Practical Recommender Systems: From Real-World Challenges to Graph Intelligence

Makbule Gulcin Ozsoy, Neo4j
March 2026

About Me

Makbule Gulcin Ozsoy – Software/Machine Learning Engineer @ Neo4j

Past Experience

- PhD & Postdoc: Recommender systems and ranking models
- Work experience: Built production-grade recommender system pipelines

Current Work

- NLP for graph databases (Text2Cypher, natural language → Cypher queries)
- Exploring natural language interfaces to democratize access to complex graph data

LinkedIn: linkedin.com/in/makbulegulcinozsoy

Outline

- **Recommender Systems**
  - Foundations and Advancements
  - Recommenders in Production

- **Recommender Systems and Graphs**
  - Recommenders As Graph Problems
  - Graph-powered Recommendations

- **Accessing Data on Graphs**
  - Text2Cypher: Natural language to Cypher query language
  - Challenges and opportunities

- **Emerging LLM & Graph Applications**
  - Graph construction from unstructured data
  - Retrieval-Augmented Generation (RAG) on graphs
  - Other research directions (graph reasoning, dynamic graphs)

- **Conclusion and Discussion**

Outline

- **Recommender Systems**
  - Foundations and Advancements
  - Recommenders in Production

- **Recommender Systems and Graphs**
  - Recommenders As Graph Problems
  - Graph-powered Recommendations

- **Accessing Data on Graph**
  - Text2Cypher: Natural language to Cypher query language
  - Challenges and opportunities

- **Emerging LLM & Graph Applications**
  - Graph construction from unstructured data
  - Retrieval-Augmented Generation (RAG) on graphs
  - Other research directions (graph reasoning, dynamic graphs)

- **Conclusion and Discussion**

Recommender Systems

- **Artificial intelligence (AI) + Personalization**

- Tailored experiences
  → More **personalized** + **engaging** + **efficient**

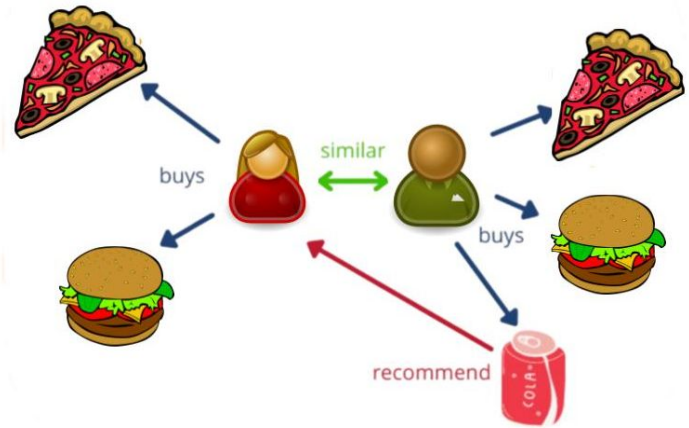
  - Video recommendation (e.g., Youtube)
  - Product recommendation (e.g., Amazon)
  - Music track recommendation (e.g., Spotify)
  - ...

References
1) https://vectorinstitute.ai/recommender-systems-where-academia-meets-industry/

Recommender Systems

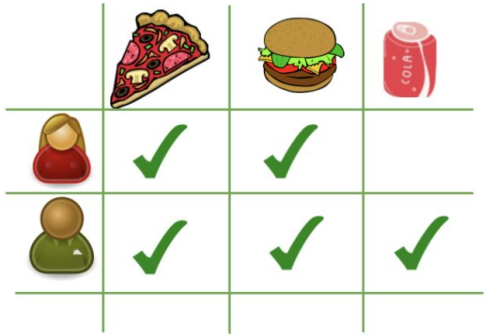
- **Input data:**
  - User Profiles:
    - Examples: Demographics, past interactions
  - Item Characteristics:
    - Examples: Genre, price, color
  - Contextual information:
    - Examples: Temporal features, location

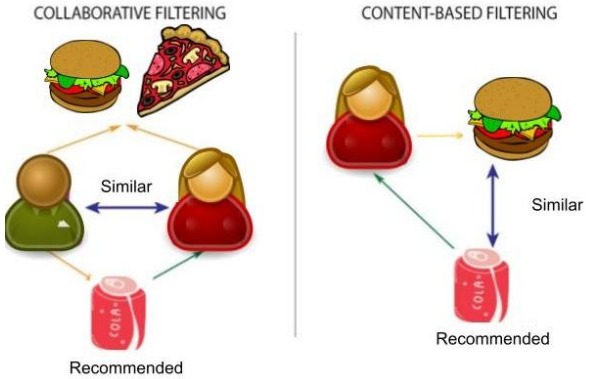
- **Feedback/Labels:**
  - Explicit (e.g., ratings)
  - Implicit (e.g., clicks)



Recommender Systems

- **Traditional methods**
  - **Collaborative filtering (CF)**
    - Based on similarities among users or items
    - Challenges: Cold-start problem
  - **Content-based filtering (CBF)**
    - Based on item attributes and user preferences
    - Challenges: Novelty, diversity
  - **Hybrid**
    - Combination of CF and CBF

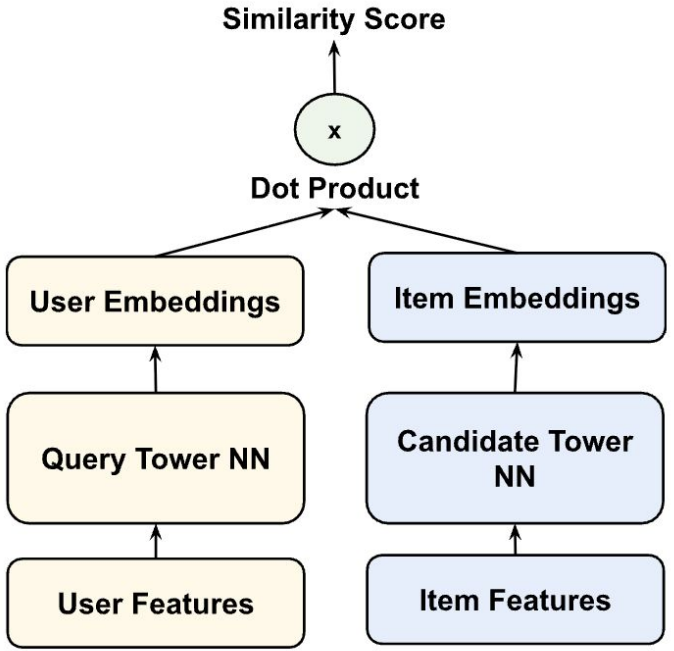




References
1) https://vectorinstitute.ai/recommender-systems-where-academia-meets-industry/

Recommender Systems

- **Deep Learning-based methods**
  - **(Earlier) Neural Networks**
    - Examples:
      Two-Tower architecture,
      Deep-and-Wide architecture,
      Mixture-of-Experts,
      Graph NN
  - **Transformers**
    - Example: BERT4Rec
  - **Challenges:**
    - High computational cost
    - Data/Label sparsity
    - Feature overfitting
    - Lack of interpretability



References

1) https://vectorinstitute.ai/recommender-systems-where-academia-meets-industry/
2) https://www.shaped.ai/blog/the-two-tower-model-for-recommendation-systems-a-deep-dive
3) https://medium.com/@kyang3200/deep-learning-wide-deep-learning-for-recommender-systems-47c6119c622d
4) https://blog.reachsumit.com/posts/2023/04/moe-for-recsys/

Recommender Systems

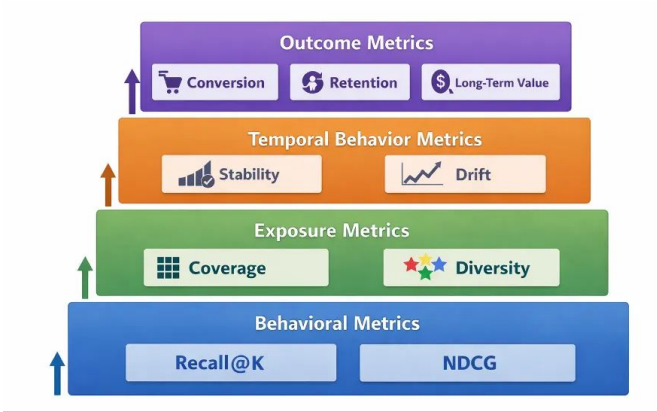
- **LLM-based methods**
  - **Textify Interactions**
    - Convert user histories to text sequences and train Transformers directly for next-item prediction
    - Examples: CALRec, beeFormer
  - **Align Id ↔ Text Models**
    - Train Id-based rankers and text models together by reconstructing masked data across both formats
    - Example: Huawei FLIP
  - **Content → Semantic Ids**
    - Turn rich content (video, text, audio) to compact, semantic ids instead of random item ids
    - Examples: YouTube, TikTok
  - **Unified Search+Recommender systems**
    - Single foundation models handle search, recommendations, and ads across many tasks
    - Examples: LinkedIn 360Brew, Netflix UniCoRn

References

1) https://eugeneyan.com/writing/recsys-llm/
2) https://lfaidata.foundation/communityblog/2025/08/25/leverage-llm-for-next-gen-recommender-systems-technical-deep-dive-into-llm-enhanced-recommender-architectures/
3) https://amatria.in/blog/recsyskeynote
4) Lee, Dong-Ho, et al. "Star: A simple training-free approach for recommendations using large language models." arXiv preprint arXiv:2410.16458 (2024).

Recommender Systems

- **Evaluation of Recommender Systems**
  - **Offline**
    - **Accuracy metrics (Behavioral)**
      - Not only accuracy:
        - E.g. Precision, Recall
      - But also ranking:
        - E.g., NDCG, Precision@k
    - **Exposure metrics**
      - Coverage, Diversity, Novelty
    - **Temporal metrics**
      - Stability, drift
  - **Online**
    - (All) Offline metrics
    - **Business metrics**
      - Long-term value, conversion, churn

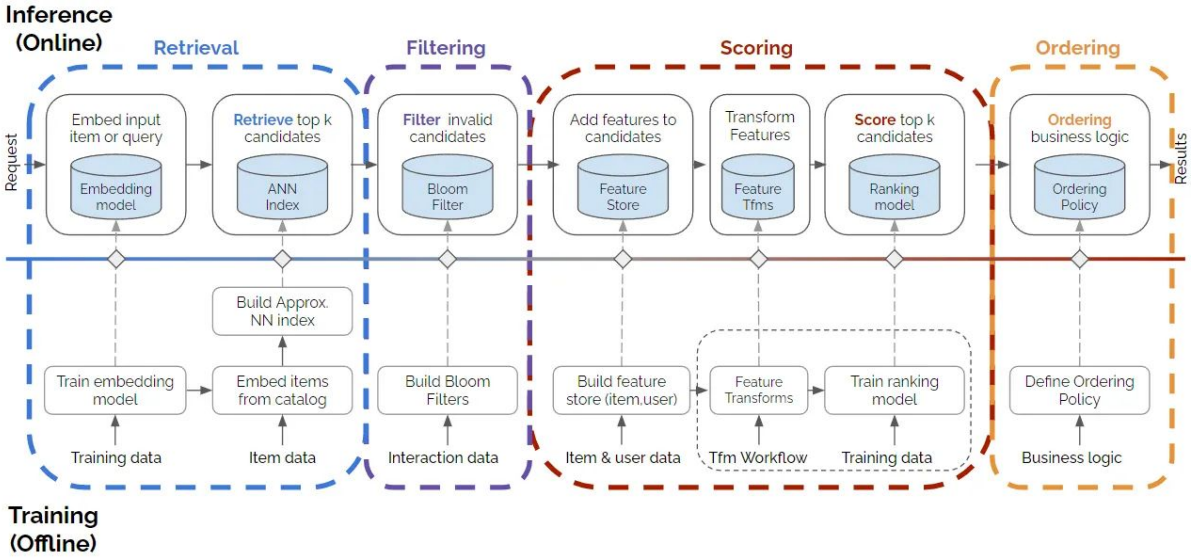


References

1) Evaluating Recommender Systems Beyond Accuracy, 2026, https://pub.towardsai.net/evaluating-recommender-systems-beyond-accuracy-3ca49c10f620

