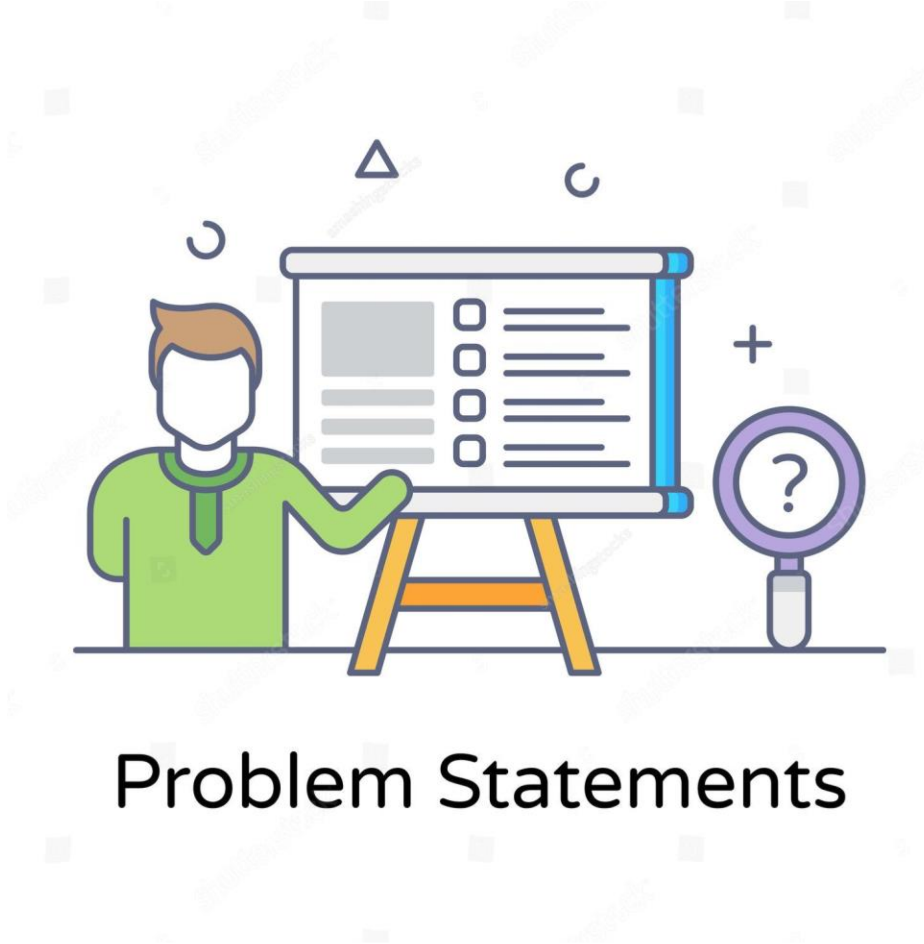
Generating answers from semantically structured documents

Pisa, 09/02/2024

Author:

