Topic 9

Traditional collaborative filtering

VS

Xian Haotian Lan Tiancheng Jia Xiaoran Wang Liming Li Rongchao

LLM enhanced recommendation

PROBLEM FARMING

TASK DEFINITION

Learn a scoring function to rank items for a user under context:

$$r^{ui} = f(u, i, C)$$

OBJECTIVE FAMILIES

POINTWISE

$$\min_{\Theta} \sum_{(u,i)\in\Omega} \ell(r_{ui}, \hat{r}_{ui}) + \lambda |\Theta|_2^2$$

PAIRWISE

$$\max_{\Theta} \sum_{(u,i,j)} \log \sigma(\hat{y}_{ui} - \hat{y}_{uj}) - \lambda |\Theta|_2^2$$

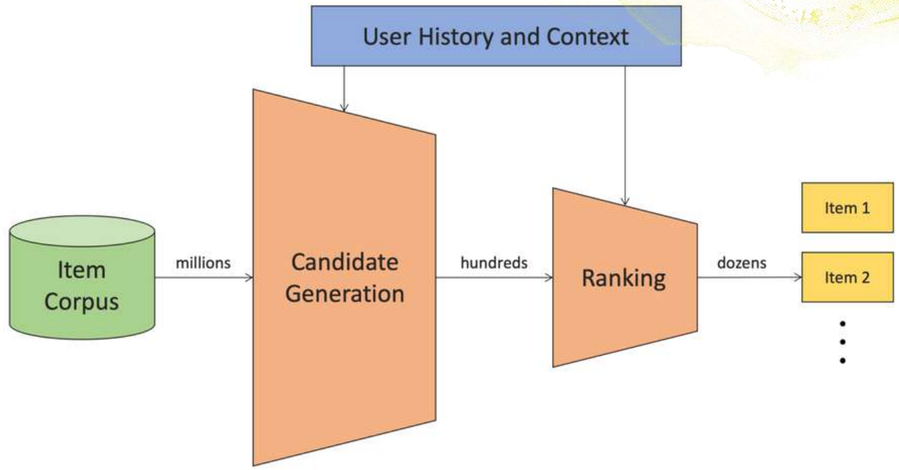
LISTWISE

$$\min_{\Theta} \sum_{u} \mathcal{L}(\pi_{\Theta}(u), \pi^*(u))$$

Three-stage pipeline: recall candidates, rank with CF/graph/sequence models, then re-rank under user intent. LLMs augment—not replace—scorers: semantic embeddings unify vector spaces, reasoning extracts constraints and explanations, and orchestration coordinates retrieval, ranking, and RAG grounding for controllable, traceable recommendations.

Image suggestion

Simple funnel diagram ("User History and Context → Candidate Generation → Ranking"). Good visual for Recall→Rank→Re-rank framing; add a "Re-rank" box in your slide.



賢 く 繋 ぎ 推 す

ASSUMPTION:

USER–ITEM PREFERENCES ARISE FROM A FEW LATENT FACTORS IN A SHARED LOW-DIMENSIONAL SPACE.

SCORING (MF)

SIMILARITY OF LATENT VECTORS DRIVES RATINGS.

TRAINING (EXPLICIT)

MINIMIZE MSE WITH L2 REGULARIZATION; OPTIMIZE VIA SGD OR ALS.

STRENGTHS

HIGH THROUGHPUT, STABLE OPTIMIZATION, SIMPLE TO SCALE AND COMBINE WITH SIDE FEATURES.

LIMITS

COLD-START PAIN, WEAK SEMANTICS, LONG-TAIL UNDEREXPOSURE, LIMITED INTERPRETABILITY.