Recommender Systems

- **Not just a model → But a system**
  - Example from NVIDIA Merlin (2022)

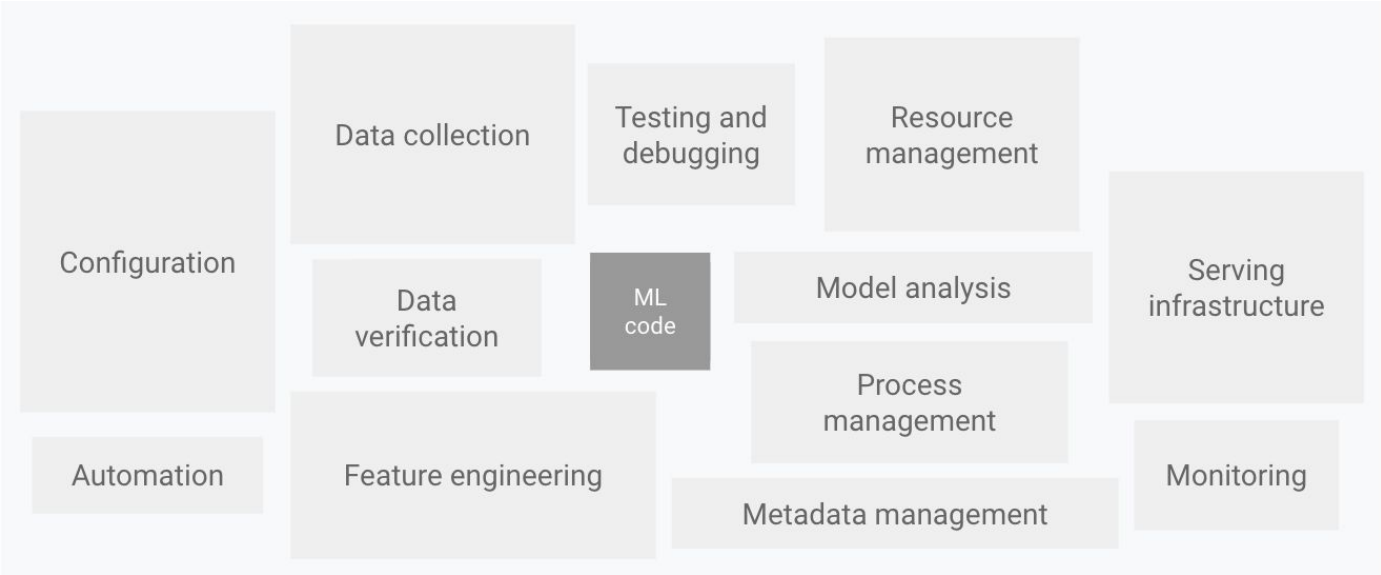


References

1) https://medium.com/nvidia-merlin/recommender-systems-not-just-recommender-models-485c161c755e
2) https://www.databricks.com/blog/guide-to-building-online-recommendation-system

Recommender Systems

- **Not just a model → But a system**

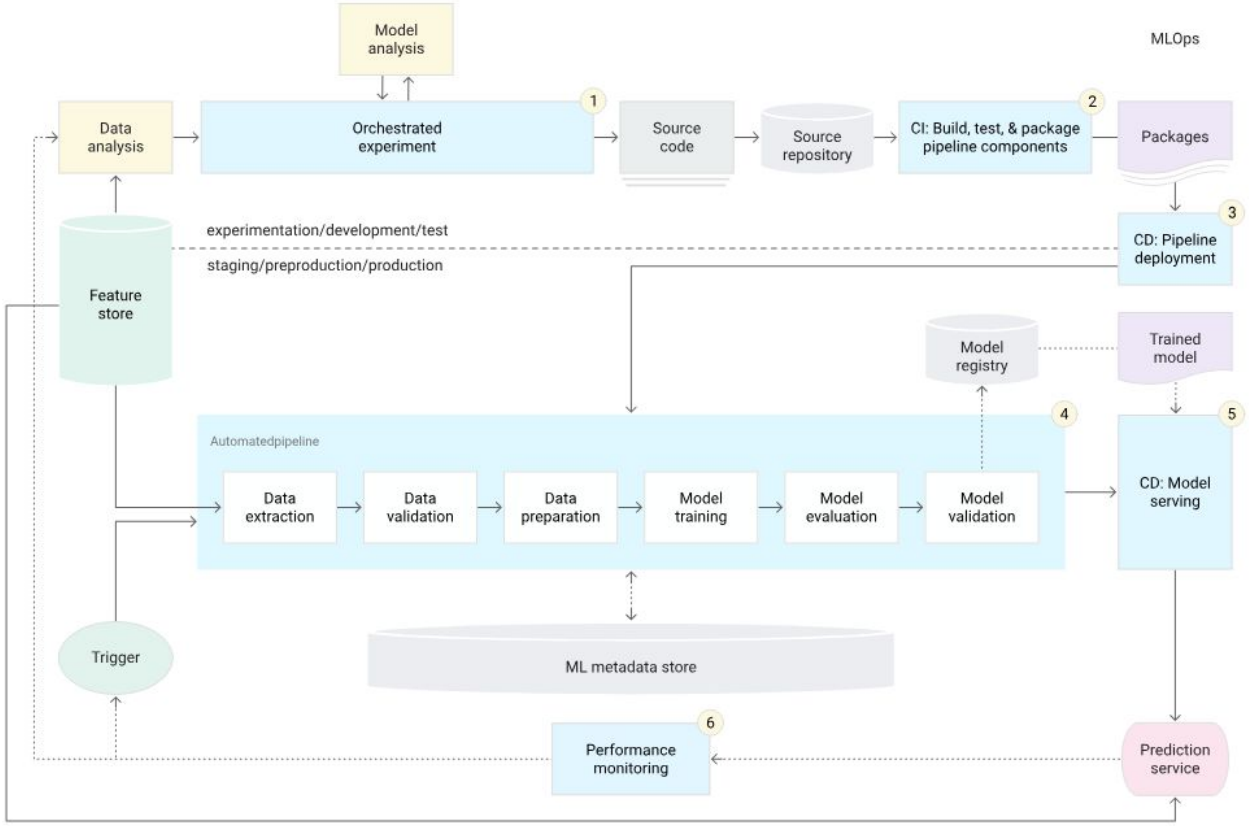


References

1) https://docs.cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning

Recommender Systems

- **Not just a model**
  → But a system



References

1) https://docs.cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning

Recommender Systems

- **Common Challenges of Recommender Systems (Algorithm/Model Level)**
  - **Scale and performance**
    - Huge item catalogues
    - High computational cost
    - Multi-modal data
  - **Business & Ethics**
    - Businness logic requirements
      - E.g., Out of stock item, age appropriateness
    - Privacy
    - Fairness
  - **Data & Quality**
    - Cold-start problem
    - Real-time adaptation
    - Diversity
    - Novelty
    - Explainability

References
1) https://vectorinstitute.ai/recommender-systems-where-academia-meets-industry/
2) https://medium.com/nvidia-merlin/recommender-systems-not-just-recommender-models-485c161c755e

Recommender Systems

- **Common Challenges of Recommender Systems (System/Production Level):**
  - **Lack of Result Reproducibility**
    - **Same** code, data, and parameters → **Different** results
    - Solution:
      - **Repeatable pipeline**
      - **Consistent environment** for training and serving
  - **Offline-Online Performance Mismatch**
    - **Good offline** evaluation metrics → **Not** observed in **online** environment
    - Solution:
      - Check **assumptions**:
        - Offline assumes static, unbiased behavior
        - Live systems create feedback loops
      - **Align** offline metrics with **business objectives**
      - Evaluate **beyond accuracy** (diversity, coverage, long-term value)

References
1) Challenges of Building a Recommender System
2) Netflix's Raising a Recommender System

Recommender Systems

- **Common Challenges of Recommender Systems (System/Production Level):**
  - **Oscillating Outputs**
    - **Quality** changes over **time**
      - E.g., **Distribution shifts** or **sampling bias**
    - Solution:
      - Biases or fluctuations in the training **data**
      - Check training **delays**, i.e., data collection vs. training time
      - Monitor **exposure** distribution and **feedback** effects
  - **Adding new features or data drops performance**
    - **New data/feature** improves performance (in **offline**) → **Performance drop** (in **online**)
    - Solution:
      - Check objective function, labels, data, features, model architecture

References
1) Challenges of Building a Recommender System
2) Netflix's Raising a Recommender System

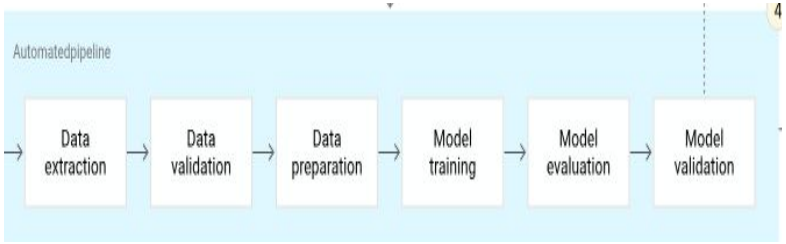
Recommender Systems

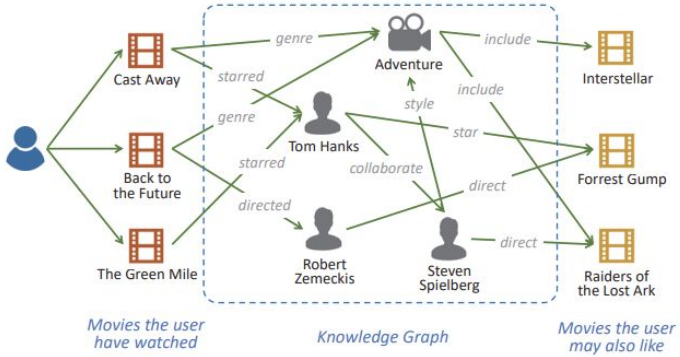
- Returning back to to the "**Development and experimentation**" stage

- Explore alternatives to
  - Represent data more efficiently
  - Explore multi-hop interactions
  - Provide reasoning or explanations

  ...

