AutoG: Towards automatic graph construction from tabular data

Zhikai Chen[1*] ,Han Xie[2] ,Jian Zhang[2], Xiang Song[2], Jiliang Tang[1],Huzefa Rangwala[2], George Karypis[2]

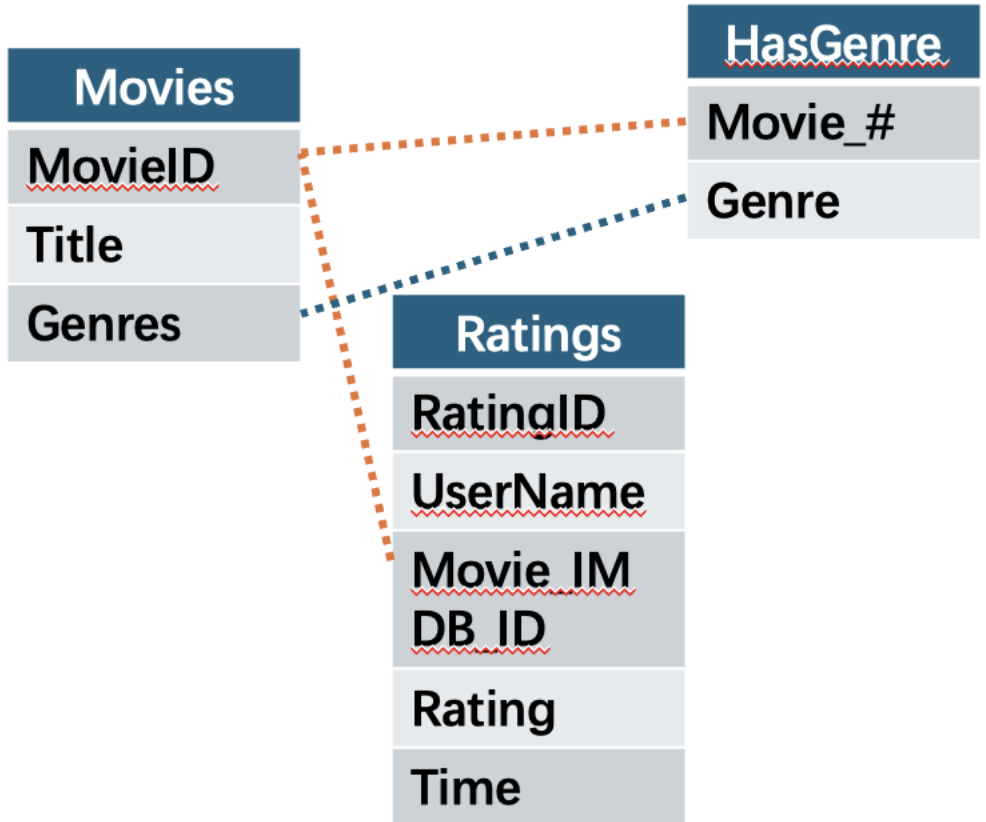
[1]Michigan State University [2]Amazon

*This work was done when Zhikai was an intern at Amazon

amazon | science

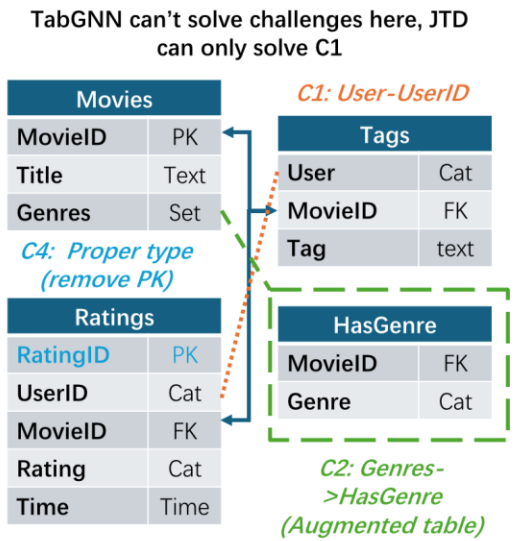
Background

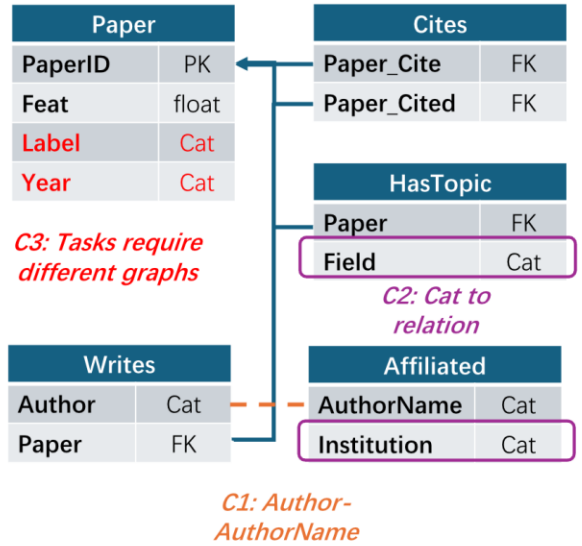
- Graph machine learning has achieved a wide range of attention. However, most focus has been paid to the development of the **model side.** From the data side, a fundamental question is how to construct a graph, for example, from popular storage types like tabular data.
- A graph consists of nodes and edges, meaning we need to automatically extract proper node types and edges from tabular data. This is a non-trivial task since tables extracted from data lakes usually present missing relations and unknown entity types.