Niko Dalla Noce

Introduction

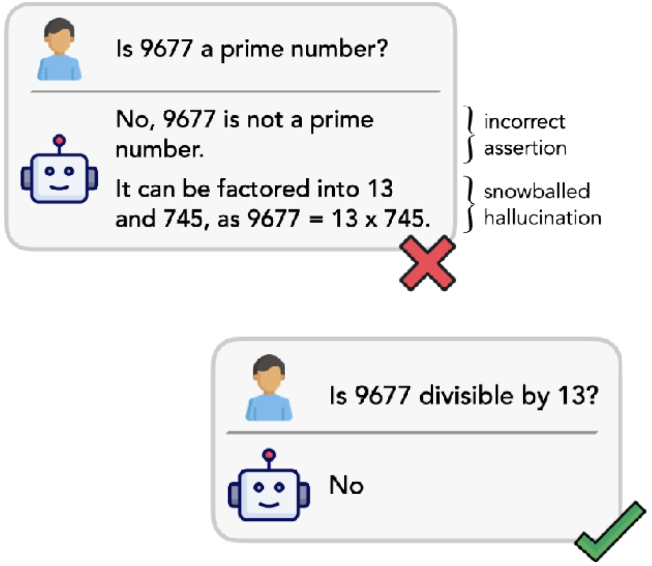


Introduction to the problem

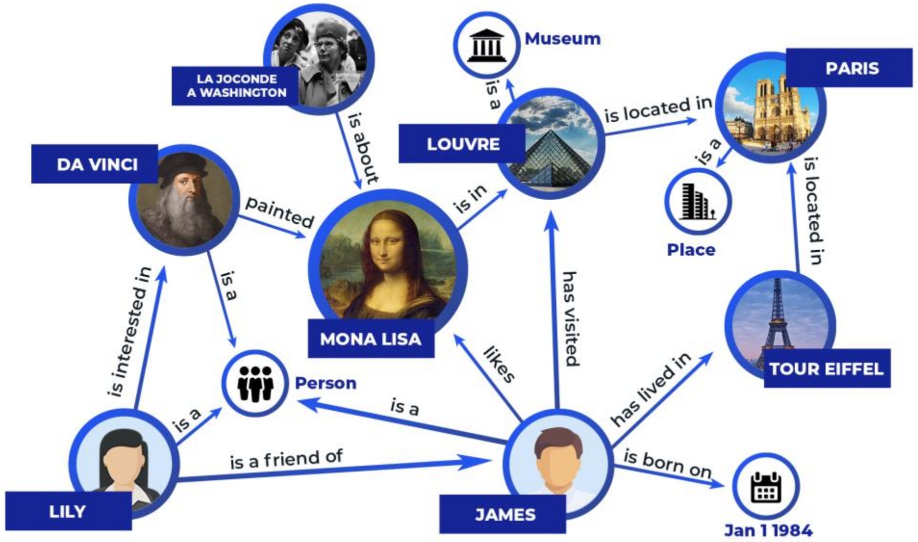
- We want to develop a system that can answer to the user's questions on a structured set of documents.

- Direct prompting to LLMs may suffer of hallucinations. In some specific cases, the faithful of an answer is crucial.

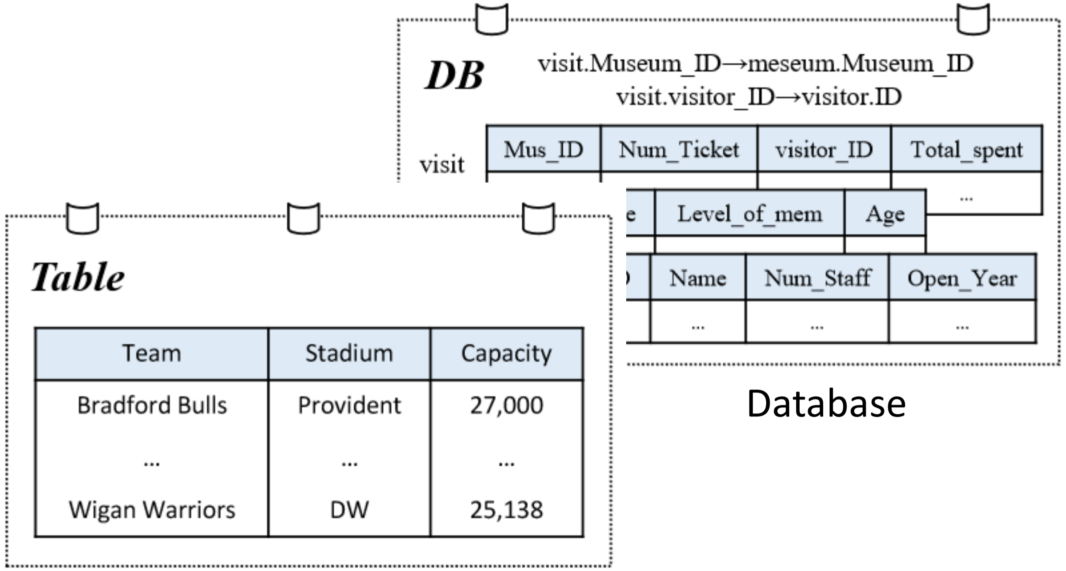
- LLMs are trained on large datasets that, for obvious reasons, cannot include private documents.



How data is semantically structured



Knowledge graph



Team	Stadium	Capacity
Bradford Bulls	Provident	27,000
...	...	...
Wigan Warriors	DW	25,138

Database

Table

$\boldsymbol{KG = (V, E)}$, where:

- $\boldsymbol{V}$ is the set of entities.
- $\boldsymbol{E \subseteq V \times R \times V}$.
- $\boldsymbol{R}$ is the set of relations.