→ Graph representation & Graph intelligence





References

1) https://docs.cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning
2) https://medium.com/data-science/addressing-the-recommendation-task-through-a-different-lens-3226998bad58
3) Wang, Hongwei, et al. "Ripplenet: Propagating user preferences on the knowledge graph for recommender systems." Proceedings of the 27th ACM international conference on information and knowledge management. 2018

Outline

- **Recommender Systems**
  - Foundations and Advancements
  - Recommenders in Production

- **Recommender Systems and Graphs**
  - Recommenders As Graph Problems
  - Graph-powered Recommendations

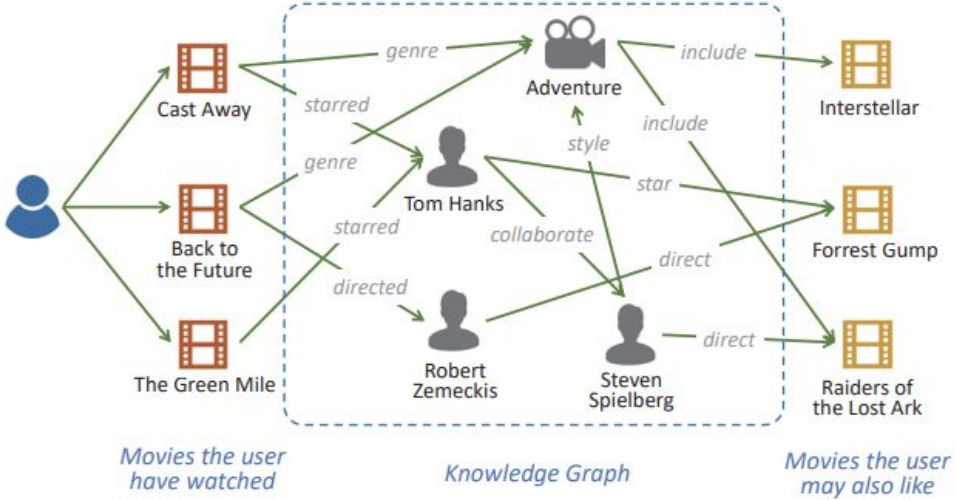
- **Accessing Data on Graph**
  - Text2Cypher: Natural language to Cypher query language
  - Challenges and opportunities

- **Emerging LLM & Graph Applications**
  - Graph construction from unstructured data
  - Retrieval-Augmented Generation (RAG) on graphs
  - Other research directions (graph reasoning, dynamic graphs)

- **Conclusion and Discussion**

Recommender Systems and Graphs

- **Knowledge graphs**
  - Directed heterogeneous graph
  - Structured format
    - Nodes (entities)
    - Relations (edges)
    - Their attributes
  - Relationships as fundamental data elements

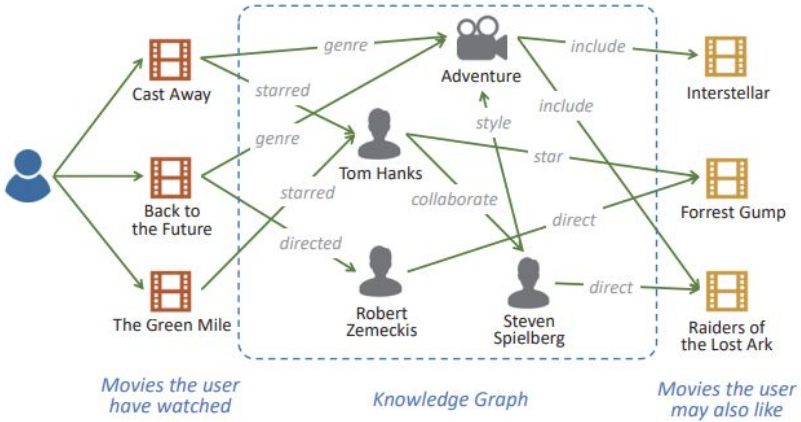


References

1) Wang, Hongwei, et al. "Ripplenet: Propagating user preferences on the knowledge graph for recommender systems." Proceedings of the 27th ACM international conference on information and knowledge management. 2018

Recommender Systems and Graphs

- **Graphs ↔ Recommender Systems**
  - **Data**
    - Represent data more efficiently
    - Provide semantic relationships
  - **(Real-time) Exploration**
    - Multi-hop interactions
    - Diverse set of relations
    - Pattern matching
  - **Explainability**
    - Provide path-based reasoning
    - Provide explanations

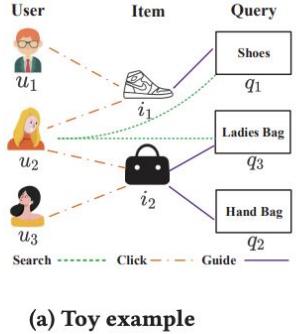


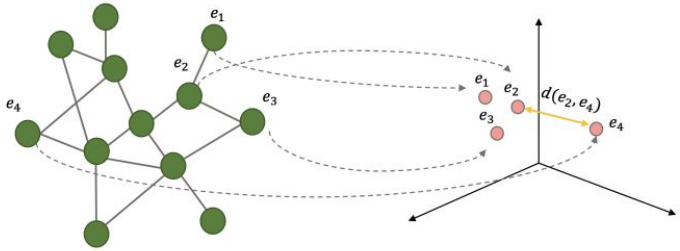
References

1) Wang, Hongwei, et al. "Ripplenet: Propagating user preferences on the knowledge graph for recommender systems." Proceedings of the 27th ACM international conference on information and knowledge management. 2018

Recommender Systems and Graphs

- **Techniques**
  - **Path-based methods**
    - Paths that exist between entities
    - Allows for multi-step capture of relationships
    - Incorporates richer contextual information
  - **Embedding-based methods**
    - Embeddings of users and items
    - Learned based on user-item interaction history
    - Predicts potential or underlying relationships and patterns



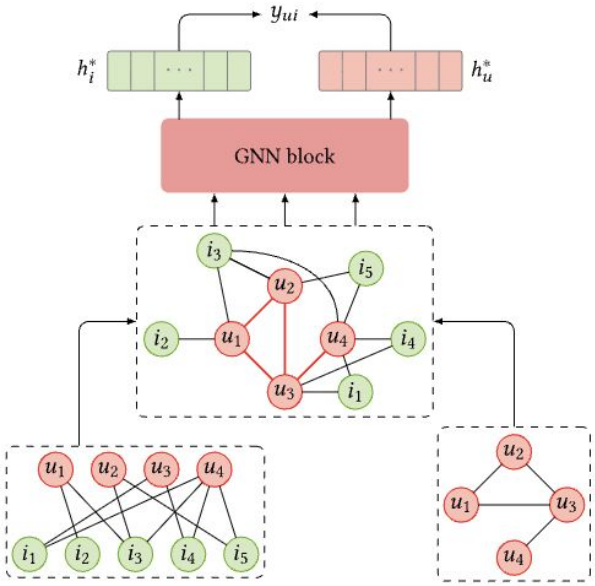


References

1) Fan, Shaohua, et al. "Metapath-guided heterogeneous graph neural network for intent recommendation." Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019.
2) Sakong, Darnbi. "Advanced Applications with Complex Graph Embedding, Hypergraph Graph Embedding, and Hypergraph Diffusion." (2025).
3) Sakurai, Keigo, et al. "Llm is knowledge graph reasoner: Llm's intuition-aware knowledge graph reasoning for cold-start sequential recommendation." *European Conference on Information Retrieval*. Cham: Springer Nature Switzerland, 2025.

Recommender Systems and Graphs

- **Techniques**
  - **Graph Neural Networks (GNN)**
    - Neural models that learn representations directly on graphs
    - Aggregate information from connected nodes and multi-hop neighborhoods
    - Capture complex, higher-order relationships in data
    - Naturally incorporate heterogeneous entities (users, items, categories, context)

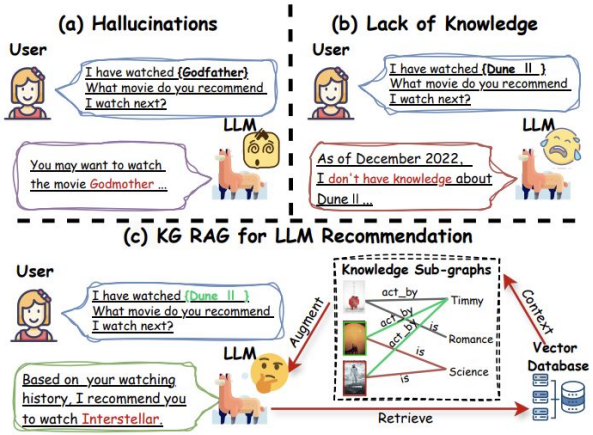


References

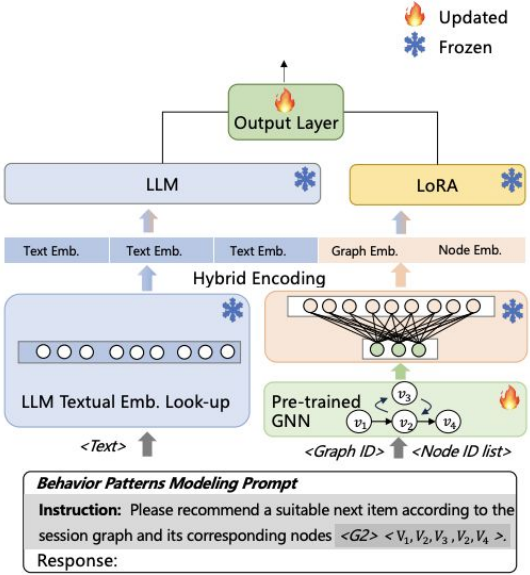
1) Wu, Shiwen, et al. "Graph neural networks in recommender systems: a survey." ACM Computing Surveys 55.5 (2022): 1-37.
2) Raza, Shaina, et al. "A comprehensive review of recommender systems: Transitioning from theory to practice." Computer Science Review 59 (2026): 100849.

Recommender Systems and Graphs

- **Techniques**
  - **Graph and LLM-based methods**
    - Graph as a source of truth
      - RAG, Grounding



  - **Hybrid GNN & LLM-based Recommendation**

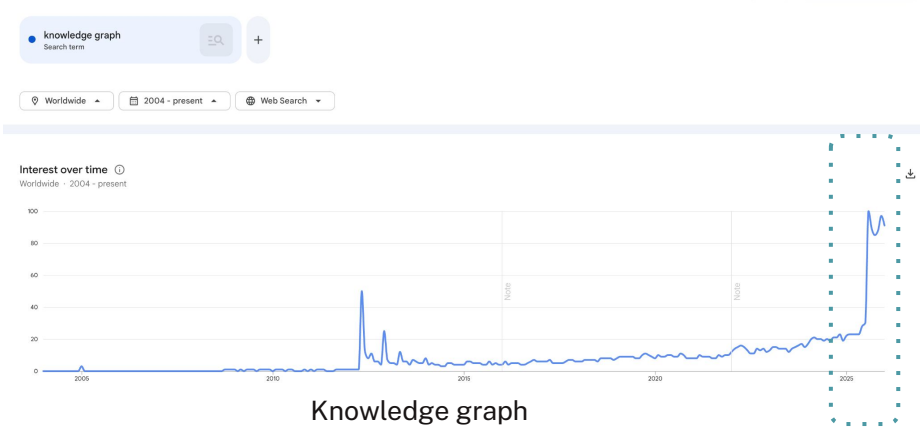


References

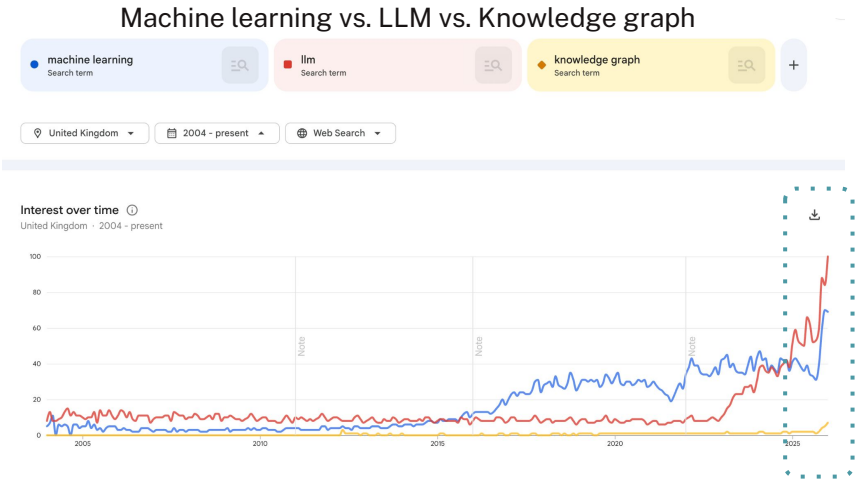
1) Wang, Shijie, et al. "Knowledge graph retrieval-augmented generation for llm-based recommendation." Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2025.
2) Guo, Naicheng, et al. "Integrating large language models with graphical session-based recommendation." arXiv preprint arXiv:2402.16539 (2024).