TAKEAWAY

MF LEARNS CO-OCCURRENCE STRUCTURE; PAIR WITH LLMS TO INJECT SEMANTICS, CONSTRAINTS, AND EXPLANATIONS.

$$\hat{r}_{ui} = \mu + b_u + b_i + \mathbf{p}_u^\top \mathbf{q}_i$$

知 恵 で 薦 め る

BPR

$$\max_{\Theta} \sum_{(u,i,j)} \log \sigma(\hat{y}_{ui} - \hat{y}_{uj}) - \lambda |\Theta|_2^2$$

Graph (LightGCN)

$$\mathbf{E}^{(k+1)} = \tilde{\mathbf{A}}\, \mathbf{E}^{(k)} \quad \text{(propagation)}$$

$$\mathbf{e}_u = \sum_{k=0}^{K} \alpha_k \mathbf{E}_u^{(k)}, \quad \mathbf{e}_i = \sum_{k=0}^{K} \alpha_k \mathbf{E}_i^{(k)} \quad \text{(layer aggregation)}$$

$$\hat{y}_{ui} = \mathbf{e}_u^{\top} \mathbf{e}_i \quad \text{(prediction)}$$

EFFICIENT, CAPTURES HIGH-ORDER NEIGHBORS.

SEQUENCE (SASREC/GRU4REC)

NEXT-ITEM FROM ORDERED CLICKS; SELF-ATTENTION/RNN MODELS SHORT-TERM INTENT.

BRIDGE

STRONG STRUCTURE, WEAK SEMANTICS → PAIR WITH LLMS FOR INTENT/CONSTRAINTS/EXPLANATIONS.

SEMANTIC BLINDNESS: CF/GRAPH/SEQUENCE RELY ON CO-OCCURRENCE; WEAK AT UNDERSTANDING TEXT, IMAGES, ATTRIBUTES, AND NUANCED INTENT.

COLD-START & OOD: NEW USERS/ITEMS, UNSEEN VOCABULARY, CROSS-DOMAIN TRANSFER ALL DEGRADE.

COMPLEX CONSTRAINTS: BUDGETS, STYLES, COMPATIBILITY, POLICIES—HARD TO ENCODE AND ENFORCE IN TRADITIONAL SCORERS.

EXPLAINABILITY: HARD TO PRODUCE FAITHFUL, USER-FACING RATIONALES TIED TO EVIDENCE.

LONG-TAIL BIAS: POPULARITY AMPLIFICATION; LIMITED NOVELTY/DIVERSITY CONTROL.

SPARSE/TEMPORAL DRIFT: DATA SPARSITY, EVOLVING TASTES, SEASONALITY HURT STABILITY AND CALIBRATION.

MULTI-OBJECTIVE TRADE-OFFS: RELEVANCE VS. DIVERSITY, BUSINESS RULES, SAFETY—DIFFICULT TO BALANCE DECLARATIVELY.

PRIVACY & COMPLIANCE: SENSITIVE SIGNALS REQUIRE SELECTIVE USE, REDACTION, AND AUDITABLE DECISIONS.

LLM Reasoning

1. Introduction

- Expectation for AI: AI should be able to learn from just a few examples, like what humans usually do
- What is missing in ML?: Reasoning
  - Humans can learn from just a few examples because humans can reason
- LLMs
  - Training LLMs is like training parrots to mimic human languages
  - Few-shot prompting for last-letter-concatenation problem: Language models are few shot learners
  - Add "reasoning" before the "answer": one demonstration is enough, like humans

2. LLM Reasoning Key Idea

Regardless of training, fine-tuning, or prompting, when provided with examples that include intermediate steps, LLMs will respond with intermediate steps

Derive Final Answers through Intermediate Steps

Papers

- Training with intermediate steps (Ling et al. 2017)
  - Ling et al. Program Induction by Rationale Generation: Learning to Solve and Explain Algebraic Word Problems. ACL 2017
- Finetuning with intermediate steps (Cobbe et al. 2021, Nye et al. 2021)
  - Cobbe et al. Training Verifiers to Solve Math Word Problems. arXiv:2110.14168 [cs.LG]. 2021
  - Nye et al. Show Your Work: Scratchpads for Intermediate Computation with Language Models. arXiv:2112.00114 [cs.LG]. 2021
- Prompting with intermediate steps (Nye et al. 2021, Wei et al. 2022)
  - Jason Wei, et al. Chain-of-thought prompting elicits reasoning in large language models. NeurIPS 2022
  - Chain-of-Thought (CoT) Prompting

