompts: actor proposes, critic scores against a rubric, actor revises.

Questions to Ask

What is an argument?

What is a chain of thoughts?

What is a tree-structured chain of thoughts?

What is an audit trail for an inference?

Which inferences 'count' and which do not?

Jobs to be Done

Hand-code chain of thought prompting and tree of thought prompting and simulate response.

Understand trees, chains and graphs as topologies of inference and reasoning.

Examine how reasoning models (GPT 5.2) and non-reasoning models (GPT 4o) differ in performance across tasks.

Readings

Wang, X., Wei, J., Schuurmans, D., et al. (2022). *Self-consistency improves chain-of-thought reasoning in language models*. **arXiv:2203.11171**. arXiv

Wei, J., Wang, X., Schuurmans, D., et al. (2022). *Chain-of-thought prompting elicits reasoning in large language models*. **arXiv:2201.11903**. arXiv

Yao, S., Bosma, M., Zhao, S., et al. (2023). *Tree of thoughts: Deliberate problem solving with LLMs*. **arXiv:2305.10601**. arXiv

Optional

Madaan, A., Tandon, N., Yazdanbakhsh, A., et al. (2023). *Self-Refine: Iterative refinement with self-feedback for LLMs*. **arXiv:2303.17651**.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). *Proximal policy optimization algorithms*. **arXiv:1707.06347**.

Session 5 — Teaching Your Agents (and, Models)

Few-shot learning uses demonstrations to convey task format and label space; surprisingly, ground-truth labels often matter less than structure and coverage (Min et al., 2022). From business data (analyst notes, style guides, SOPs), you can *extract procedures*—decision trees, checklists, templates—by asking the model to induce rules, then test them against counterexamples. Pair few-shot with *instruction-following* alignment to make outputs consistent and safe (Ouyang et al., 2022).

System-level instructions (developer/system prompts) specify rules, heuristics and guidelines for the agent's behavior: tone, scope, safety, tool use, and verification. Providers offer best-practice guidance for writing developer messages, delimiting context, and modularizing tasks across steps/windows (OpenAI; Anthropic). The system prompt can be thought of as a as *policy*: version it, tie it to tests, and log failures for revision.

For robust procedures, add retrieval (docs, policies), enforce schemas, and inject *verification* calls (fact-checking queries, calculations). This hybridization—ICL + RAG + tests—turns tacit team know-how into an auditable, improvable agent.

Readings

Min, S., Lyu, X., Holtzman, A., et al. (2022). *Rethinking the role of demonstrations*: What makes In-Context learning Work? … **arXiv:2202.12837**.

Ouyang, L., Wu, J., Jiang, X., et al. (2022). *Training language models to follow instructions with human feedback*.

Please review the Brown et al (2020) paper, as well.

Questions to consider

What is 'transfer learning'? What is cognitive skill transfer?

How do they work in humans?

How do they work in LLM's?

What is Reinforcement Learning?

What is Reinforcement Learning with Human Feedback?

Jobs to be Done

I. Understand transfer learning: what is transferred, what is learned when transfer happens, where is the learning transferred to?
II. Understand the role and structure of few-shot examples for LLA design.

Session 6 — Failure Modes: Hallucinations, Complexity, and Other Infelicities

Agent designers need to track *agent quality* like developers track software quality: test suites, failure taxonomies, causal paths analysis for failure modes and likelihoods and dashboards. But, the quality of the outputs of an agent is strongly dependent on the validity and reliability of the baseline LLM the agent uses. In order to do so, they need to understand the ways in which LLM's and LLA's fail. These failure modes have something in common with those of human agents, but also have distinctive and peculiar components that are less intuitive. In this session, we will consider in some detail the nature and role of hallucinations and other forms of failure (like 'complexity failure') in LLM's, and examine the mechanisms by which they come about in different phases of development (pre-training, training, post-training). We will also look at different ways in which we can

Questions to Consider

1. What is a hallucination?
    a. How does it differ from an error?
    b. How does it differ from an omission?
2. What other kinds of failures of registration, testimony and inference do you expect from an LLM given the design, training and testing regimens of Kalai?
3. What is 'complexity failure' in an LLM?
4. How does it relate to claims about 'reasoning' in LLM's?
5. What are three tests you would perform on an LLM to figure out if it is prone to complexity-related failures of reasoning?