Recommender Systems and Graphs

- **Knowledge graphs beyond recommendations**
  - Getting more attention nowadays

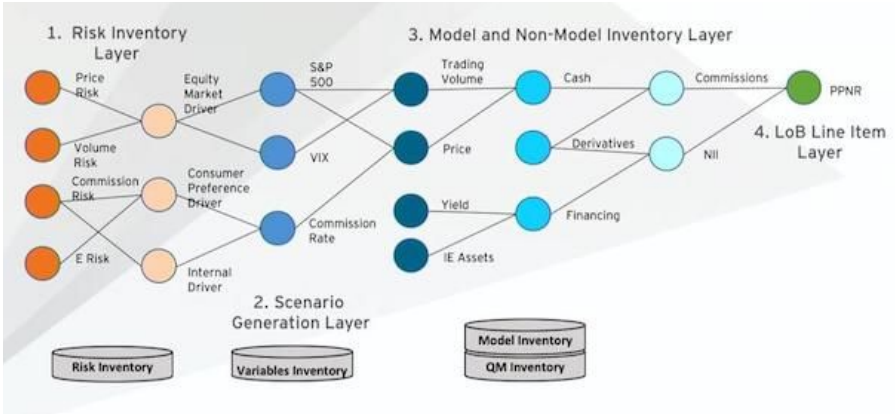


Knowledge graph



Recommender Systems and Graphs

- **Knowledge graphs beyond recommendations**
  - Bussiness Intelligence
  - Cybersecurity and threat detection
  - Fraud-detection and investigation
  - Social network anlaysis
  - Drug discovery & bio-informatics
  - Enterprise AI/LLM applications
    - Grounding
    - Explanations



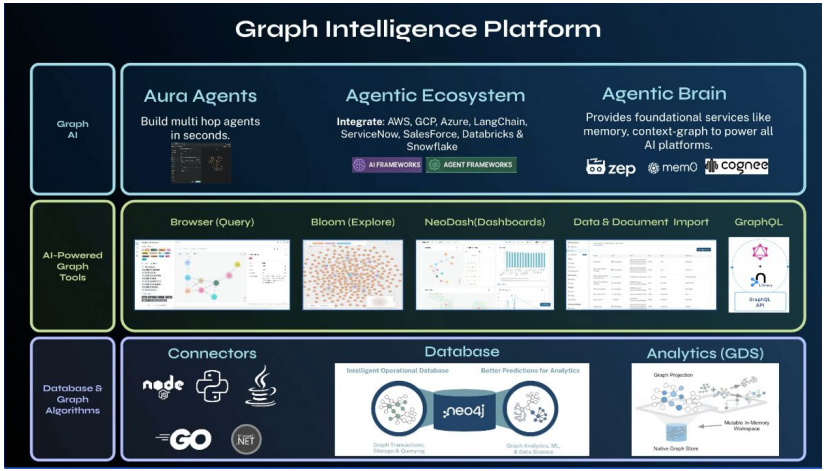
References

1) https://www.puppygraph.com/blog/intelligence-graph#:~:text=An%20intelligence%20graph%20is%20a,social%20networks%2C%20and%20knowledge%20graphs
2) https://neo4j.com/blog/developer/mastering-fraud-detection-temporal-graph/
3) https://neo4j.com/blog/knowledge-graph/knowledge-graphs-path-to-enterprise-ai/

Recommender Systems and Graphs

- **Operationalizing Knowledge Graphs**
  - Need to store large, evolving graphs efficiently
  - Need fast multi-hop traversals
  - Need flexible schemas for changing data

  - Need a platform built for connected data and intelligence



Outline

- **Recommender Systems**
  - Foundations and Advancements
  - Recommenders in Production

- **Recommender Systems and Graphs**
  - Recommenders As Graph Problems
  - Graph-powered Recommendations

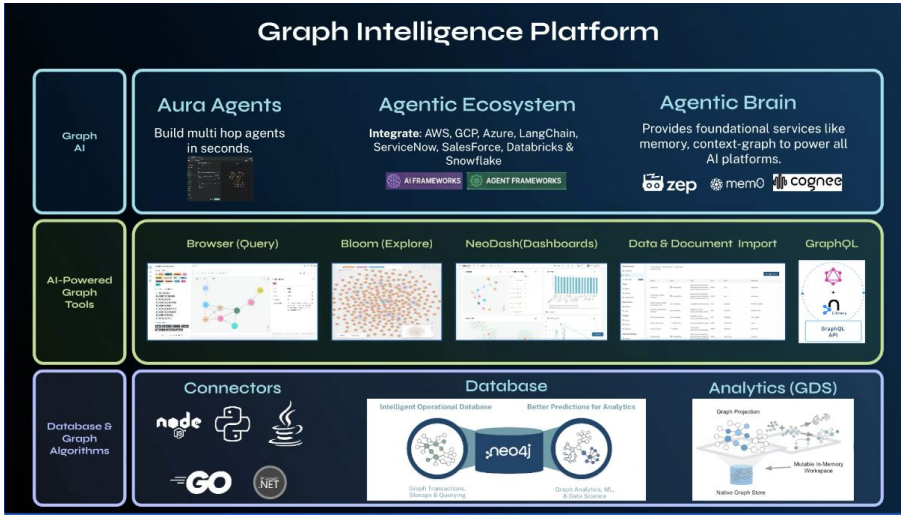
- **Accessing Data on Graph**
  - Text2Cypher: Natural language to Cypher query language
  - Challenges and opportunities

- **Emerging LLM & Graph Applications**
  - Graph construction from unstructured data
  - Retrieval-Augmented Generation (RAG) on graphs
  - Other research directions (graph reasoning, dynamic graphs)

- **Conclusion and Discussion**

Accessing Data on Graph

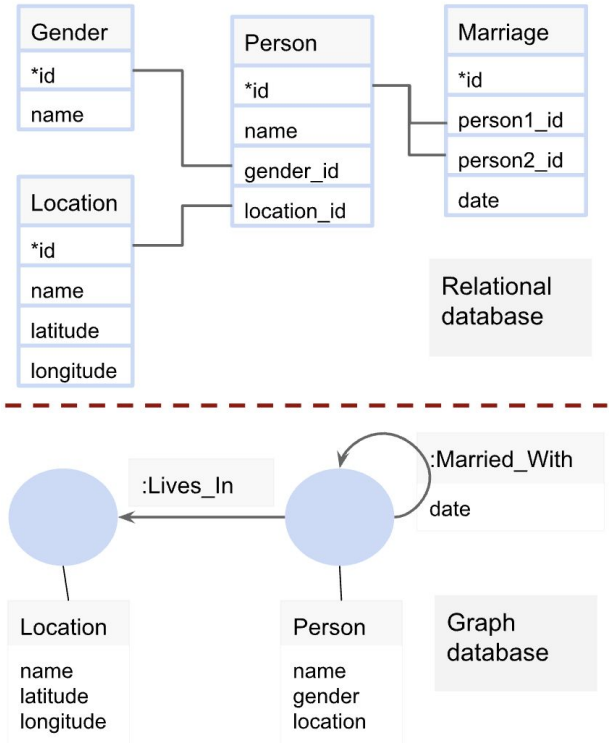
- **Neo4j as Graph Intelligence Platform**
  - Native graph storage:
    - Nodes & relationships as first-class entities
  - Optimized for multi-hop traversals and evolving graphs
  - Supports flexible schemas for dynamic, connected data



Accessing Data on Graph

Graph Intelligence

- One aspect is:
  - Data storage and Knowledge management
  - E.g., Databases
- Accessed via domain-specific languages
  - SQL (for relational databases)
  - SPARQL (for RDF graphs)
  - Cypher (for graph databases)



Cypher query language

- Used by Neo4j
- Efficient modeling and querying knowledge graphs
- Powerful, but requires learning its syntax

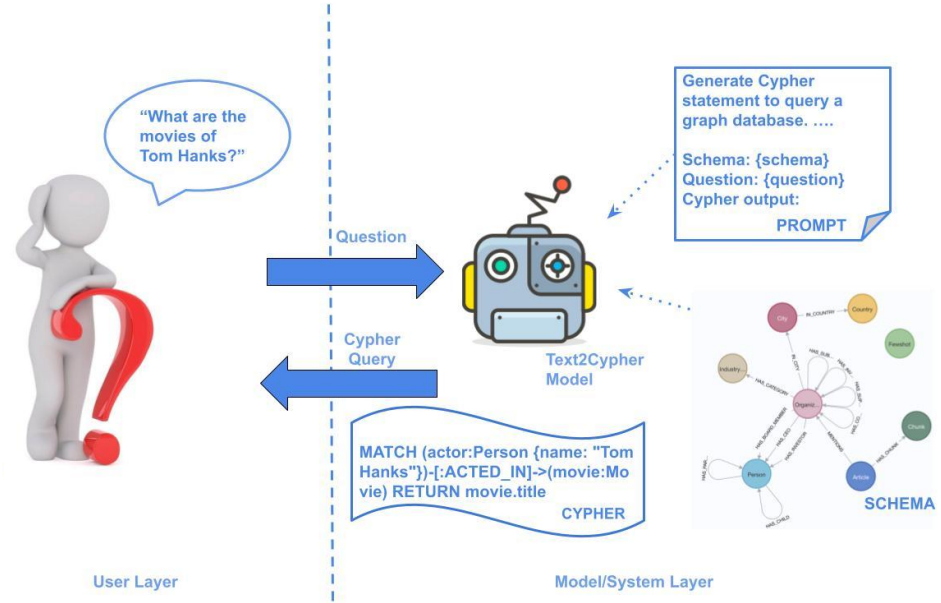
Accessing Data on Graph - Text2Cypher

Text2Cypher

Letting anyone **query the graph** using **plain language**

Application areas

- Direct query generation
  - “Write me the Cypher query.”
- GraphRAG
  - Retrieve relevant data
- UI components and dashboards
  - Data retrieval for analytics
- …



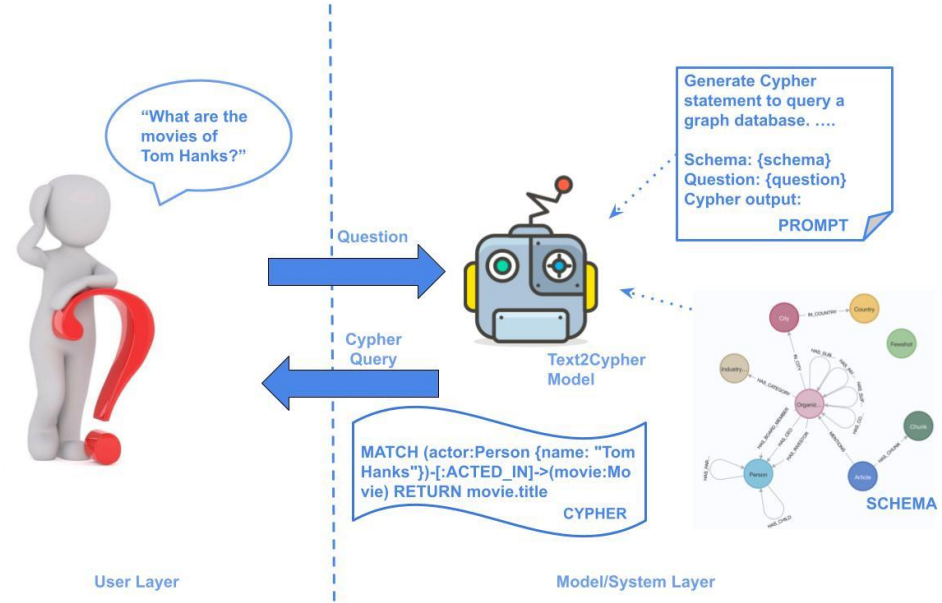
Accessing Data on Graph - Text2Cypher

Text2Cypher:

Translating **natural language** into **Cypher** query

Components

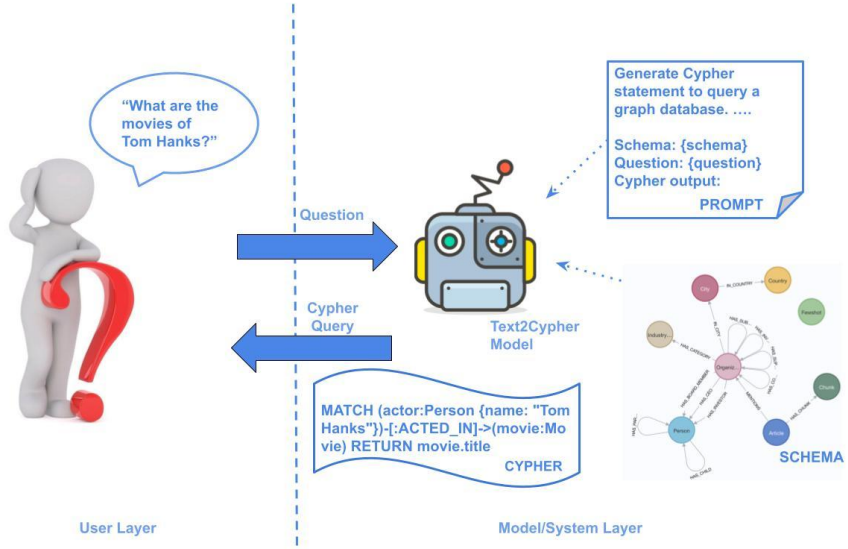
- User question
- Prompts
- Database schema
- Model