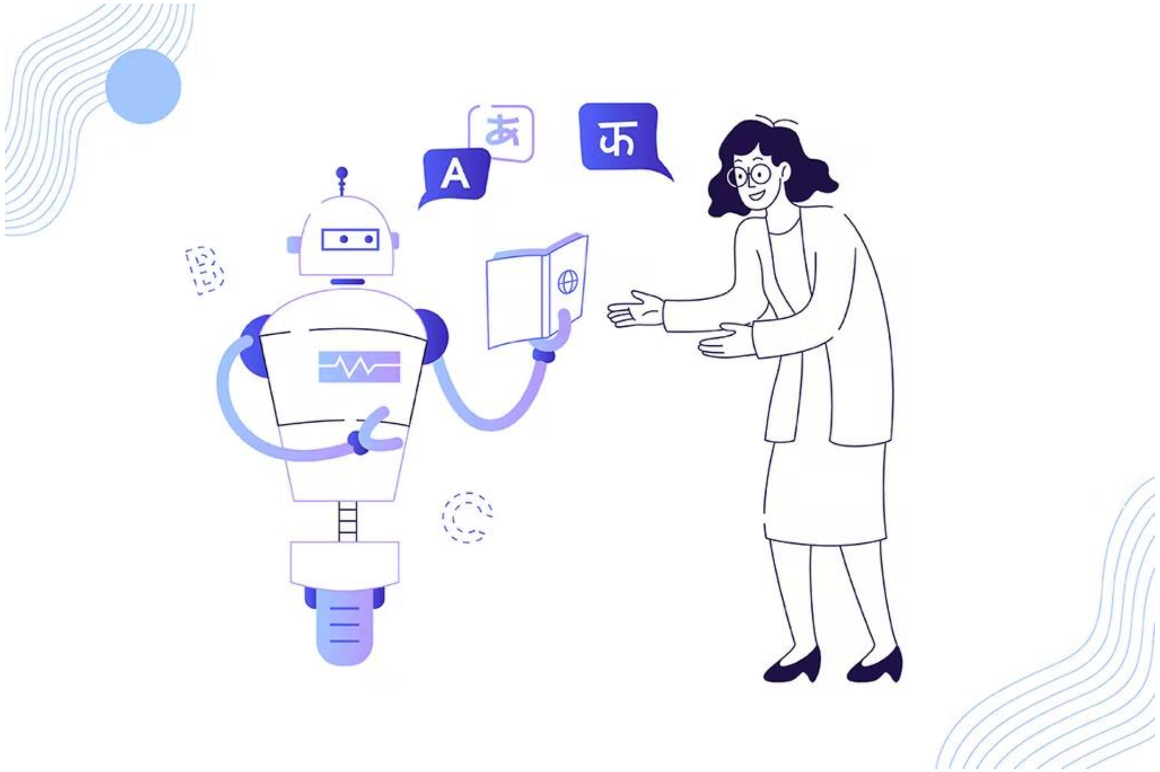
How data is semantically structured [cont.]

- An *ontology*, is a data model that represents a set of concepts within a domain and the relationships between those concepts.
- Main components of an ontology:
  - **Individuals:** things that can be named in the data.
  - **Classes:** a collection of individuals.
  - **Properties:** these form a connection between an individual and a value.
    E.g. Owner → is → Type of Owner (business or individual)
  - **Relationships:** define how two individuals are related to each other.
    E.g. Team → hasCaptain → Player

	Teams	Captains	Wins	Owner
0	CSK	MS Dhoni	3	['Chennai Super Kings Cricket Limited']
1	SRH	David Warner	2	['Sun TV Network']
2	MI	Rohit Sharma	5	['Reliance Industries']
3	RR	Steve Smith	1	['Amisha Hathiramani', 'Manoj Badale', 'Lachla...
4	KXIP	KL Rahul	0	['Mohit Burman', 'Ness Wadia', 'Preity Zinta',...

