Large Language Models and Recommender Systems: Opportunities and Challenges

ChengXiang ("Cheng") Zhai

Department of Computer Science

(Carl R. Woese Institute for Genomic Biology

School of Information Sciences

Department of Statistics)

University of Illinois at Urbana-Champaign

czhai@illinois.edu http://czhai.cs.illinois.edu/

Generative AI Revolution:
The Emergence of Large Language Models (LLMs)

- **Models**: massive transformer-based neural networks with billions/trillions of parameters
- **Data**: massive amounts of data, particularly text data, but also other kinds of data (e.g., software code, images etc.)
- **Learning techniques**: self-supervised (unsupervised) learning, supervised learning (fine-tuning), instruction tuning, reinforcement learning from human feedback (RLHF)
- **Applications**: a single model to perform many different (intelligent) tasks via natural language instructions
- **Potential:** moving closer to artificial general intelligence (AGI)

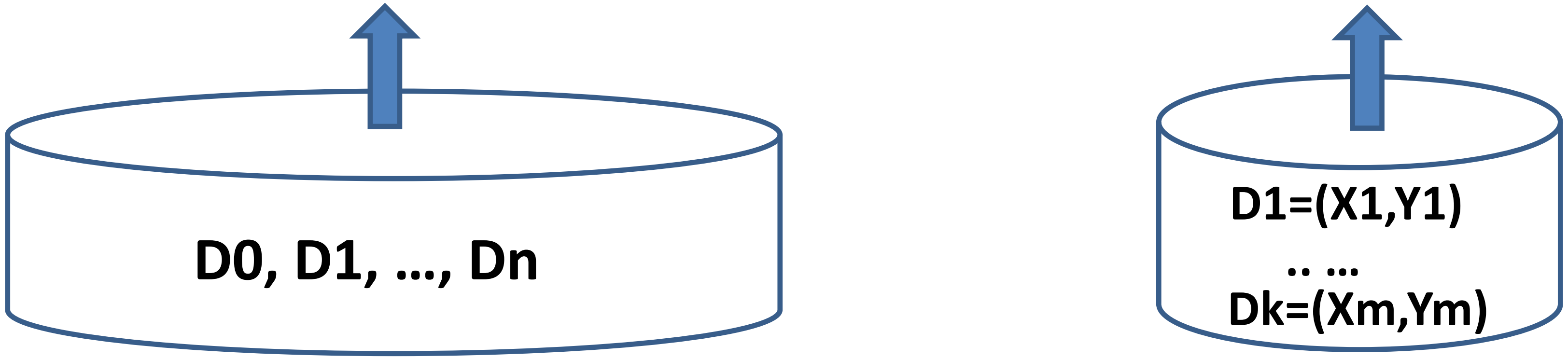
Broad Impacts of Large Language Models

- Broad impacts
  - Immediate opportunities for enhancing existing applications and creating new applications
    - Being able to understand natural language → enables conversational user-system interaction
    - Being able to generate fluent natural language → enables many text-generation tasks that would otherwise be infeasible
    - Fine-tuning (with instructions) further enlarges scope of applications
  - Disruption of research in many related fields, including particularly, NLP, IR, CV, and AI in general

Limitations of LLMs

- Trustworthiness **→ Explainable LLMs (Neuro-symbolic AI)**
  - Inability to explain why an output is produced (non-interpretability)
  - Hallucination (over-generalization)
  - Bias, fairness, ethics, …
- Objective functions are human-designed, not learned

$$\Phi^* = argmin_{\Phi}\ \lambda ReconstructError(D_0, D_1, \dots, D_n|\Phi) + (1-\lambda) TaskErrorInstruct((X_1, Y_1), \dots (X_m, Y_m)|\Phi)$$