3. Why intermediate steps are helpful?

1. Transformer generating intermediate steps can solve inherently serial problems as long as its depth exceeds a constant threshold
2. Transformers generating direct final answers either requires a huge depth to solve or cannot solve at all.

Tons of practical implications of this theory

- Generating more intermediate steps (think longer)
- Too long to generate? Calling external tools e.g. MCTS
- Distill test-time search? You will need a huge depth!
- Which to choose: transformer vs RNN/SSM

Paper: Zhiyuan Li, Hong Liu, Denny Zhou, and Tengyu Ma. Chain of Thought Empowers Transformers to Solve Inherently Serial Problems. ICLR 2024

4. Is it also helpful to add reasoning strategies in demonstration examples?

Least-to-Most Prompting

- Enable easy-to-hard generalization by decomposition
- Teach the model to breakdown complex tasks into a sequence of simpler problems
- Denny Zhou, Nathanael Schärli, etal. Least-to-Most Prompting, Enables Complex Reasoning in Large Language Models. ICLR 2023.

Decomposing and Recombining are important operations of mind: Book: How to Solve it by G. Polya

SCAN (Compositional Generalization): Using just 0.1% demonstration examples achieves perfect generalization

CFQ (Compositional Generalization): Text-to-Code

- Andrew Drozdov, Nathanael Schärli,et al. Compositional Semantic Parsing with Large Language Models. ICLR 2023

5. Is it possible to trigger step by step reasoning without using demonstration examples?

Let's think step by step: Kojima, T., et al. Large language models are zero-shot reasoners. NeurIPS 2022.

Hmm, zero-shot is cool but usually significantly worse than few-shot ...: LLMs as Analogical Reasoners

- Analogical reasoning
  - adaptively generate relevant examples and knowledge, rather than just using a fix set of examples
- Paper: Michihiro Yasunaga, et al. Large Language Models as Analogical Reasoners. ICLR 2024

6. Is it possible to trigger step by step reasoning even without using any prompt like "let's think step by step"?

Chain-of-Thought Reasoning without Prompting

- Paper
  - Chain-of-Thought Decoding
  - Xuezhi Wang and Denny Zhou. Chain-of-Thought Reasoning Without Prompting. arXiv preprint arXiv:2402.10200 (2024).

Key takeaways

- Pre-trained LLMs, without further finetuning, has been ready for step-by-step reasoning, but we need a non-greedy decoding strategy to elicit it
- Pre-trained LLMs have had responses with step-by-step reasoning among the generations started with top-k tokens
- When a step-by-step reasoning path is present, LLMs have much higher confidence in decoding the final answer than direct-answer decoding
- Higher confidence in decoding the final answer when a step-by-step reasoning path is present

7. Any concern on generating intermediate steps instead of direct answers?

Self-Consistency

What LLM does in decoding:

$$\arg\max_{\text{reasoning path, final answer}} P(\text{reasoning path, final answer}|\text{problem})$$

What we want:

$$\arg\max_{\text{final answer}} P(\text{final answer}|\text{problem})$$

not aligned!

$$\arg\max_{\text{final answer}} P(\text{final answer}|\text{problem}) = \arg\max_{\text{final answer}} \sum_{\text{reasoning path}} P(\text{reasoning path}, \text{final answer}|\text{problem})$$

One step further

Sampling! How to compute the sum then?

Always keep in mind that LLMs are probabilistic models of generating next tokens. They are not humans.