	Owners	Type
0	Chennai Super Kings Cricket Limited	Business
1	Sun TV Network	Business
2	Reliance Industries	Business
3	Amisha Hathiramani	Individual
4	Manoj Badale	Individual

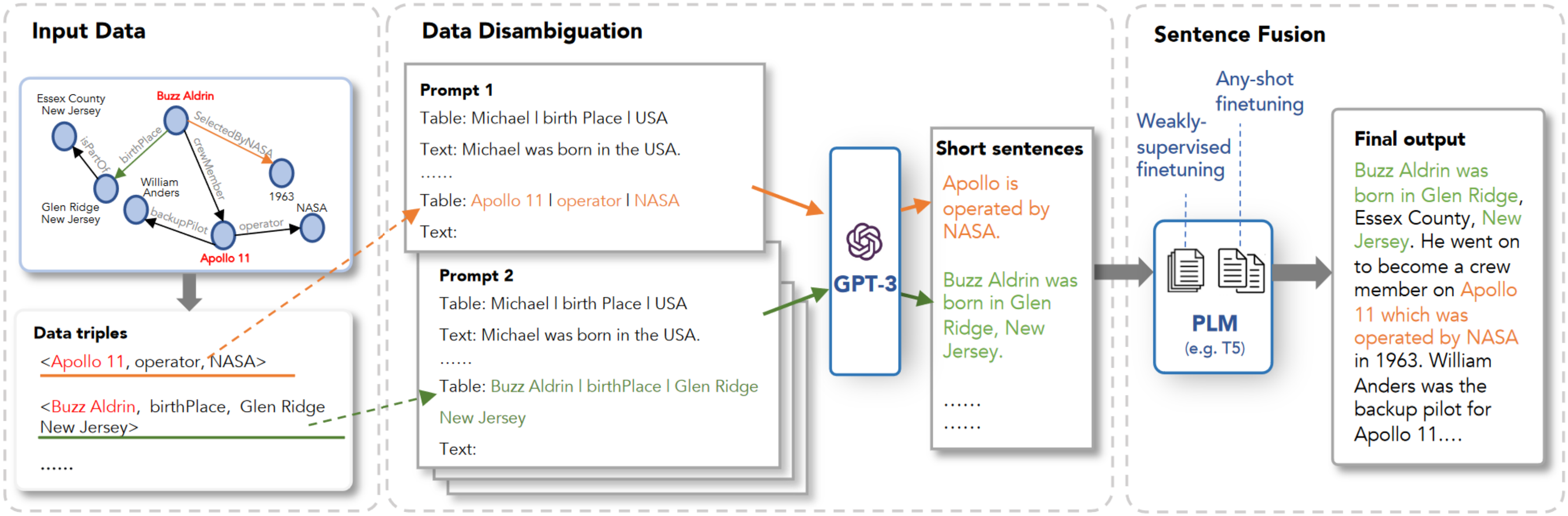
Text generation



Documents

- Pre-trained language models work well for summarizing or answering questions about set documents.
- Data is stored in a semantically structured way for creating a knowledge base (KB).
- How do we capture the semantic connections between the data? We take inspiration from knowledge graphs.

Any-Shot Data-to-Text Generation



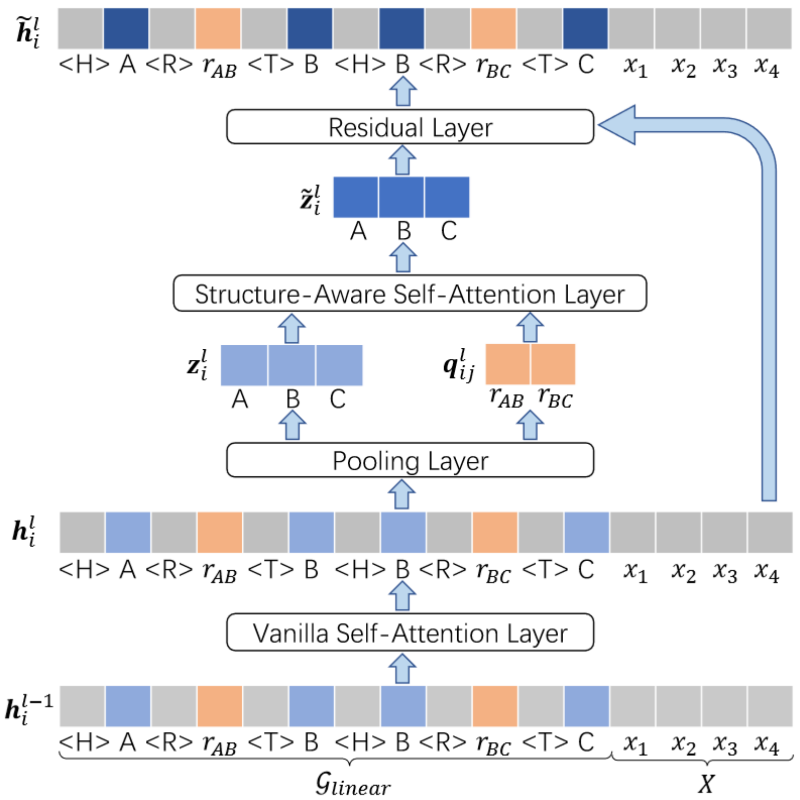
Graph-Text Joint Representation Learning