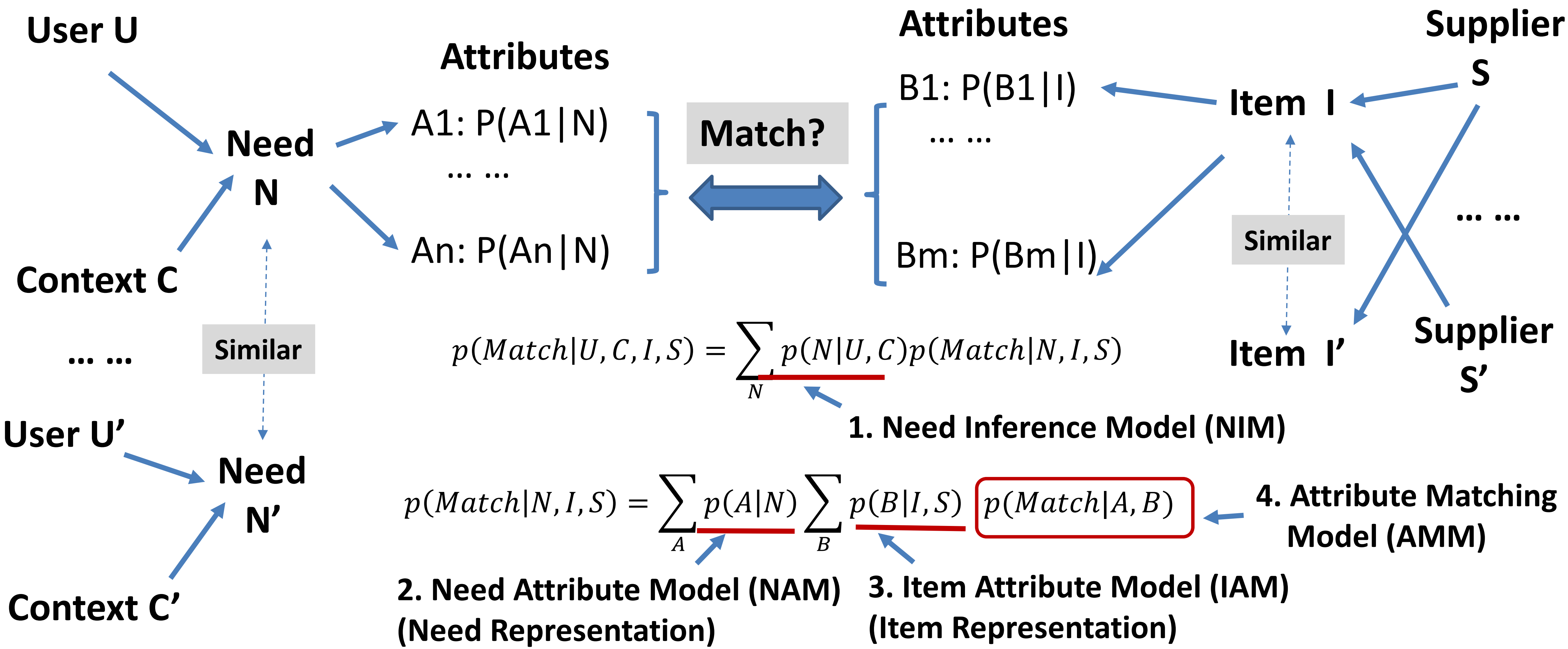


→ Motivation-Driven (Fully Self-Supervised) Learning

Recommender Systems

Problem: $p(Match|U,C,I) = ?$

Solution: End-to-End vs. Explainable Prob. Model



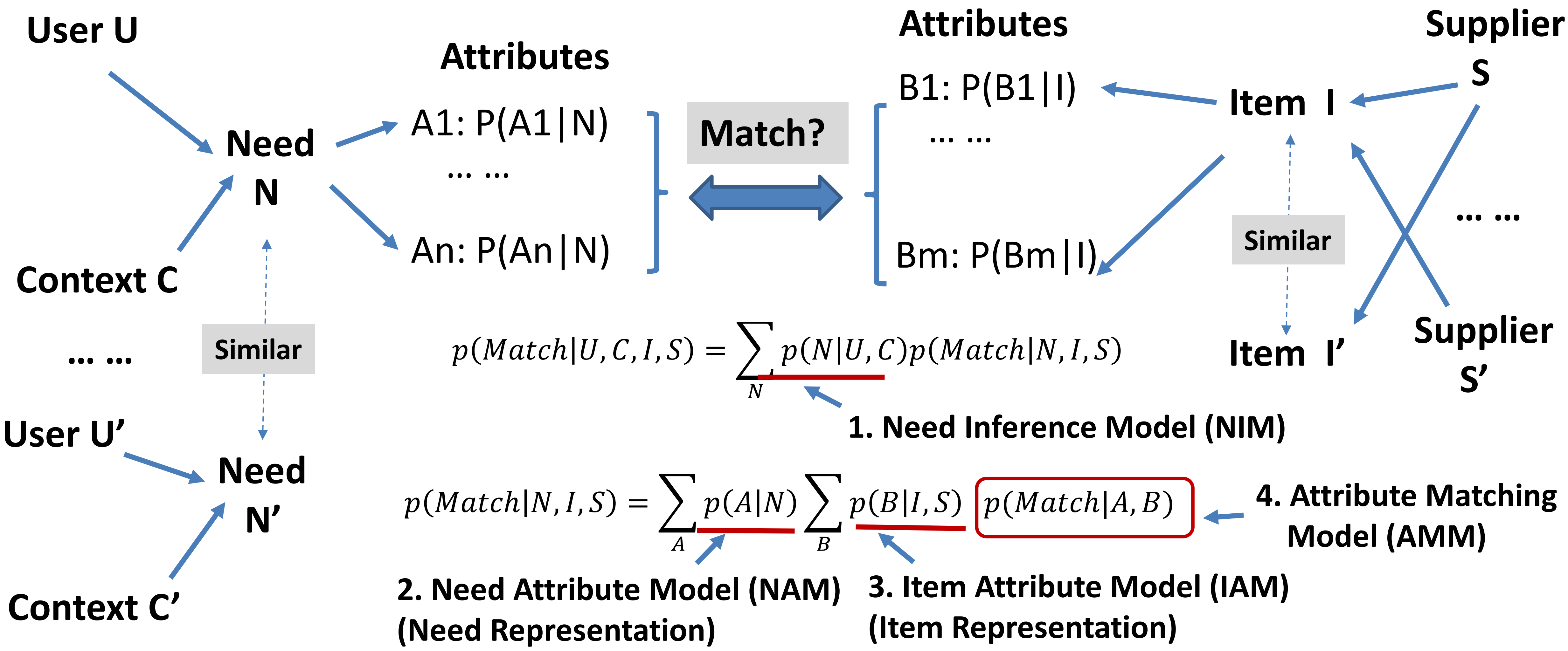
The Data and Information Systems Laboratories at The University of Illinois at Urbana-Champaign