Accessing Data on Graph - Text2Cypher

Each component of Text2Cypher can be explored from various angles

- **User question**
  - Question as is vs. Re-phrasing
  - Questions in English vs. Other languages
- **Prompts**
  - Zero-shot vs. Few-shot vs. Reasoning
- **Database schema**
  - All schema vs. Filtered/Pruned schema
- **Model**
  - Foundational vs. Finetuned model



Accessing Data on Graph - Text2Cypher

What we have done so far

- **Datasets**
  - Text2Cypher-2024v1: HF/neo4j, Link
    - Collected and combined instances from publicly available datasets
    - Cleaned and organized them for smoother use
  - Text2Cypher-2025v1: HF/neo4j, Link
    - A slightly cleaned version of Text2Cypher-2024v1

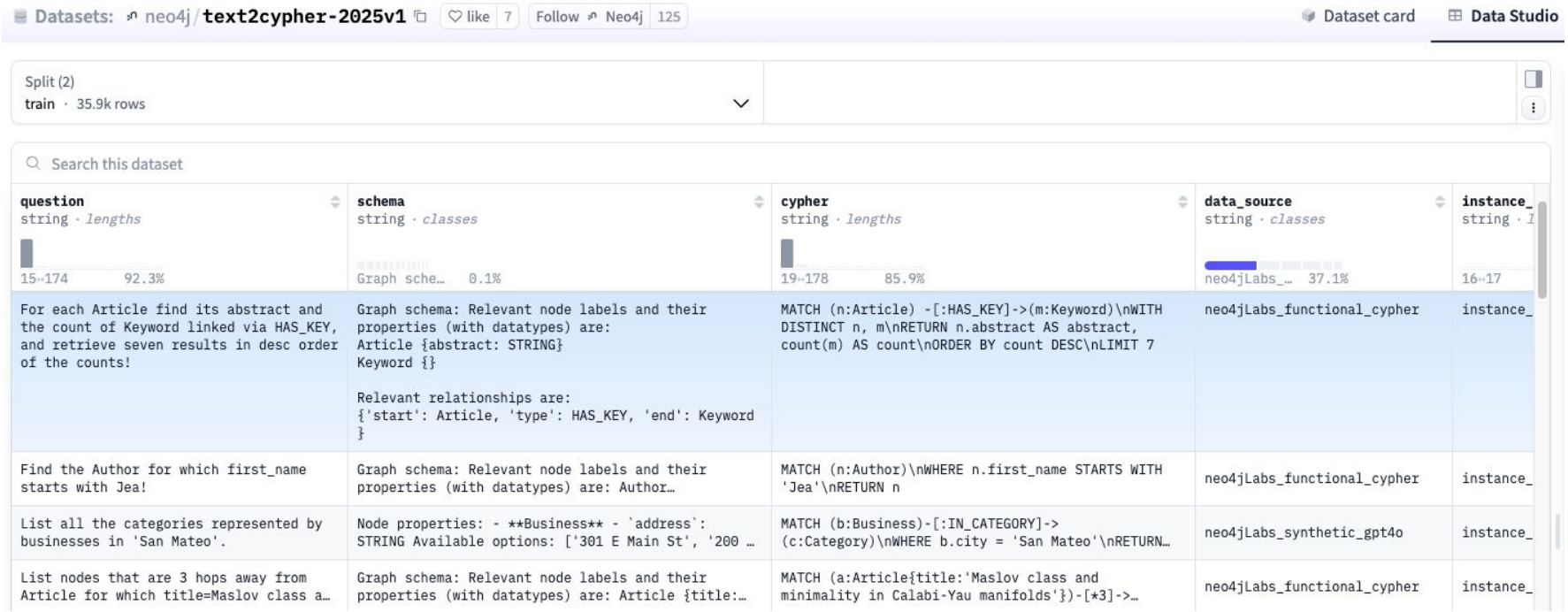
- **Models**
  - Finetuned-Gemma2: HF/neo4j, Link
  - Finetuned-Gemma3-4B: HF/neo4j, Link
  - Finetuned-Gemma3-27B: HF/neo4j, Link

- **Academic publications**
  - Text2Cypher: Bridging Natural Language and Graph Databases, GENAIK workshop, COLING, 2025, Link
  - Enhancing Text2Cypher with Schema Filtering, LLM-TEXT2KG workshop, ESWC, 2025, Link
  - Text2Cypher: Data Pruning using Hard Example Selection, LLM-DPM workshop, SIGMOD/PODS, 2025, Link
  - Text2Cypher Across Languages: Evaluating and Finetuning LLMs, NLPIR, 2025, Link

- **Additonal Blogposts**
  - Neo4j Text2Cypher: Analyzing Model Struggles and Dataset Improvements, Medium, 2025, Link
  - Text2Cypher Across Languages: Evaluating Foundational Models Beyond English, Medium, 2025, Link
  - Exploring Iterative Refinement for Text2Cypher, Medium, 2025, Link

Accessing Data on Graph - Text2Cypher

Datasets



References

1) Text2Cypher-2024v1, HF/Neo4j, https://huggingface.co/datasets/neo4j/text2cypher-2024v1
2) Text2Cypher-2025v1, HF/Neo4j, https://huggingface.co/datasets/neo4j/text2cypher-2025v1

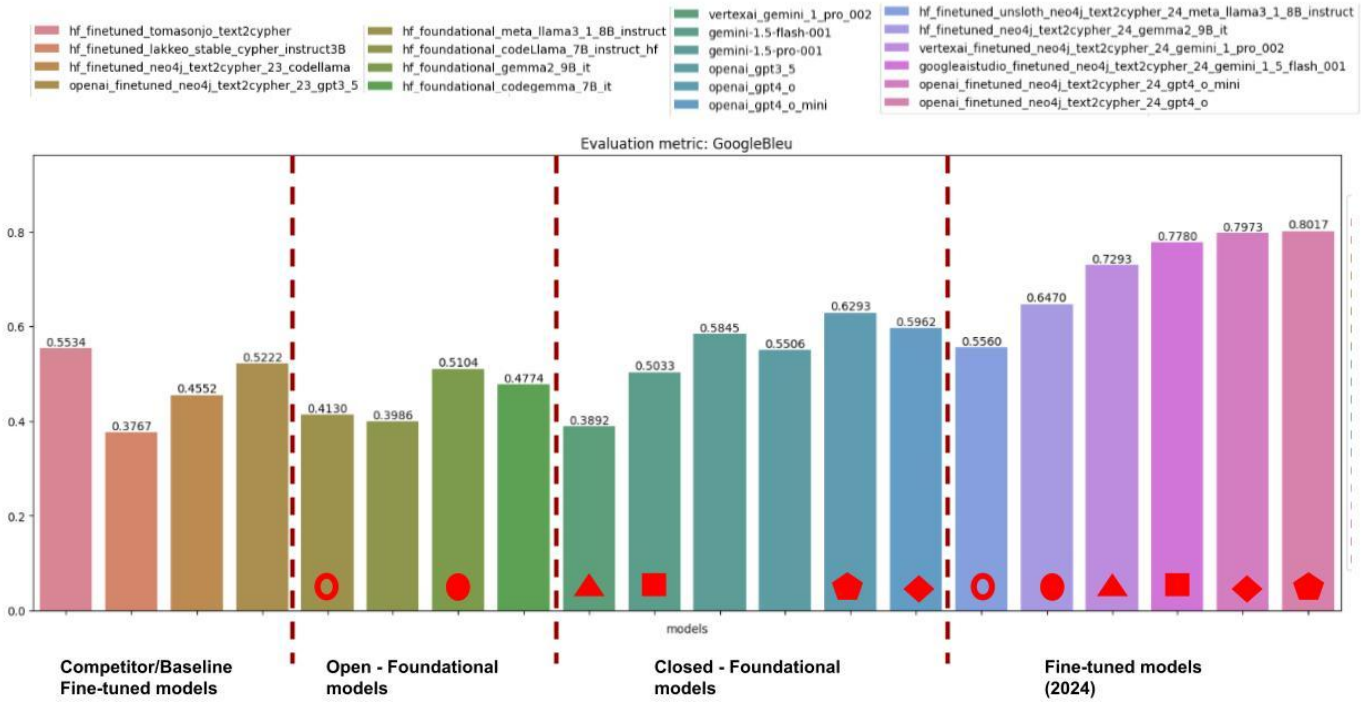
Accessing Data on Graph - Text2Cypher

Benchmarking Foundational and FinetunedModels

Lexical comparison (Translation based)

- Ground-truth
- Generated

Note: Results for 'execution output' comparison are also available



References

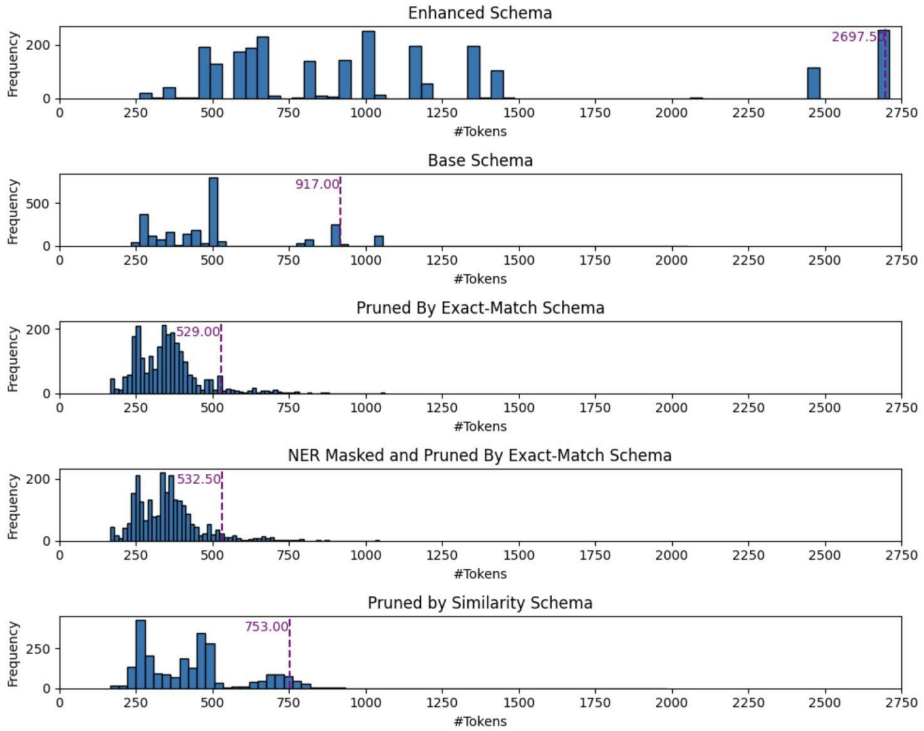
1) Text2Cypher Models, HF/Neo4j, https://huggingface.co/neo4j/models
2) Text2Cypher: Bridging Natural Language and Graph Databases, GENAIK workshop, COLING, 2025, Link