- Introduces a structure-aware semantic aggregation module on top of vanilla self-attention.
- Uses a mean pooling layer to obtain the representation of each entity and relation from the output of the vanilla self-attention layer.
- Structure-aware self-attention layer:

$$\tilde{z}_i^l = \sum_{j=1}^{|\mathcal{V}|} \beta_{ij}^l \left(\boldsymbol{z}_j^l \boldsymbol{W}^{VS} + \boldsymbol{q}_{ij}^l \boldsymbol{W}^{VR} \right)$$

$$\beta_{ij}^l = \frac{\exp(u_{ij}^l)}{\sum_{p=1}^{|\mathcal{V}|} \exp(u_{ip}^l)} \qquad u_{ij}^l = \frac{\left(\boldsymbol{z}_i^l \boldsymbol{W}^{QS}\right)\left(\boldsymbol{z}_j^l \boldsymbol{W}^{KS} + \boldsymbol{q}_{ij}^l \boldsymbol{W}^{KR}\right)^\top}{\sqrt{d_k}}$$

$$i = 1, 2, \cdots, |\mathcal{V}|$$

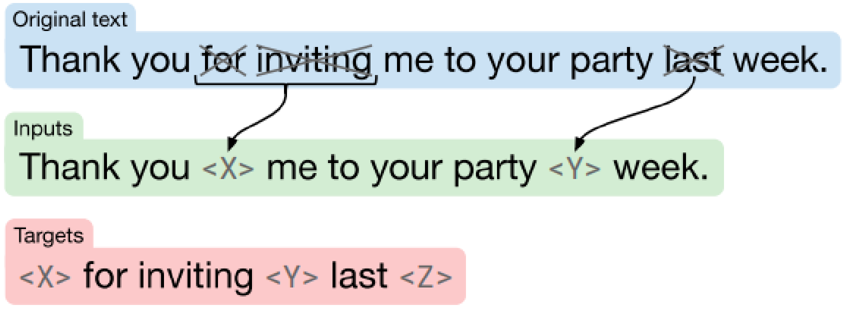


Pre-training on knowledge graphs

- Pre-training allows the model to acquire the representation of the words within the sentences.

- Also, it helps the models to converge faster on the fine-tuning task.

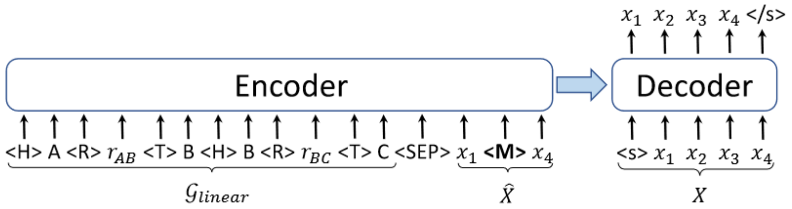
- We will see some approaches to apply the pre-training phase also on knowledge graphs.



T5's mask language modeling (Raffel et al., 2019)

Graph-Text Joint Representation Learning [cont.]

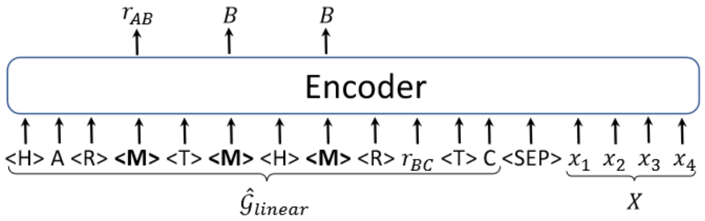
- **Two pre-training approaches that exploits the target text X:**
  - **Graph Enhanced Text Reconstruction:** recover the masked text sequence based on the complete knowledge graph.