Jobs to be Done

i. Map and discuss different modes of epistemic failure in humans and machines;
ii. Understand the role, function and discipline of an adversarial approach to testing memory, skill and other capabilities in people and machines.
iii. Understand ways in which one can evaluate LLM's for the kind of capabilities we need in the agents we want, and possible remedies at the level of agent design (as opposed to LLM design).

Readings

Kalai, AT, O. Nachum, SS. Vempala, E. Zhang. 2025. *Why Language Models Hallucinate*. arXiv:2509.04664.

Shojaee, P. et al. 2025. *The Illusion of Thinking: Understanding the Strengths and Limitations of Reasoning Models via the Lens of Problem Complexity.* arXiv: 2506-06941.

Opus, C, and A. Lawson. 2025. *The Illusion of the Illusion of Thinking*, arxiv: 2506.09250v1.

Optional

Liang, P., Bommasani, R., Lee, T., et al. (2022). *HELM*. **arXiv:2211.09110**. arXiv

Lin, S., Hilton, J., & Evans, O. (2021). *TruthfulQA: Measuring how models mimic human falsehoods*. **arXiv:2109.07958**. arXiv

Session 7. Constraining Your Agent: Resource-Augmented Generation and Fine-Tuning.

Our agents often need to be constrained so as to process information that is constrained by all and only the information we want them to refer to. Examples are numerous, and range from answering service calls, negotiating rates for medical services, providing teaching and research assistance, writing market reports and board reports, and many others. In this case, we want to be able to design methods for constraining context, which is what RAG (Resource Augmented Generation) architectures do. In this session, we will delve into the mechanics and structure of RAG, starting from vectorized memory systems, interfaces to the Large Language Model which functions as an operating system that implements cognitive functions, ranking of possible answers or replies, selection of answers or replies, and the avoidance of leaks. In this session, we will devote ourselves to understanding LLM memory management systems, again, using original papers and contributions.

Questions to be considered

1. What is a vectorized memory?
    a. How does it work?
    b. What do vectors encode?
    c. How are they accessed?
2. What is a RAG engine?
    a. How does it work?
    b. How does it constrain the behavior of an agent?
    c. What are its drawbacks?

Jobs to be done

I. Understand the mechanics of memory management systems.
II. Understand how RAG systems work
III. Understand how RAG systems can fail, and, why.

Readings

Gao, Y., et al. (2023). *Retrieval-Augmented Generation for LLMs: A survey*. **arXiv:2312.10997**.

Sarthi, P. et al. 2024. *RAPTOR: Recursive Abstractive processing for Tree-Organized Retrieval*. arXiv: 2401.18059.

Session 8 — Cognitive Operators for Agentic Design I: Explain, Justify,

To design an agent, we need a language for describing what they do. This is not our ‘regular’ language, but rather a ‘meta-language’. Its terms do not refer to what we think, but to how we think. They refer, specifically to the words that refer to objects, events, modalities and relations, rather than to the objects, events and modalities themselves.

In this session, we will focus on descriptors, verifiers and falsifiers for a special set of cognitive operators: “Explain X in terms of Y,” “Justify X with Z constraints,” “Search S subject to ZC,” “Verify claim using sources A–C,” “Prove or disprove X from Y.” Encoding these operators improves reliability and *composability*: teams can snap together explain-justify-verify loops across tasks (e.g., investment memos, policy drafts). To structure multi-step reasoning, we can use Chain of Thought-structured instruction sets with explicit scoring functions and rejection sampling; to constrain outputs, we can edit, test and re-compile instruction sets.