LLMs meet Recommender Systems

- How can LLMs be leveraged to improve the current recommender systems?
- What is the potential for LLMs to transform the future recommender system applications?
- What are the major challenges in applying LLMs to recommender systems?
- Given the anticipated growth of LLMs, what would recommender systems look like in the future?

Improve Recommender Systems with LLMs

LLMs can be leveraged to build each of the four component probabilistic models and provide explainable recommendation



$$p(Match|U,C,I,S) = \sum_N p(N|U,C)p(Match|N,I,S)$$

1. Need Inference Model (NIM)

$$p(Match|N,I,S) = \sum_A p(A|N) \sum_B p(B|I,S) \boxed{p(Match|A,B)}$$

2. Need Attribute Model (NAM) (Need Representation)

3. Item Attribute Model (IAM) (Item Representation)

4. Attribute Matching Model (AMM)

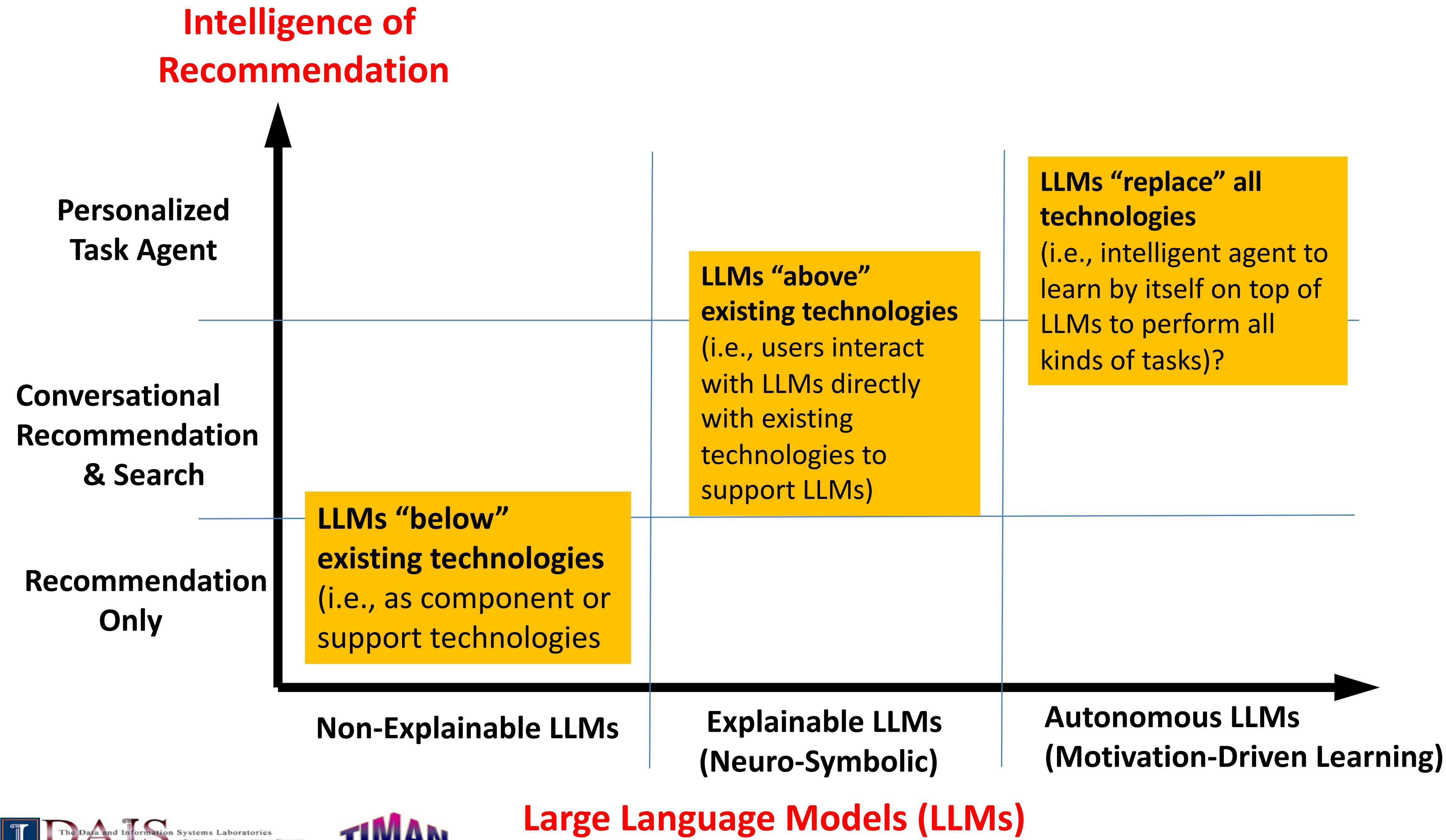
Potential of LLMs to transform recommendation