Movies
MovieID
Title
Genres
HasGenre
Movie_#
Genre
Ratings
RatingID
UserName
Movie_IM
DB_ID
Rating
Time

Formalizing problems

- Before discussing how to solve graph construction problems from multi-tabular data, we need first to clarify the definition of this problem.





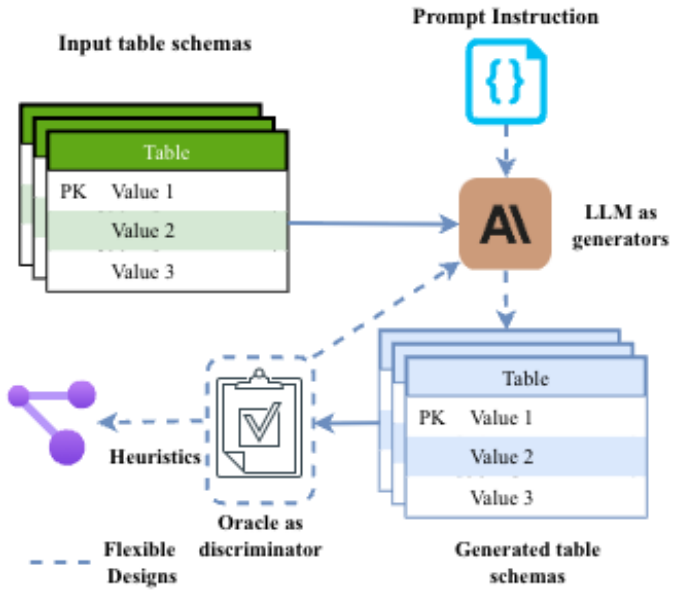
- We identify four key problems in constructing a downstream task-friendly graph: (C1) Identifying missing cross-table PK-FK/FK-FK relationships (C2) identifying self-induced relationships (C3) Transforming tables into proper node or edge types (C4) Generating proper graphs for different downstream tasks(*) adaptively.
- *This makes graph construction a complicated decision-making task going beyond rule-based methods!*