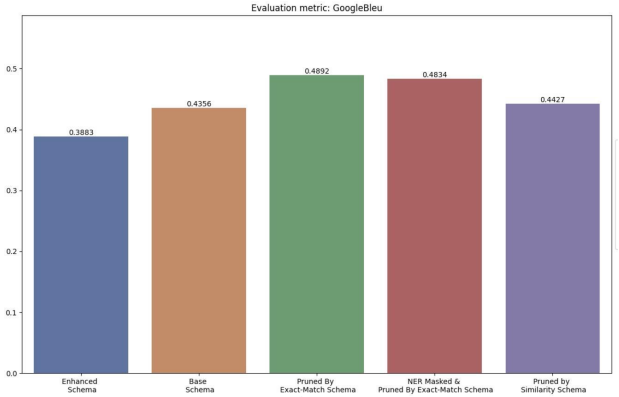
Accessing Data on Graph - Text2Cypher

Schema Filtering

Reduced number of **tokens** used in inference time using **heuristics.**

Performance remained **same or increased,** i.e., noise reduction





References

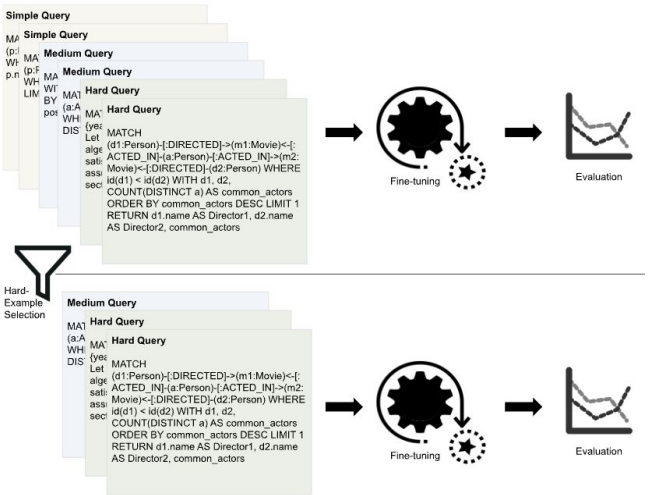
1) Enhancing Text2Cypher with Schema Filtering, LLM-TEXT2KG workshop, ESWC, 2025, Link

Accessing Data on Graph - Text2Cypher

Data Pruning using Hard Example Selection

Reduced number of **instances** used in training time using **heuristics.**

Performance remained **close** to baseline.



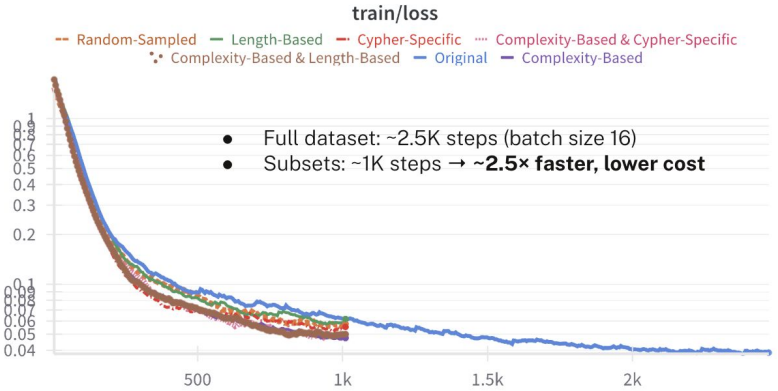


Table 1: Performance Comparison: Original (2.5K steps) vs. Randomly-Sampled (1K steps) vs. Hard-Example Selection (best scores - 1K steps)

	Translation-Based		Execution-Based	
	Google-Bleu	Exact-Match	Google-Bleu	Exact-Match
Original	0.7585	0.3642	0.2534	0.2740
Randomly-Sampled	0.6971	0.2048	0.2121	0.2550
Hard Example Selection (best)	0.7140	0.2599	0.2473	0.2639

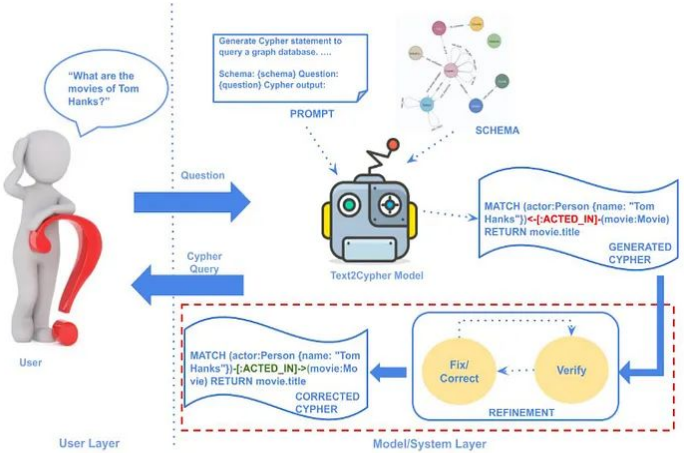
References
1) Text2Cypher: Data Pruning using Hard Example Selection, LLM-DPM workshop, SIGMOD/PODS, 2025, Link

Accessing Data on Graph - Text2Cypher

Refinement Loops

Verification
- Rule-based
- CyVer-based
- Execution-based
- LLM-based

Correction
- Rule-based
- LLM-based

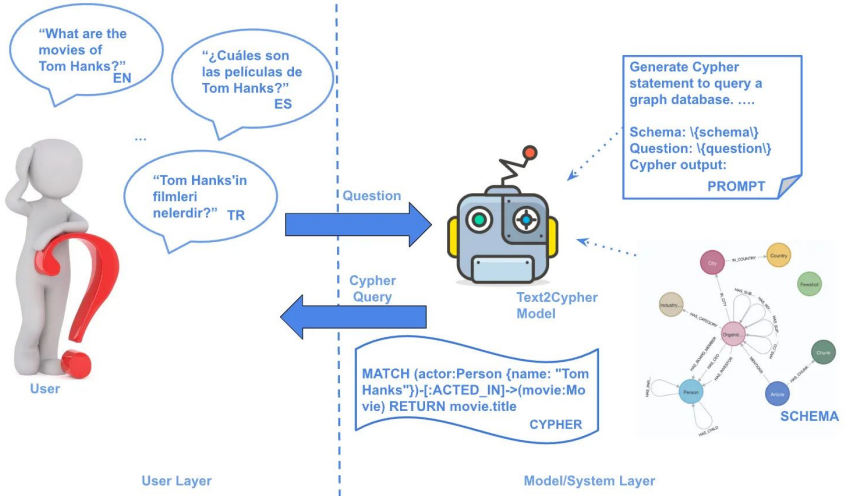


Verifiers: [RB, CYVER], Correctors: [RB, LLM], Metadata: Yes

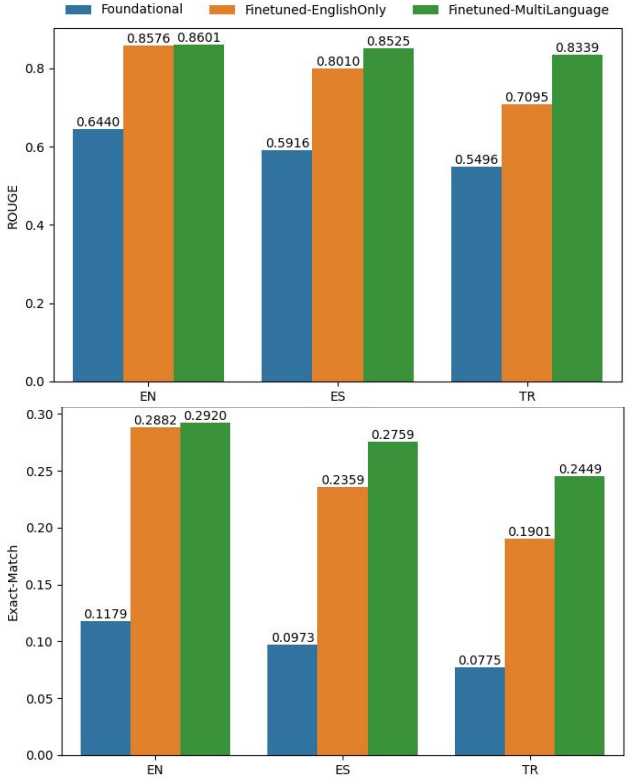
	Cypher query & Execution output
Generation	**Predicted Cypher:** "MATCH (u:User)-[r:RATED]->(m:Movie)-[:IN_GENRE]->(g:Genre)\nWITH u, g, COUNT(m) AS movies_in_genre\nORDER BY movies_in_genre DESC\nWITH u, COLLECT(g.name) AS genres\nWHERE SIZE(genres) = (SELECT COUNT(*) FROM (MATCH (g:Genre) RETURN g))\nRETURN u.name AS User, COUNT(*) AS Total_Movies_Rated\nORDER BY Total_Movies_Rated DESC\nLIMIT 3"
	Execution output - *Manually checked* Neo.ClientError.Statement.**SyntaxError** Invalid input 'COUNT': expected an expression (line 1, column 192 (offset: 191)) ....
Iteration 1	**Corrected Cypher:** "MATCH (u:User)-[r:RATED]->(m:Movie)-[:IN_GENRE]->(g:Genre)\nWITH u, COLLECT(DISTINCT g.name) AS genres\nWITH u, genres, SIZE(genres) AS genre_count\nMATCH (g:Genre)\nWITH u, genre_count, COUNT(g) AS total_genres\nWHERE genre_count = total_genres\nRETURN u.name AS User, COUNT(DISTINCT m) AS Total_Movies_Rated\nORDER BY Total_Movies_Rated DESC\nLIMIT 3",
	Execution output - *Manually checked* Neo.ClientError.Statement.**SyntaxError** Variable `m` not defined (line 1, column 280 (offset: 279)) ...
Iteration 2	**Corrected Cypher:** "MATCH (u:User)-[r:RATED]->(m:Movie)-[:IN_GENRE]->(g:Genre)\nWITH u, COLLECT(DISTINCT g.name) AS genres, COUNT(DISTINCT m) AS total_movies_rated\nWITH u, genres, total_movies_rated, SIZE(genres) AS genre_count\nMATCH (g:Genre)\nWITH u, genre_count, COUNT(DISTINCT g) AS total_genres, total_movies_rated\nWHERE genre_count = total_genres\nRETURN u.name AS User, total_movies_rated AS Total_Movies_Rated\nORDER BY Total_Movies_Rated DESC\nLIMIT 3",
	Execution_output - *Manually checked* User , Total_Movies_Rated → ("Darlene Garcia", 2391), ("Karen Avila", 1735), ("Angela Robertson", 1610)
Iteration 3	**N/A** - *Verifiers identified as valid*

References
1) Exploring Iterative Refinement for Text2Cypher, Medium, 2025, Link
2) https://neo4j.com/blog/developer/verify-neo4j-cypher-queries-with-cyver/

Text2Cypher Across Languages



Foundational vs finetuned



Multilingual dataset

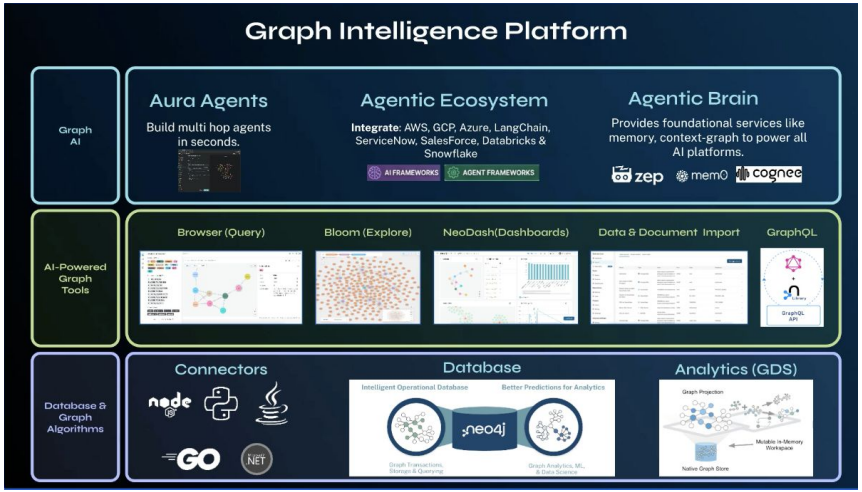
- Test set: HF/[mgoNeo4j/translated_text2cypher24_testset]
- Training set: HF/[mgoNeo4j/translated_text2cypher24_trainset_sampled]

References

1) Text2Cypher Across Languages: Evaluating Foundational Models Beyond English Link
2) Ozsoy, Makbule Gulcin, and William Tai. "Text2Cypher Across Languages: Evaluating and Finetuning LLMs." *NLPIR* (2025).