(a) Graph Enhanced Text Reconstruction

$$\mathcal{L}_{text} = -\log P(X|\mathcal{G}, \hat{X})$$

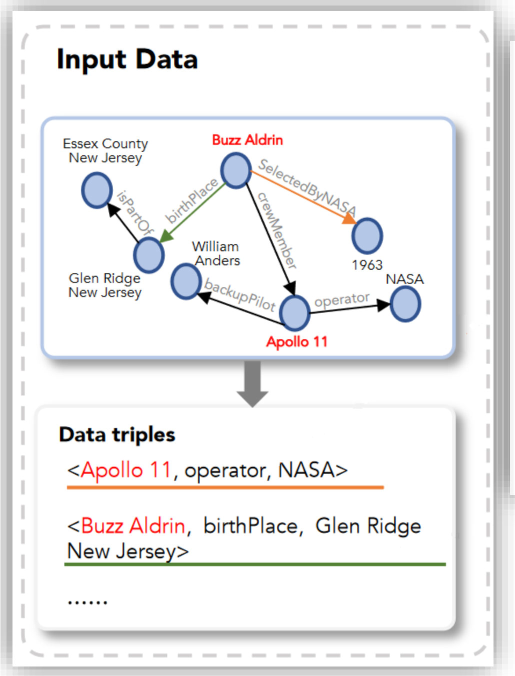
  - **Text Enhanced Graph Reconstruction:** recover the masked entities and relations in the linearized knowledge graph.



(b) Text Enhanced Graph Reconstruction

$$\mathcal{L}_{graph} = -\log P(\mathcal{G}|\hat{\mathcal{G}}, X)$$

Self-supervised Graph Masking



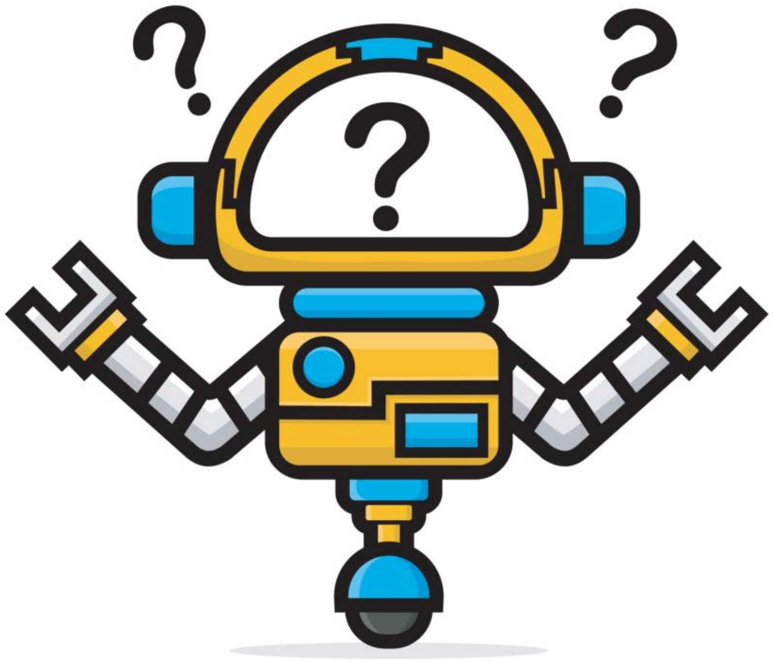
<table>
<tr><th>Pre-training Task</th><th>Input (Triples format: $[\texttt{S}|\texttt{head}_1, \texttt{P}|\texttt{relation}_1, \texttt{O}|\texttt{tail}_1, l_1]$)</th><th>Target Output</th></tr>
<tr><td>Triple Prediction</td><td>[<X>, 1], [S | New York City, P | country, O | United States, 2], [S | New York City, P | is Part Of, O | Manhattan, 2], [S | Manhattan, P | leader Name, O | Cyrus Vance Jr., 3], [S | Manhattan, P | is Part Of, O | New York, 3]</td><td><X> [S | Asser Levy Public Baths, P | location, O | New York City] <Z></td></tr>
<tr><td>Relation Prediction</td><td>[S | Asser Levy Public Baths, P | location, O | New York City, 1], [S | New York City, <Y>, O | United States, 2], [S | New York City, P | is Part Of, O | Manhattan, 2], [S | Manhattan, P | leader Name, O | Cyrus Vance Jr., 3], [S | Manhattan, P | is Part Of, O | New York, 3]</td><td><Y> P | country <Z></td></tr>
<tr><td>Triple Prediction + Relation Prediction</td><td>[<X>, 1], [S | New York City, P | country, O | United States, 2], [S | New York City, P | is Part Of, O | Manhattan, 2], [S | Manhattan, <Y>, O | Cyrus Vance Jr., 3], [S | Manhattan, P | is Part Of, O | New York, 3]</td><td><X> [S | Asser Levy Public Baths, P | location, O | New York City] <Y> P | leader Name <Z></td></tr>
</table>

Table: The input-output format for graph masking strategies.