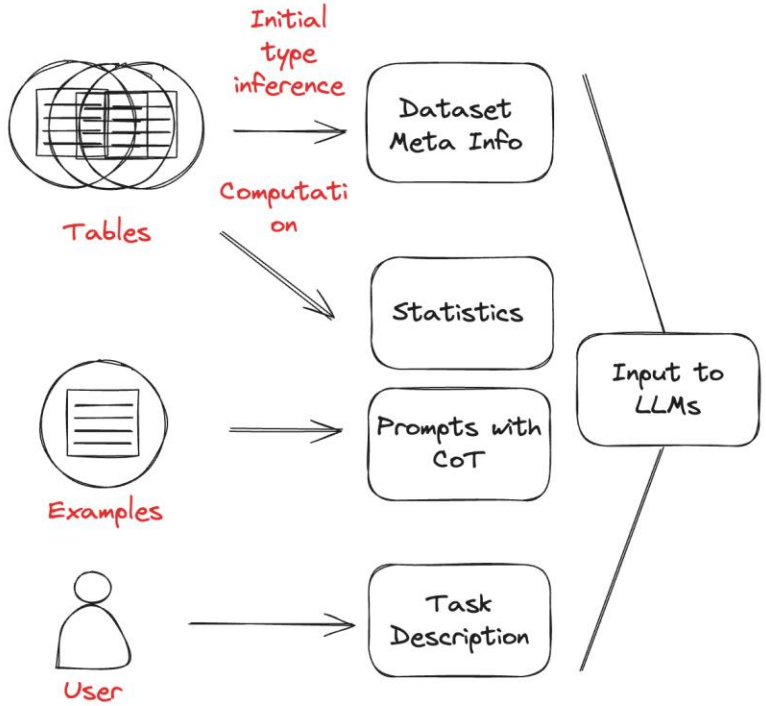
A benchmark for graph construction

- Despite the importance of graph construction, as far as we know, there are no existing multi-tabular datasets for studying **constructing a graph from tabular data**. To address this gap, we first adopt a set of existing multi-tabular datasets from [2,3,4] and pre-process them to fit our objective.
- **Preprocessing strategy.** To generate a test bench for automatic graph construction, we (1) first remove explicit FK and PK relations from the original table; (2) remove augmented dummy tables designed by experts [2]; (3) undo data normalization conducted by the original datasets. This results in non-trivial test cases for automatic graph construction.

Name of the dataset	#Tasks	#Tables	Inductive	C1	C2	C3	C4	Task type
Movielens	1	3	✓	✓	✓	✓	✗	Relation Attribute
MAG	3	5	✗	✓	✓	✓	✓	Entity Attribute, FK Prediction
AVS	2	3	✓	✓	✓	✓	✓	Entity Attribute
IEEE-CIS	1	2	✗	✗	✓	✓	✗	Entity Attribute
Outbrain	1	8	✓	✓	✓	✓	✗	Relation Attribute
Dignetica	2	8	✓	✓	✓	✓	✓	Relation Attribute, FK Prediction
RetailRocket	1	5	✓	✓	✓	✓	✗	Relation Attribute
Stackexchange	3	7	✓	✓	✓	✓	✓	Entity Attribute