Accessing Data on Graph

- **Neo4j as Graph Intelligence Platform**
  - More than Text2Cypher
    - GraphRAG
    - RAG Agents
    - Graph Data Science (GDS)
    - GDS Agents
    - Knowledge Graph Extraction
    - ...



Outline

- **Recommender Systems**
  - Foundations and Advancements
  - Recommenders in Production
- **Recommender Systems and Graphs**
  - Recommenders As Graph Problems
  - Graph-powered Recommendations
- **Accessing Data on Graph**
  - Text2Cypher: Natural language to Cypher query language
  - Challenges and opportunities
- **Emerging LLM & Graph Applications**
  - Graph construction from unstructured data
  - Retrieval-Augmented Generation (RAG) on graphs
  - Other research directions (graph reasoning, dynamic graphs)
- **Conclusion and Discussion**

Emerging LLM & Graph Applications

Large Language Models (LLMs)

Pros

- Flexible across many NLP tasks
- Strong contextual understanding
- Zero/few-shot learning
- Scalable

Cons

- Hallucinations, weak factual grounding
- High compute & cost
- Limited interpretability
- Weak multi-step reasoning
- Bias risk

Knowledge Graphs (KGs)

Pros

- Explicit structured knowledge
- Strong multi-hop reasoning & querying
- High accuracy in domains
- Explainable & reusable

Cons

- Costly to build & maintain
- Scalability challenges
- Hard with unstructured data
- Limited coverage

References

1) Cai, Linyue, et al. "Practices, opportunities and challenges in the fusion of knowledge graphs and large language models." *Frontiers in computer science* 7 (2025): 1590632.

Emerging LLM & Graph Applications

- **LLM** and **KGs complement** each other
  - Increase knowledge coverage
  - Reduce construction cost
  - Reduce hallucinations
  - Improve reasoning
  - Objective factual verification
  - …

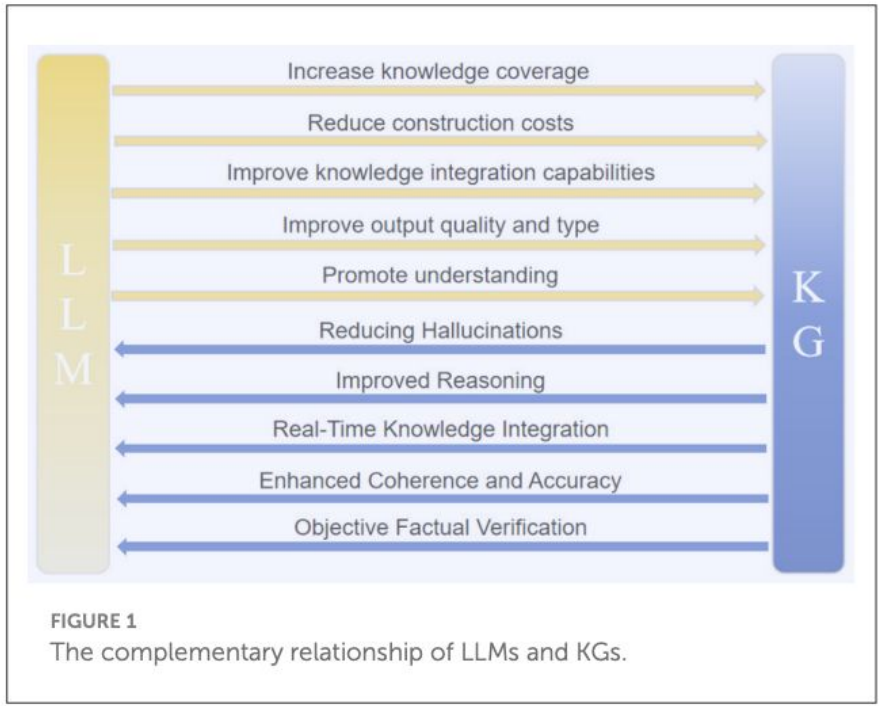


FIGURE 1
The complementary relationship of LLMs and KGs.

References

1) Cai, Linyue, et al. "Practices, opportunities and challenges in the fusion of knowledge graphs and large language models." *Frontiers in computer science* 7 (2025): 1590632.

Emerging LLM & Graph Applications

- LLMs for Knowledge Graph (KG)
  - KG Construction
  - KG Reasoning
  - KG Completion
  - KG Embedding
  - KG Validations
  - …

- Knowledge Graph (KG) for LLMs
  - Reasoning
  - Fact Inference
  - Semantic Analysis and Validation
  - KG Data as Input
  - …

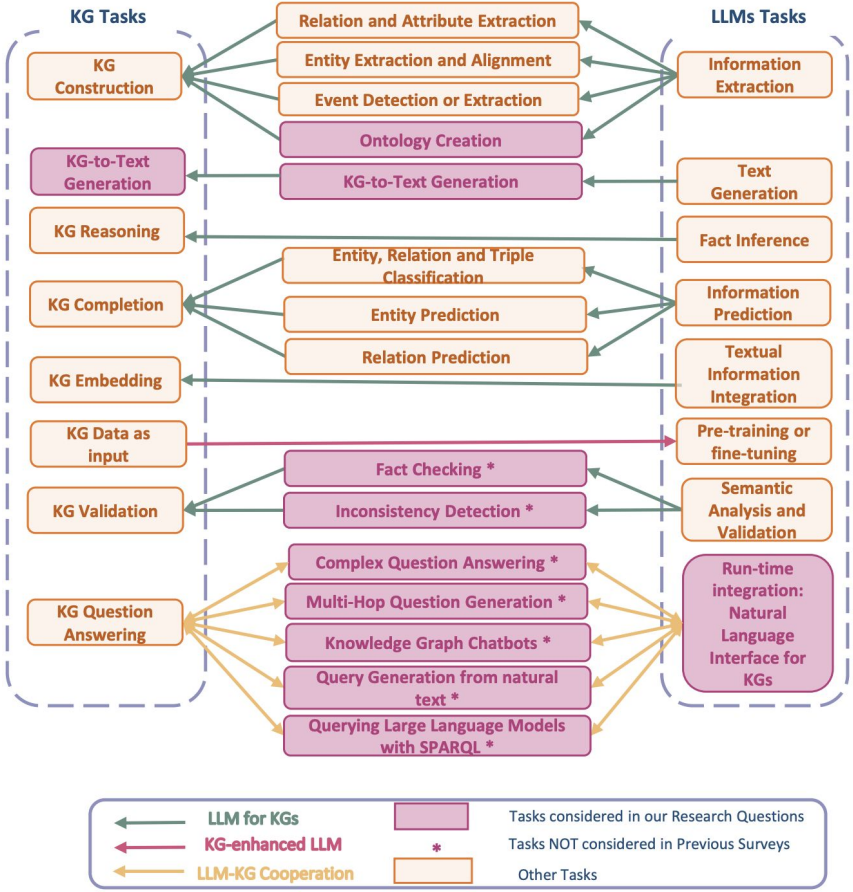


Figure 1: Categorization of the interplay between LLMs and KGs

References

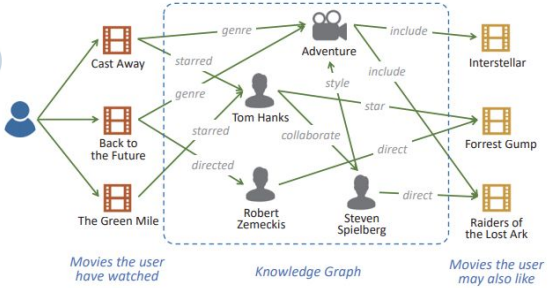
1) Khorashadizadeh, Hanieh, et al. "Research trends for the interplay between large language models and knowledge graphs." arXiv preprint arXiv:2406.08223 (2024).

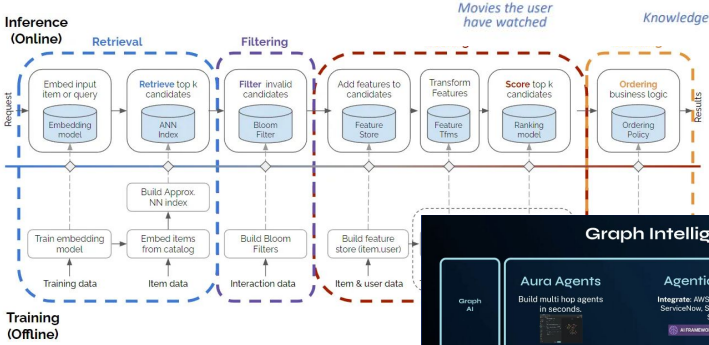
Outline

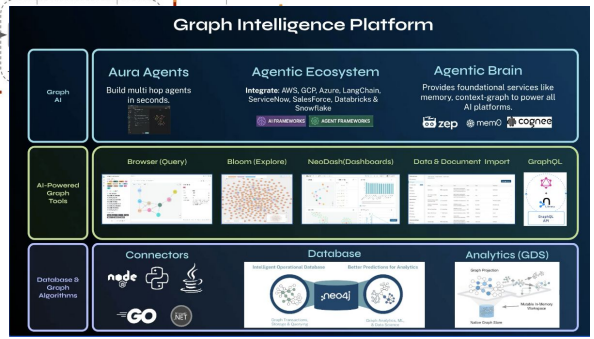
- **Recommender Systems**
  - Foundations and Advancements
  - Recommenders in Production
- **Recommender Systems and Graphs**
  - Recommenders As Graph Problems
  - Graph-powered Recommendations
- **Accessing Data on Graph**
  - Text2Cypher: Natural language to Cypher query language
  - Challenges and opportunities
- **Emerging LLM & Graph Applications**
  - Graph construction from unstructured data
  - Retrieval-Augmented Generation (RAG) on graphs
  - Other research directions (graph reasoning, dynamic graphs)
- **Conclusion and Discussion**

Conclusion and Discussion

- **Recommender Systems**
  - Summary of techniques
  - Not only model, but a system
- **Recommender Systems and Graphs**
  - Graph-based recommenders
  - Knowledge graphs beyond recommendations
- **Graph Intelligence**
  - Neo4j Graph Intelligence Platform
  - Accessing Data: Text2Cypher
- **Emerging LLM & Graph Applications**
  - LLM and KGs complement each other







Conclusion and Discussion

Key Takeaways

- **Recommender** systems
  - **Not just** ML models
  - **End-to-end** systems
- **Challenges**
  - Algorithmic/Model-level
  - System-level
  - Business priorities
- **Evalution**
  - Not only: Accuracy
  - Also: Exposure, temporal, business

- **Graph intelligence platforms**
  - **Graphs** → **Complex** relationships in real data
  - Make knowledge graphs practical **at scale**
- **LLMs** are **transforming**
  - How we build recommender systems
  - How we access graph-based systems

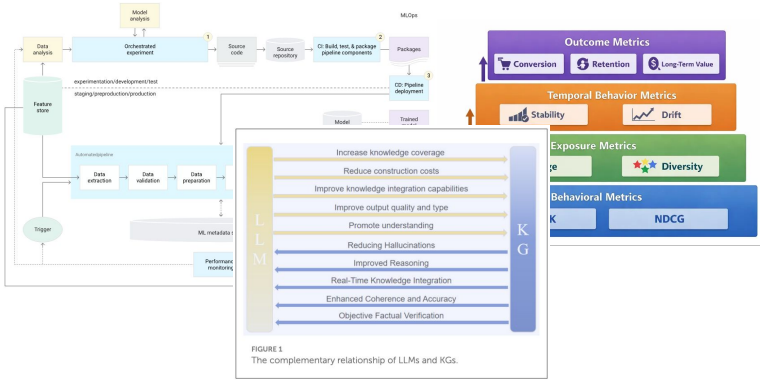


FIGURE 1
The complementary relationship of LLMs and KGs.