Operators pair naturally with RAG and program-of-thoughts (delegating exact calculations to code), and with *verifiers* (unit tests for logic). In management work, we can define a *library of operators* (e.g., Diagnose–Debias–Decide for pricing; Frame–Forecast–Falsify for strategy), then instrument usage to learn what works. We will consider these and other examples.

Readings

Moldoveanu, MC. 2025. *Meta-Human (excerpt): Cognitive Operators for Large Language Agent Design*.

Summers, TR, et al. 2024. *Cognitive Architectures for Language Agents. arXiv: 2309.02427*.

Questions to be Considered

1. What is a cognitive operator?
    a. How does it work?
    b. What are its satisfiability conditions?
    c. What are its failure modes?
2. How does an explanation explain?
3. How does a justification justify?
4. How does a search operator ‘know’ when to stop searching?

Jobs to be Done

I. Understand the structure and function of cognitive operators.
II. Sketch and design a set of specific cognitive operators for explaining, justifying and searching in a constrained space.

Session 9 — Cognitive Operators II: Structural, Relational, Causal, Teleological, and Functional Modeling.

Many business questions are *causal* ("Did campaign A cause lift?"), *teleological* (goal-oriented, "What is our objective in implementing this process?"), or *functional* ("What is the role of each subpart?"). LLMs can *scaffold* various kinds of modeling, including causal modeling—naming assumptions, drawing DAGs, proposing identification strategies, which should always be subject to testing, verification and revision. Recent surveys emphasize integrating causal inference with LLM pipelines—explicit graphs, refutation tests, and do-calculus-aware tooling (DoWhy) (Sharma & Kiciman, 2020; Ma, 2024). Teleological modeling, relevant to agent design, keeps purpose and success criteria explicit—clarifying when the agent should *not* act (e.g., uncertainty too high) (Noller, 2024).

For an investment-banking analyst agent, encode operators: "Construct DAG for driver → outcome," "List identifying assumptions," "Select estimator," "Run refutation tests," "Summarize limits." This guards against spurious correlations and over-confident memos. Organizationally, pair these agents with human reviewers for model selection and sanity checks.

Readings

Ma, J. 2024. *Causal inference with large language model: A survey*. **arXiv:2409.09822**.

Moldoveanu, MC. 2025. *Cognitive Operators for Modeling, Inference, Prediction and Judgment,* excerpt from Meta-Human.

Optional

Noller, J. 2024. *Extended human agency: Towards a teleological account of AI*. **Humanities & Social Sciences Communications**.

Questions to be Considered

1. What is a model?
2. What is a dynamical model? How does it differ from other models?
3. What is a causal model? How is it different from a predictive model?
4. What is an intervention? What role does it play in corroborating a causal link?

Jobs to be Done

I. Understand Modeling as a set of cognitive operators.
II. Understand failure and success modes of such operators.
III. Understand compositional approaches to modeling and the discipline of designing modeling operators.

Session 10 — Dialogical & Interactive Agents I: Interaction architecture; know-your-user; theory of mind; interview agent

Some of the most useful agents you can build do not just automate or augment your own cognitive tasks: they also interact with others and help them accomplish tasks more quickly, or help them accomplish tasks they could not otherwise accomplish. In this and the next session, we will Interactive agents need a *dialogue architecture*: turn management, memory, user modeling, evaluation protocols, testbeds and scoring rubrics for interactions.

We will start in this session with three related topics: 'Theory of Mind' in LLM's – or, the ways in which they interpret the aims, goals, beliefs and interaction patterns of a user – and matters of tone (sycophancy) and dialogical 'blueprint' and the ways in which they can be brought together in the design of useful instruction sets.

Claims of "theory of mind" in LLMs are debated: some evidence suggests emergent capabilities on ToM-like tests, while others argue performance reflects dataset artifacts and test design or show that performance varies with prompt and context (Wilf et al., 2023).