- Capabilities of LLMs
  - Being able to understand natural language ➔enables conversational user-system interaction
  - Being able to generate fluent natural language ➔ enables many text-generation tasks that would otherwise be infeasible
  - Fine-tuning (with instructions) further enlarges scope of applications
- LLMs directly trained/tuned to perform conversational content recommendation
  - It can already support question answering, summarization, and other text generation tasks
- LLMs will enable conversational recommender systems or a conversational information access agent to integrate support of search and recommendation

Challenges in Applying LLMs to Recommendation

- General Challenge: How can we leverage the imperfect LLMs to build a definitely useful application?
  - Strategy 1: Leveraging LLMs as component technologies in a system
  - Strategy 2: Having humans in loop (post-editing/validating results of LLMs)
  - Strategy 3: Give users direct control over LLMs (e.g., OpenAI's ChatGPT)
- Specific Challenges:
  - How can we leverage LLMs to estimate a probabilistic model? How to design prompts?
  - How can we integrate LLMs with an existing recommender system
    - How to fine-tune LLMs for recommendation? How to design the data sets, instructions, and the objective function?
    - How to optimize efficiency? How to transfer the knowledge from an LLM explicitly to a specific recommender systems (to avoid real-time execution of LLMs)?

Given the anticipated growth of LLMs, what would recommender systems look like in the future?


Intelligence of Recommendation
Personalized Task Agent
Conversational Recommendation & Search
Recommendation Only
LLMs “below” existing technologies
(i.e., as component or support technologies
LLMs “above” existing technologies
(i.e., users interact with LLMs directly with existing technologies to support LLMs)
LLMs “replace” all technologies
(i.e., intelligent agent to learn by itself on top of LLMs to perform all kinds of tasks)?
Non-Explainable LLMs
Explainable LLMs (Neuro-Symbolic)
Autonomous LLMs (Motivation-Driven Learning)
Large Language Models (LLMs)

New Direction: LLMs for Decision Support

- Decision Problem Specification
- Criteria/Preferences Refinement/Clarification
- Retrieval/Recommendation of Candidates
  - Multi-dimensional matching
  - Explanation of matching in each dimension
  - Aggregation of matching scores in multiple dimensions
- Analysis of Pros & Cons of a Candidate
- User-in-Loop for instant feedback and update
- Augmenting user intelligence
  - Scaffold learning
  - Question answering

Summary

- How can LLMs be leveraged to improve the current recommender systems?
  - **➔ Enable explainable probabilistic models for recommendation**
- What is the potential for LLMs to transform the future recommender system applications?
  - **➔ Conversational recommendation & search, particularly content recommendation**
- What are the major challenges in applying LLMs to recommender systems?
  - **➔ How can we tolerate imperfect LLMs? (Use LLMs as components, human-in-loop)**
- Given the anticipated growth of LLMs, what would recommender systems look like in the future?
  - **➔ Conversational recommendation & search**
  - **➔ Personalized intelligent task agent (LLMs for decision support)**

Thank You!

Questions/Comments?

czhai@illinois.edu

http://czhai.cs.illinois.edu/