Greatly improves step-by-step reasoning

More consistent, more likely to be correct

Definition: Requiring the model to produce multiple solutions and selecting the most consistent final answer

Questions

- When the LLM outputs a direct answer without intermediate steps, will you still sample several times, and then choose the most common answer?
- Change self-consistency by letting LLM generate multiple responses, instead of sampling multiple times, and then choosing the most common answer. Does this make sense?
  - No
  - $\arg\max_{\text{final answer}} P(\text{final answer}|\text{problem})$

Universal Self-Consistency (USC)

How about freeforms answers?

Ask LLMs to self-select the most consistent answer

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, Denny Zhou. Universal Self-Consistency for Large Language Model Generation. arXiv:2311.17311 [cs.CL], 2023. paper

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, Denny Zhou. Self-Consistency Improves Chain of Thought Reasoning in Language Models. ICLR 2023. Paper

8. Limitations

LLMs Can Be Easily Distracted by Irrelevant Context

Psychology studies show that irrelevant information may significantly decrease some children and even adults problem-solving accuracy

Does this observation hold for LLMs?: Humans may be easily distracted by irrelevant context

Instructing LLMs to Ignore irrelevant context can take some performance back

We can simply add many irrelevant sentences

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael Schärli, and Denny Zhou. Large Language Models Can Be Easily Distracted by Irrelevant Context. ICML 2023 Paper

LLMs Cannot Self-Correct Reasoning Yet

While allowing LLMs to review their generated responses can help correct inaccurate answers, it may also risk changing correct answers into incorrect ones

- Self-correcting results in worse results
- Reported improvements need oracle answers
- Oracle: Let LLMs self correct only when the answer is wrong

Worse than self-consistency! Multi-LLM debate?

Oracle feedback needed for LLM to self-correct: Lesson learned

Xinyun Chen, Maxwell Lin, Nathanael Schärli, Denny Zhou. Teaching Large Language Models to Self-Debug. arXiv:2302.00093 [cs.CL], 2023. ICLR 2024. Paper

Self-debug naturally leverage unit tests as oracle

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, Denny Zhou. Large Language Models Cannot Self-Correct Reasoning Yet. ICLR 2024. Paper

Premise Order Matters in LLM Reasoning

drop 10 points in accuracy across all frontier LLMs: Re-ordering a prompt

Logical Inference tasks

- The rules are ordered according to their use in the inference process
- If the rules relevant to the query are randomly ordered, 30+ points performance drop across all frontier LLMs

9. Summary

Training / finetuning / prompting with intermediate steps

Zero-shot, analogical reasoning, special decoding: Generating intermediate steps improves LLM performance