We will consider how structured 'preplay' can assist in designing agents that get theory-of-mind-intensive tasks 'right', and hpow to effect 'personality changes' in LLA's using highly expressive instruction scripts (HEIS).

Questions to consider

1. What is a 'theory of mind' capability in a human?
    a. What about an LLM? Any differences in detection?
    b. Measurement?
    c. Interpretation?
2. What implications do TOM capabilities have for our agentic designs?
3. What are the failure modes of agents related to TOM mishaps?

Jobs to be done.

I. Understand TOM as a set of functions and skills that can be applied to tasks that fulfill objectives.
II. Understand the ways in TOM is important to agentic design.
III. Understand the ways in which TOM failures and mishaps can

Readings

Sharma, M. et al. 2023. *Towards Understanding Sycophancy in Language Models,* arXiv 2310:13548.

Street, W et al. 2024. *LLM's Achieve Adult Human Performance on Higher Order Theory of Mind Tasks.* arXiv: 2405:18870

Ullman, TD. 2023. *Large language models fail on theory of mind tasks*. **arXiv:2302.02083**.

Wagner, E. et al. 2025. *Mind Your Theory: Theory of Mind Goes Deeper Than Reasoning.* arXiv: 2412:13631.

Session 11 — Dialogical Agents II: Scorekeeping, Scoring Rubrics, Transcript Evaluation

"Pre-play", duly worked into instruction sets and few shot examples that structure an agent, can recursively refine the interaction by an agent with a user by eliciting goals, prior knowledge, constraints, and evaluation criteria—thus reducing misalignment. However, we still need a set of guidelines, heuristics and procedures for *designing* interactions that our agents will have with end users. We will consider the role of instruction set architecture, the design of scorekeeping rubrics for conversational interactions, the evaluation of interactions using such rubrics, the re-design of instructions and agents to heed the results of such testing. This is the at the core of successful agentic design, as we will see from a number of examples, ranging from debt collection and service calls, to negotiations, explanations, and coaching sessions.

Questions to Consider

1. What are the key norms of dialogical 'ethics' for humans?
2. What are they for LLA's?
3. Why is each important?
4. How do they figure into the design of a useful interactive agent?

Jobs to be done

I. Understand why dialogical ethics needs to be deployed to score and evaluate conversational interactions.
II. Relate dialogical ethics to inferential, distributive and representational semantics.
III. Understand how the bot-breaking protocols can be used to refine the interactive protocols agents use.

Readings

Moldoveanu, MC. 2026. *Dialogical Grammars for Interactive Agent Design*, excerpt from *Meta-Human.*

Bai, Y., Kadavath, S., Kundu, S., et al. *Constitutional AI: Harmlessness from AI feedback*. **arXiv:2212.08073**.

Session 12. Advances and Advanced Applications.

We will aim to put it all together here, both via live answers to your questions and concerns, and the discussion of a set of forward-looking applications that, variously –

- Go ‘multi-modal’ in the design of AI agents (video, voice, physical interactions) (Jeppa models);
- Create ‘hermetic’ AI only workflows that replicate entire human activity chains (FutureSecure AI);
- Are fully integrated into the application ecosystem currently available (Noira).

The selection of specific topics (and, readings) will be adaptive to your interests and concerns.

Missed Tests and Assignments
Students who miss a term test or assignment for reasons entirely beyond their control (e.g. illness) may request special consideration **within 2 business days** of the missed midterm/test/assignment due date.

In such cases, students must:

1. Complete the Request for Special Consideration form: https://uoft.me/RSMConsideration
2. Provide documentation to support the request, eg. Absence Declaration from ACORN, Verification of Illness Form etc.