We want to minimise the negative log-likelihood of the masked part of the graph:

$$\mathcal{L}_{GMP} = -\sum_{i=1}^{N} \log p(m_i | x_i)$$

Quering the Knowledge Base



Introducing Q&A with structured data

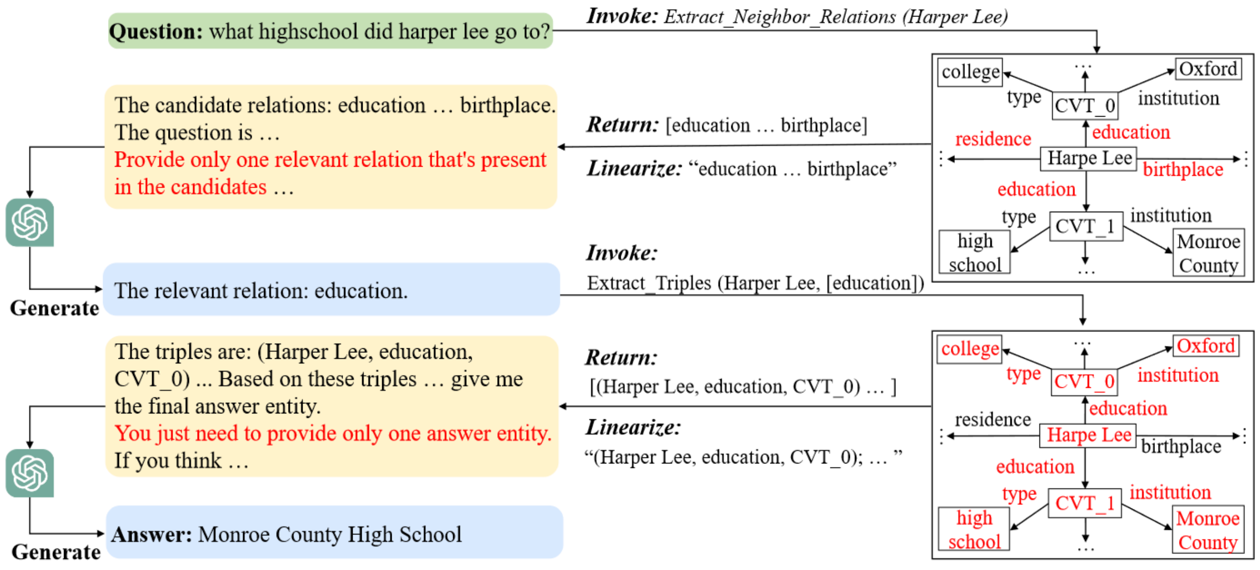
- So far, we have seen how text can be generated from semantically structured data.

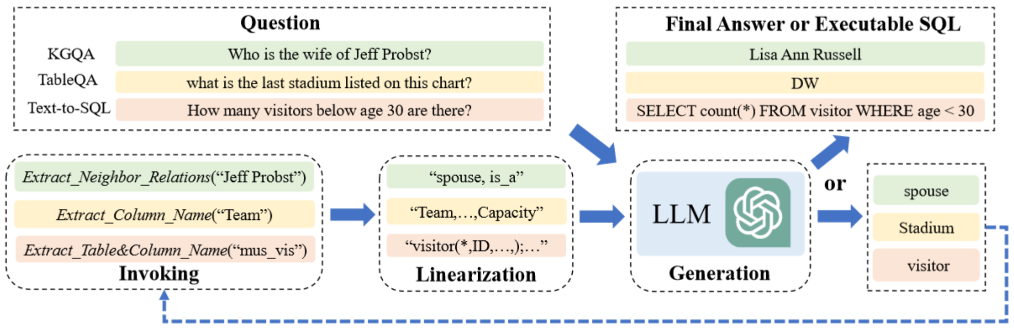
- The next step is to combine user's questions and text generation to obtain an answer.