Automatic graph construction using LLMs



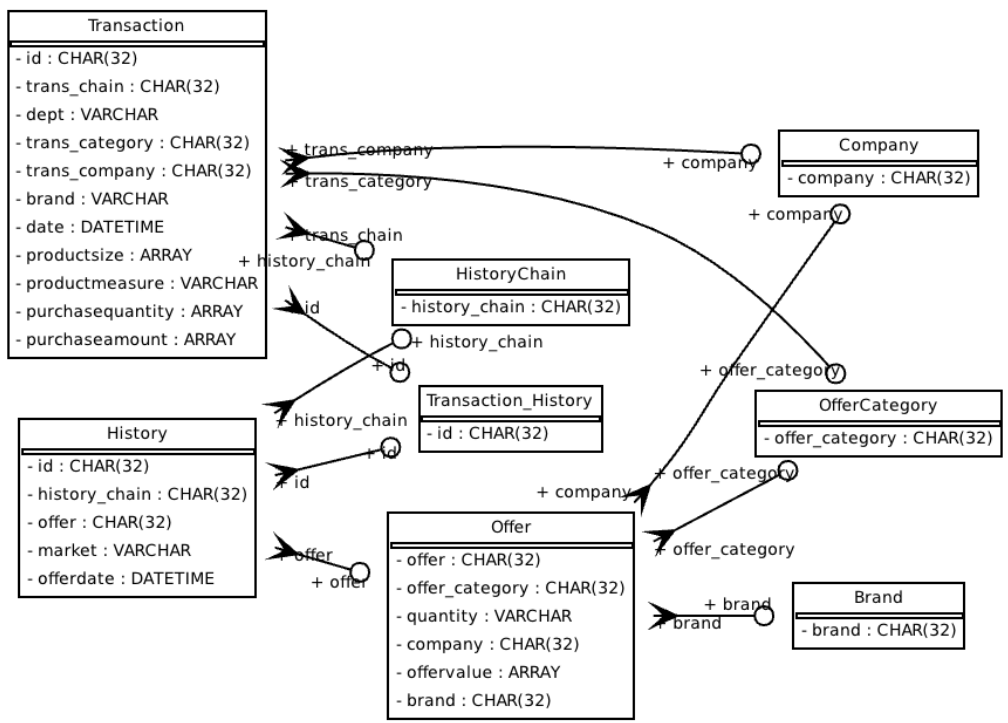


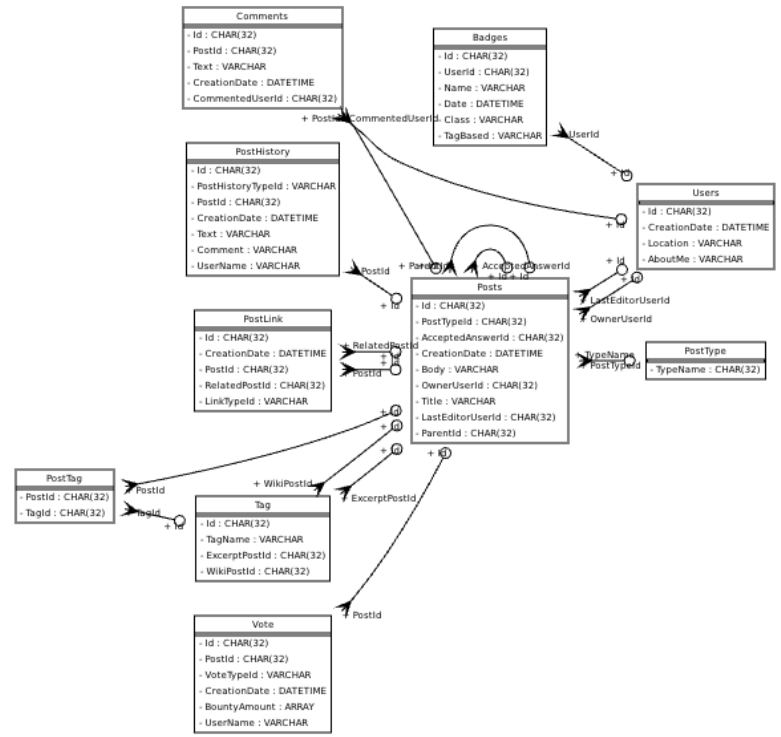
- Based on the design space of our benchmark, we find that graph construction is a complicated decision-making task that requires designers to make choices adaptively. Inspired by LLM agents' capability in decision-making in software engineering and data analysis, we present the very first step to evaluate LLM's potential in conducting graph construction as a data scientist.
- The whole framework consists of the following steps:
- 1**. Input Context Module** Wrap all relevant task and dataset information in the prompts.
- 2. **LLM as generators** After receiving the relevant context information from the input prompts, LLMs conduct **close-ended generation** to generate augmentation actions and change the schema.
- 3. **Oracle as discriminators** Finally, we adopt a GML model as a discriminator to determine the best schema based on downstream task performance.

- **LLM as generators** LLMs will follow Chain of thought prompts to select one of the following augmentations

1. CONNECT_TWO_COLUMNS
2. GENERATE_NEW_TABLE
3. REMOVE(ADD)_PRIMARY_KEY
4. UNFOLD_MULTI_CATEGORY_COLUMNS

Dataset	Task	XGBOOST	DeepFM	TabGNN	Vanilla Schema		JTD Schema		AutoG	Expert
		N/A	N/A	TabGNN	R2N	R2NE	R2N	R2NE	AutoG	Expert
Datasets with a Single Downstream Task										
IEEE-CIS	Fraud (AUC)	90.14	**90.28**	75.38	89.17	89.17	89.17	89.17	89.17	87.28
RetailRocket	CVR (AUC)	50.35	49.33	82.84	50.45	49.90	50.82	48.99	74.78	**84.70**
Movielens	Ratings (AUC)	53.62	50.93	55.34	57.34	61.20	54.55	57.23	69.10	**69.73**
Outbrain	CTR (AUC)	50.05	51.09	62.12	49.33	52.06	49.35	52.23	61.32	**62.71**
AVS	Repeat (AUC)	52.71	52.88	54.48	47.75	48.84	53.27	53.27	**56.03**	55.08
Datasets with Multiple Downstream Tasks										
MAG	Venue (Acc)	21.95	28.19	42.84	27.24	46.26	21.26	46.97	**49.88**	49.66
	Citation (MRR)	3.29	45.06	70.65	65.29	65.29	72.53	**81.50**	80.84	80.86
	Year (Acc)	28.09	28.42	52.77	**54.09**	30.90	53.07	53.07	**54.09**	35.35
Diginetica	CTR (AUC)	53.50	50.57	50.00	68.44	65.92	50.05	50.00	**75.07**	**75.07**
	Purchase (MRR)	3.16	5.02	5.01	5.64	7.70	11.37	15.47	**36.91**	**36.91**
Stackexchange	Churn (AUC)	58.20	59.84	78.27	77.67	76.47	85.58	84.85	**86.92**	85.58
	Upvote (AUC)	86.69	87.64	85.28	86.45	86.47	**88.61**	67.98	87.43	**88.61**
Ranking (RGCN)		5.9	5.2	4.7	4.2		3.1		2.0	2.0
Ranking (Average)		5.8	5.3	4.5	4.4		3.3		2.0	2.0