Please note: Students may use the Absence Declaration on ACORN ***one time per term*** to report an absence and request consideration. **Any subsequent absence will require a Verification of Illness form or other similar relevant documentation.**

Students who do not submit their requests and documentation within 2 days may receive a grade of 0 (zero) on the missed course deliverable.

Statement on Equity, Diversity and Inclusion
The University of Toronto is committed to equity, human rights and respect for diversity. All members of the learning environment in this course should strive to create an atmosphere of mutual respect where all members of our community can express themselves, engage with each other, and respect one another's differences. U of T does not condone discrimination or harassment against any persons or communities.

Commitment to Accessibility
The University is committed to inclusivity and accessibility, and strives to provide support for, and facilitate the accommodation of, individuals with disabilities so that all may share the same level of access to opportunities and activities offered at the University.

If you require accommodations for a temporary or ongoing disability or health concern, or have any accessibility concerns about the course, the classroom or course materials, please email Accessibility Services or visit the Accessibility Services website for more information as soon as possible. Obtaining your accommodation letter may take up to several weeks, so get in touch with them as soon as possible. If you have general questions or concerns about the accessibility of this course, you are encouraged to reach out to your instructor, course coordinator, or Accessibility Services.

Academic Integrity
Academic Integrity is a fundamental value essential to the pursuit of learning and scholarship at the University of Toronto. Participating honestly, respectfully, responsibly, and fairly in this

academic community ensures that the U of T degree that you earn will continue to be valued and respected as a true signifier of a student's individual work and academic achievement. As a result, the University treats cases of academic misconduct very seriously.

The University of Toronto's Code of Behaviour on Academic Matters outlines the behaviours that constitute academic misconduct, the process for addressing academic offences and the penalties that may be imposed. You are expected to be familiar with the contents of this document. Potential offences include, but are not limited to:

In papers and assignments

- Using someone else's ideas or words without appropriate acknowledgement.
- Submitting your own work in more than one course without the permission of the instructor.
- Making up sources or facts.
- Obtaining or providing unauthorized assistance on any assignment (this includes collaborating with others on assignments that are supposed to be completed individually).

On test and exams

- Using or possessing any unauthorized aid, including a cell phone.
- Looking at someone else's answers.
- Misrepresenting your identity.
- Submitting an altered test for re-grading.

Misrepresentation

- Falsifying institutional documents or grades.
- Falsifying or altering any documentation required by the University, including (but not limited to) medical notes.

All suspected cases of academic dishonesty will be investigated by the procedures outlined in the *Code of Behaviour on Academic Matters*. If you have any questions about what is or is not permitted in the course, please do not hesitate to contact the course instructor. If you have any questions about appropriate research and citation methods, you are expected to seek out additional information from the instructor or other U of T or RC resources such as the RC Centre for Professional Skills, the College Writing Centres or the Academic Success Centre.

Email

At times, the course instructor may decide to communicate important course information by email. As such, all U of T students are required to have a valid UTmail+ email address. You are responsible for ensuring that your UTmail+ email address is set up and properly entered on ACORN. For more information visit the Information Commons Help Desk.

Forwarding your utoronto.ca email to a Gmail or other type of email account is not advisable. In some cases, messages from utoronto.ca addresses sent to Gmail accounts are filtered as junk mail, which means that important messages from your course instructor may end up in your spam or junk mail folder.

Recording Lectures

Lectures and course materials prepared by the instructor are considered by the University to be an instructor's intellectual property covered by the Canadian Copyright Act. Students wishing to record a lecture or other course material in any way are required to ask the instructor's explicit permission and may not do so unless permission is granted. Students who have been previously granted permission to record lectures as an accommodation for a disability are excepted. This includes tape recording, filming, photographing PowerPoint slides, Quercus materials, etc.

If permission for recording is granted by the instructor (or via Accessibility Services), it is intended for the individual student's own study purposes and does not include permission to "publish" them in any way. It is forbidden for a student to publish an instructor's notes to a website or sell them in any other form without formal permission.