Self-consistency greatly improves step-by-step reasoning

Limitation

- irrelevant context
- self-correction
- premise order

10. What is next?

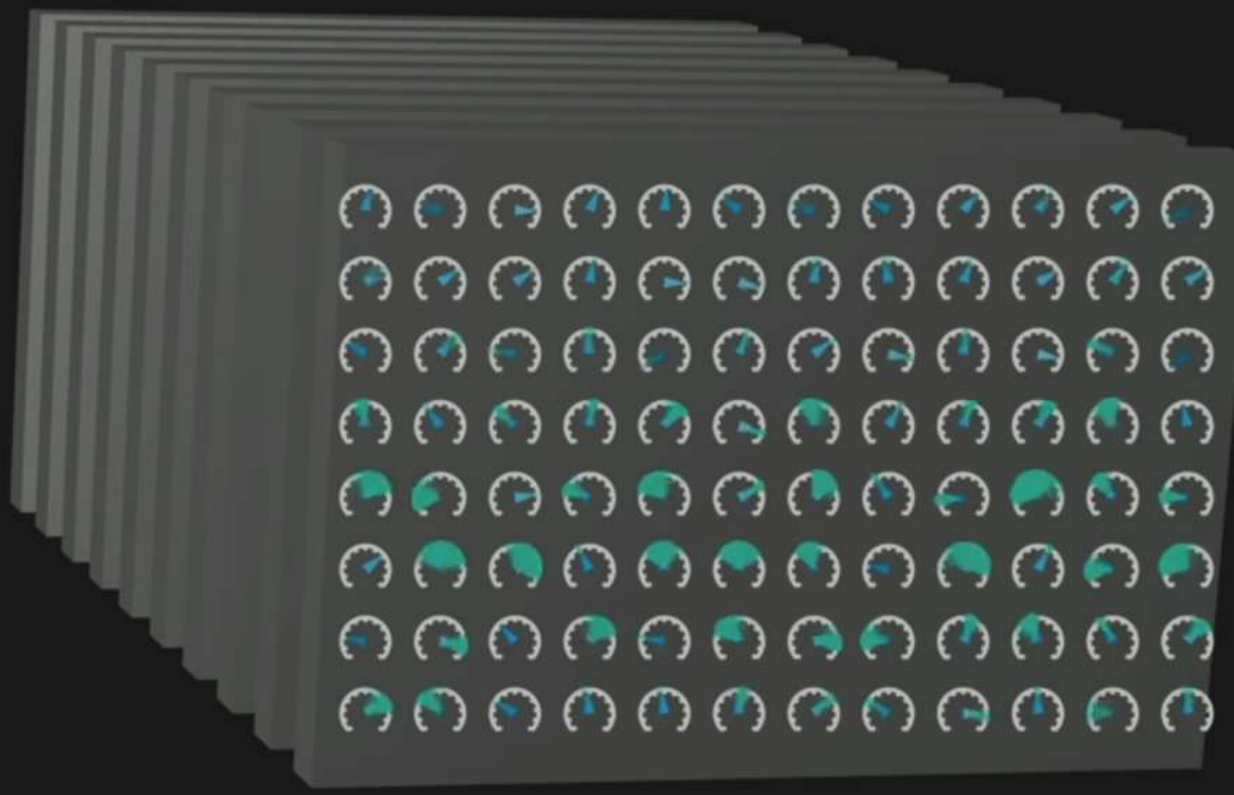
If I were given one hour to save the planet, I would spend 59 minutes defining the problem and one minute resolving it.
— Albert Einstein

1. Define a right problem to work on
2. Solve it from the first principles

"The truth always turns out to be simpler than you thought."
— Richard P. Feynman

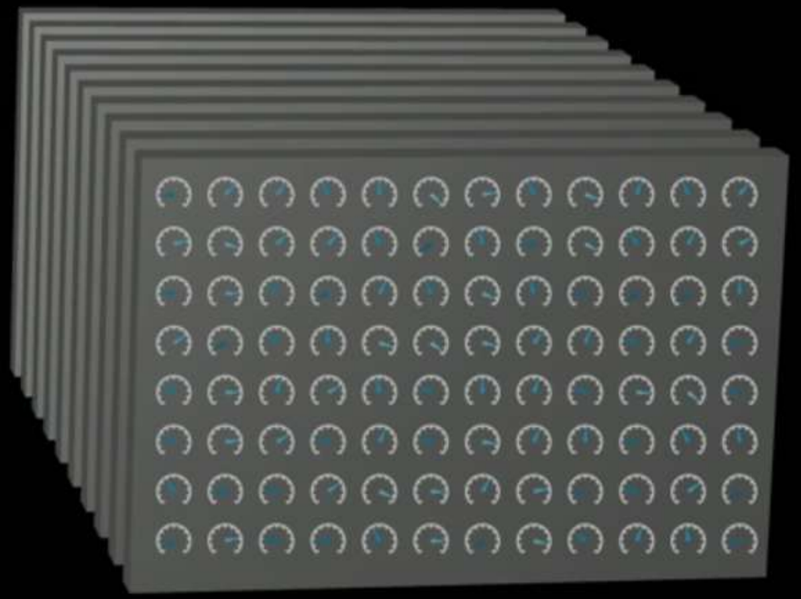
"The best way to predict the future is to invent it." — Alan Kay