References

1) https://docs.cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning

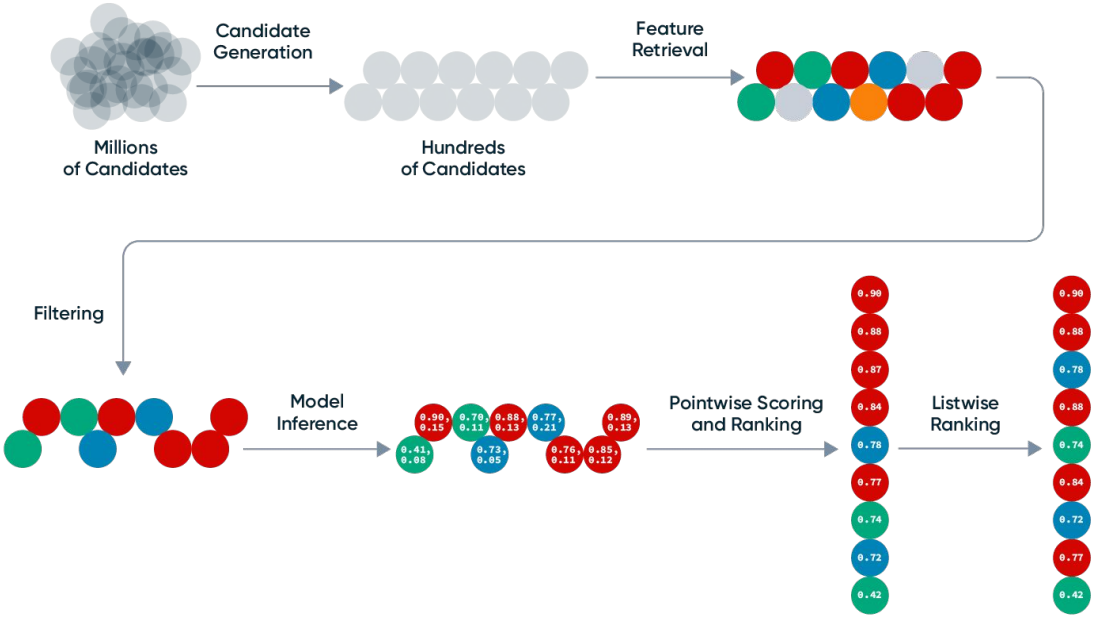
Thank you

Questions?

Additional Slides

Recommender Systems

- **Not just a model → But a system**
  - Example from DataBricks (2025)



References

1) https://medium.com/nvidia-merlin/recommender-systems-not-just-recommender-models-485c161c755e
2) https://www.databricks.com/blog/guide-to-building-online-recommendation-system

Recommender Systems

- **Common Challenges of Recommender Systems (System/Production Level):**
  - **Lack of Result Reproducibility**
    - **Same** code, data, and parameters → **Different** results
    - Randomness in
      - model initialization,
      - differences in environments
      - minor changes in code etc.
    - Solution:
      - **Repeatable pipeline**
      - **Consistent environment** for training and serving
  - **Offline-Online Performance Mismatch**
    - Good offline evaluation metrics → Not observed in online environment
    - E.g. **Optimize** for **click** → **Cheap** items
    - Solution:
      - Recheck **offline metrics**, align with **business objectives**
      - E.g. Click-Through Rate (CTR) vs dwell time or repeat visits or session length

References
1) Challenges of Building a Recommender System
2) Netflix's Raising a Recommender System

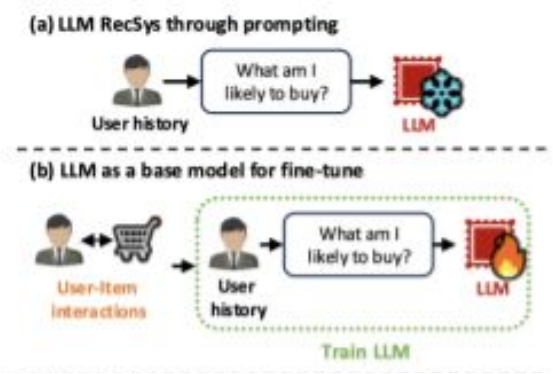
Recommender Systems

- **Common Challenges of Recommender Systems (System/Production Level):**
  - **Oscillating Outputs**
    - **Quality** of recommendation changes over **time**
    - E.g., **Distribution shifts** or **sampling bias**
    - Solution:
      - **Check** biases or fluctuations in the training **data**
      - Check training **delays**, i.e., data collection vs. training time
  - **Adding new features or data drops performance**
    - **New data/feature** included to improve performance (in offline) → **Performance drop** (in online)
    - **Mismatch** between model objective and data/feature
    - Solution:
      - Check objective function, labels, data, features, model arhitecture

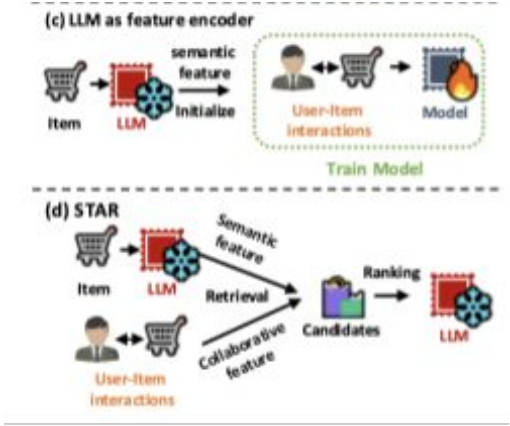
References
1) Challenges of Building a Recommender System
2) Netflix's Raising a Recommender System

Recommender Systems

- **LLM-based methods**
  - Prompting
  - Fine-tuning



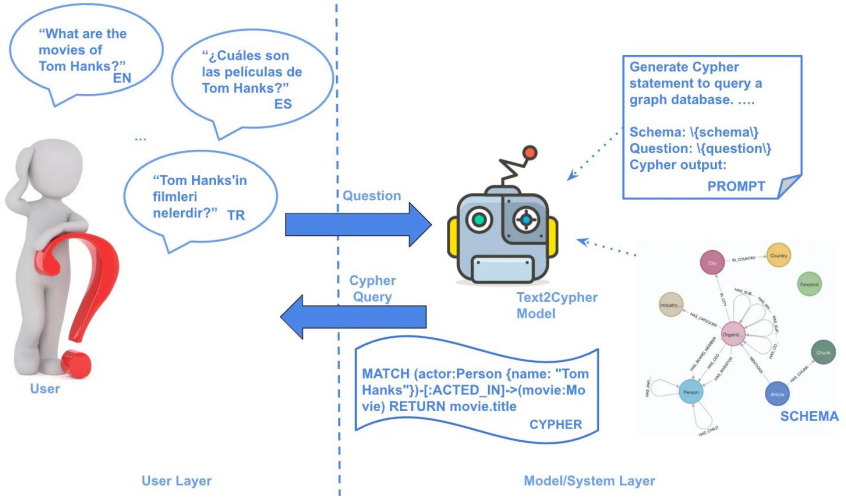
  - Feature encoder
  - Two stage: Retrieval + Ranking



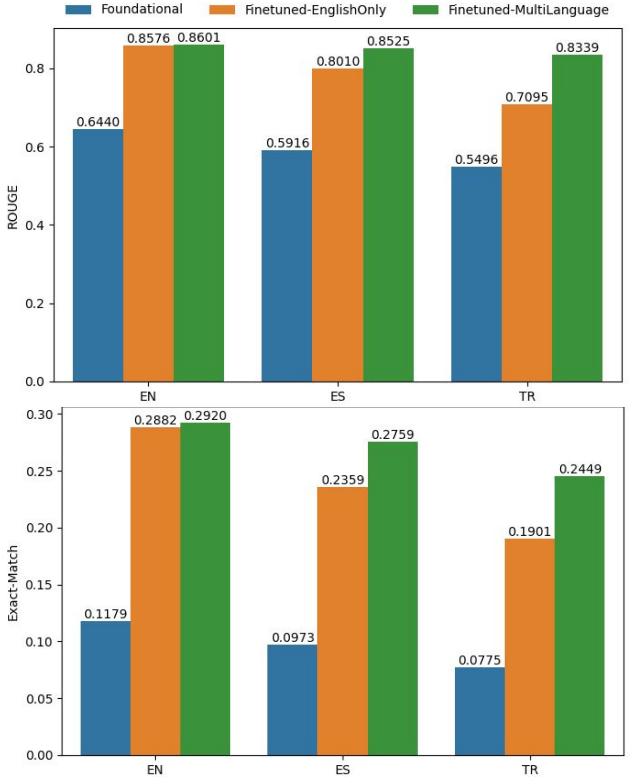
References

1) https://eugeneyan.com/writing/recsys-llm/
2) https://lfaidata.foundation/communityblog/2025/08/25/leverage-llm-for-next-gen-recommender-systems-technical-deep-dive-into-llm-enhanced-recommender-architectures/
3) https://amatria.in/blog/recsyskeynote
4) Lee, Dong-Ho, et al. "Star: A simple training-free approach for recommendations using large language models." arXiv preprint arXiv:2410.16458 (2024).

Text2Cypher Across Languages



Foundational vs finetuned



Multilingual dataset

- Test set: HF/[mgoNeo4j/translated_text2cypher24_testset]
- Training set: HF/[mgoNeo4j/translated_text2cypher24_trainset_sampled]

References

1) Text2Cypher Across Languages: Evaluating Foundational Models Beyond English Link
2) Ozsoy, Makbule Gulcin, and William Tai. "Text2Cypher Across Languages: Evaluating and Finetuning LLMs." *NLPIR* (2025).

Conclusion and Discussion

(Near) Future Skills That Matter

- Designing **production ML systems** (not just training models)
- **Combining LLMs** with
  - Graph data, Knowledge graphs, Un/Structured data
  - E.g., RAG, Text2Cypher, reasoning
- **Evaluation**
  - Not only for Recommender Systems or Knowledge Graphs
  - Topics:
    - Accuracy + Other dimensions
      - Presentation, temporal, business
    - Structured vs. unstructured outputs
    - Agents

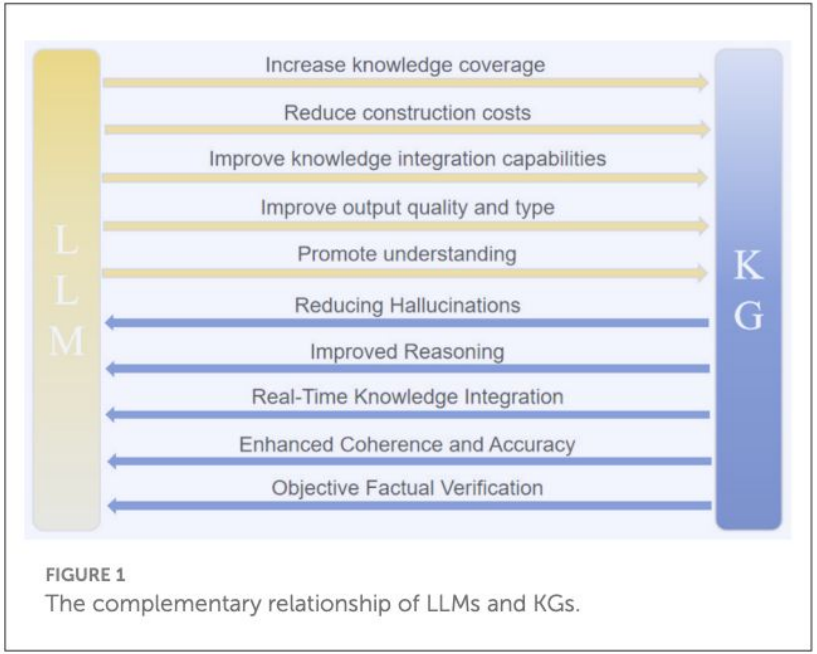


FIGURE 1
The complementary relationship of LLMs and KGs.