Transaction
- id : CHAR(32)
- trans_chain : CHAR(32)
- dept : VARCHAR
- trans_category : CHAR(32)
- trans_company : CHAR(32)
- brand : VARCHAR
- date : DATETIME
- productsize : ARRAY
- productmeasure : VARCHAR
- purchasequantity : ARRAY
- purchaseamount : ARRAY
+ trans_company
+ trans_category
+ company
Company
- company : CHAR(32)
+ company
+ trans_chain
+ history_chain
HistoryChain
- history_chain : CHAR(32)
+ id
+ history_chain
+ id
+ offer_category
Transaction_History
- id : CHAR(32)
+ history_chain
OfferCategory
- offer_category : CHAR(32)
History
- id : CHAR(32)
- history_chain : CHAR(32)
- offer : CHAR(32)
- market : VARCHAR
- offerdate : DATETIME
+ id
+ id
+ offer_category
+ company
Offer
- offer : CHAR(32)
- offer_category : CHAR(32)
- quantity : VARCHAR
- company : CHAR(32)
- offervalue : ARRAY
- brand : CHAR(32)
+ offer
+ offer
+ offer_category
+ brand
+ brand
Brand
- brand : CHAR(32)


Comments
- Id : CHAR(32)
- PostId : CHAR(32)
- Text : VARCHAR
- CreationDate : DATETIME
- CommentedUserId : CHAR(32)
Badges
- Id : CHAR(32)
- UserId : CHAR(32)
- Name : VARCHAR
- Date : DATETIME
- Class : VARCHAR
- TagBased : VARCHAR
PostHistory
- Id : CHAR(32)
- PostHistoryTypeId : VARCHAR
- PostId : CHAR(32)
- CreationDate : DATETIME
- Text : VARCHAR
- Comment : VARCHAR
- UserName : VARCHAR
Users
- Id : CHAR(32)
- CreationDate : DATETIME
- Location : VARCHAR
- AboutMe : VARCHAR
Posts
- Id : CHAR(32)
- PostTypeId : CHAR(32)
- AcceptedAnswerId : CHAR(32)
- CreationDate : DATETIME
- Body : VARCHAR
- OwnerUserId : CHAR(32)
- Title : VARCHAR
- LastEditorUserId : CHAR(32)
- ParentId : CHAR(32)
PostLink
- Id : CHAR(32)
- CreationDate : DATETIME
- PostId : CHAR(32)
- RelatedPostId : CHAR(32)
- LinkTypeId : VARCHAR
PostType
- TypeName : CHAR(32)
PostTag
- PostId : CHAR(32)
- TagId : CHAR(32)
Tag
- Id : CHAR(32)
- TagName : VARCHAR
- ExcerptPostId : CHAR(32)
- WikiPostId : CHAR(32)
Vote
- Id : CHAR(32)
- PostId : CHAR(32)
- VoteTypeId : VARCHAR
- CreationDate : DATETIME
- BountyAmount : ARRAY
- UserName : VARCHAR

- Despite the promising performance demonstrated on the benchmark, we must acknowledge the limitations of LLM-based automatic graph construction and even whole GML-based methods when tackling industrial tabular data. The first case represents those cases that involve bad heterophilous relationships. We try to address this using semantic information, while currently, the effective solution is to generate multiple candidates and select based on performance. The second problem is related to network effect,

- Case 1: An example is the year prediction task on the MAG dataset. Taking
- a deeper look at the generated graph statistics, it shows that when predicting the venue of “Paper”, the adjusted homophily of labels based on metapath “Paper-Author-Paper” is 0.156.
- Case 2: An example is the IEEE-CIS dataset. We find that despite the reasonable data normalization process (for example, generating a new table based on columns forming an independent entity “MatchStatus”, a full example can be shown in Appendix E.3), such graph construction negatively affects the performance of GML model compared to the original one