Develop a model that autonomously learns all the reasoning techniques we have introduced while addressing all the limitations we have identified



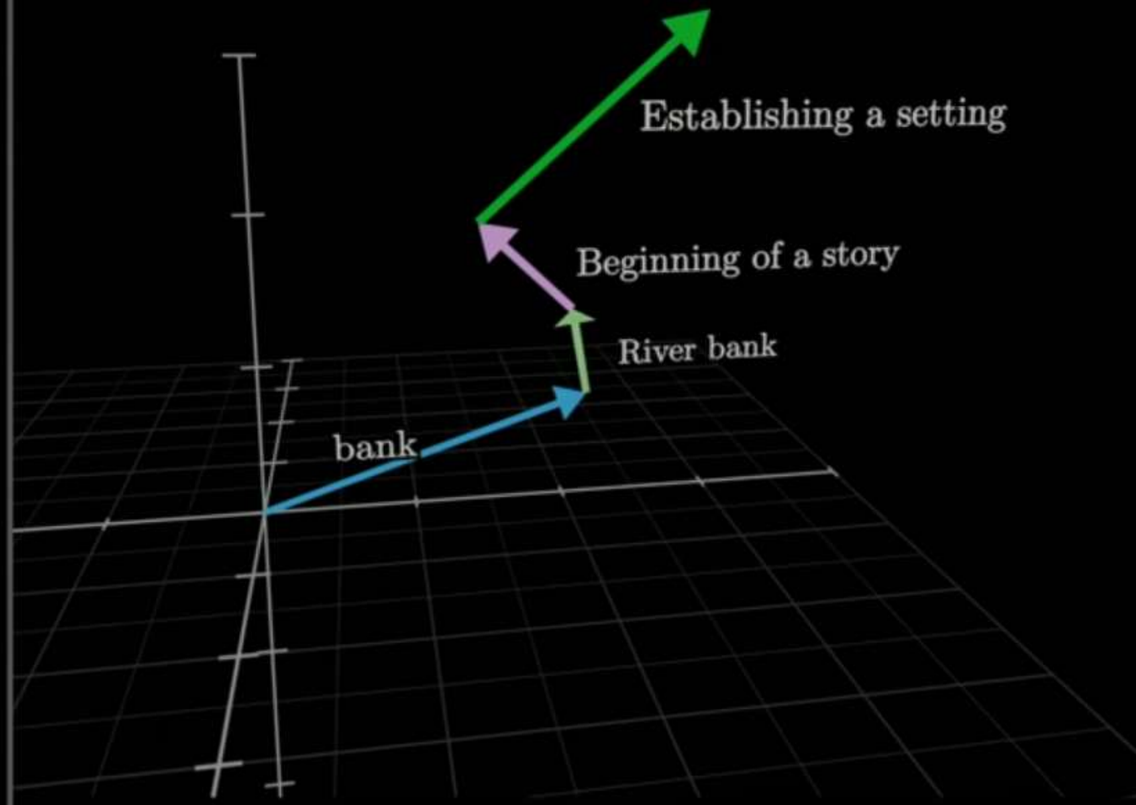
It was the best of times it was the worst

It was the best of times it was the _____ →



worst	10%
age	87%
worse	1%
best	0%
most	0%
end	0%
very	0%
blur	0%
⋮	

Down by the river bank



Theoretical Summary

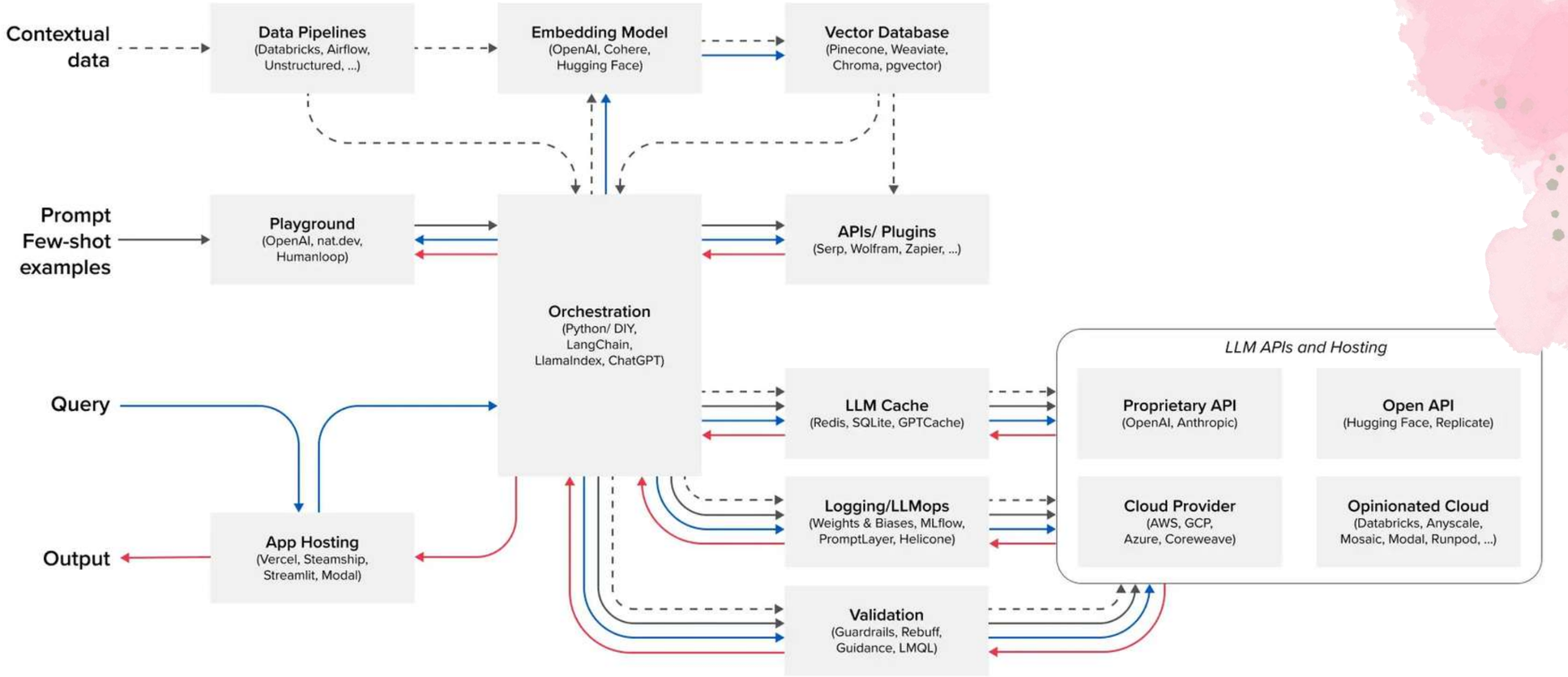
LLMs enhance recommendation pipelines by unifying semantics across recall, modeling, and ranking.

Candidate Generation: Texts (titles, reviews) are embedded into a shared vector space for ANN retrieval, combining sparse BM25 and dense vectors to improve findability.

User Modeling: LLMs summarize history and dialogue into structured JSON constraints (e.g., price_range, must_have).

Re-ranking: Using these constraints, LLMs re-rank candidates with interpretable scores and rationales, balancing relevance, diversity, and controllability.

Emerging LLM App Stack



LEGEND

Gray boxes show key components of the stack, with leading tools/systems listed

Arrows show the flow of data through the stack

- Contextual data provided by app developers to condition LLM outputs
- Prompts and few-shot examples that are sent to the LLM
- Queries submitted by users
- Output returned to users