- The aim is to prompt a language model in such a way that it can generate responses leveraging on the information that is contained in the knowledge base.

Q&A on knowledge graph: StructGPT

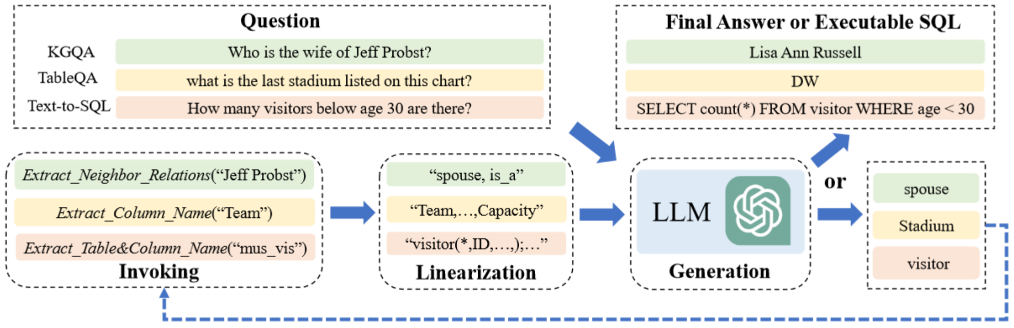
- We encode the question q into an entity e_t in such a way it can be connected to the KG.
- Starting from e_t, we perform the invoking-linearization-generation procedure two times using the two interfaces in KG sequentially.
- First, we extract the candidate one-hop relations then leverage the LLM to select the useful relations {r}.
- Then, based on {r}, we collect the relevant triples for the head entity e_t, and relation in {r} and finally employ the LLM to select the most relevant triples.





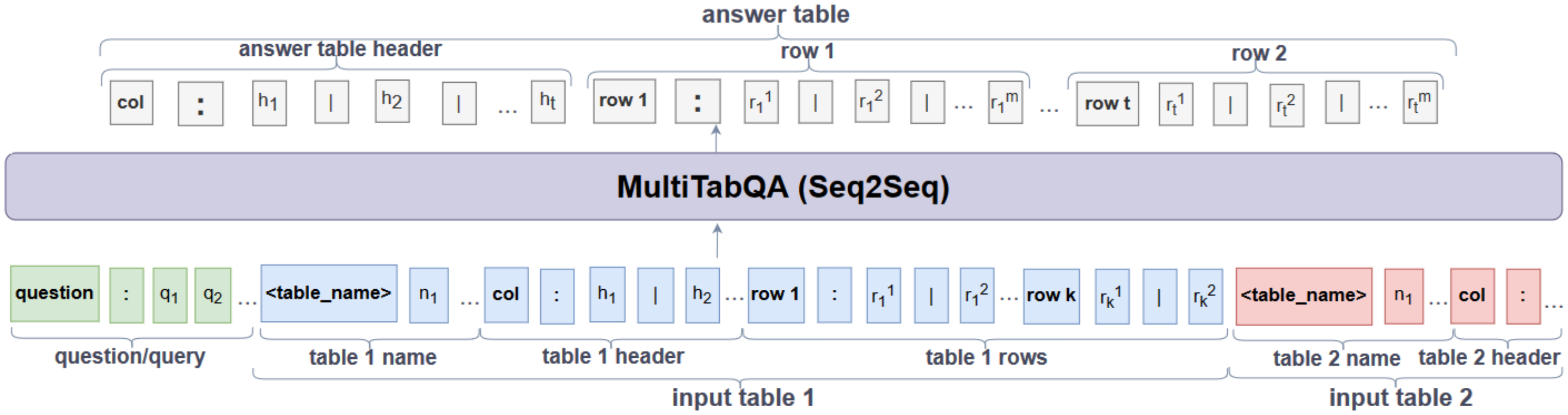
Q&A on tabular data: StructGPT

- First, we extract all column names of a table, linearize them, and leverage LLMs to select the relevant ones {c} according to the question.
- Then, we extract the contents of all relevant columns, and select the useful row indices {j} by LLMs.
- The selected columns and rows indexes are used for creating a sub-table, from the original one.
- Based on the linearized sub-table, the LLM finally generates the answer to the question.



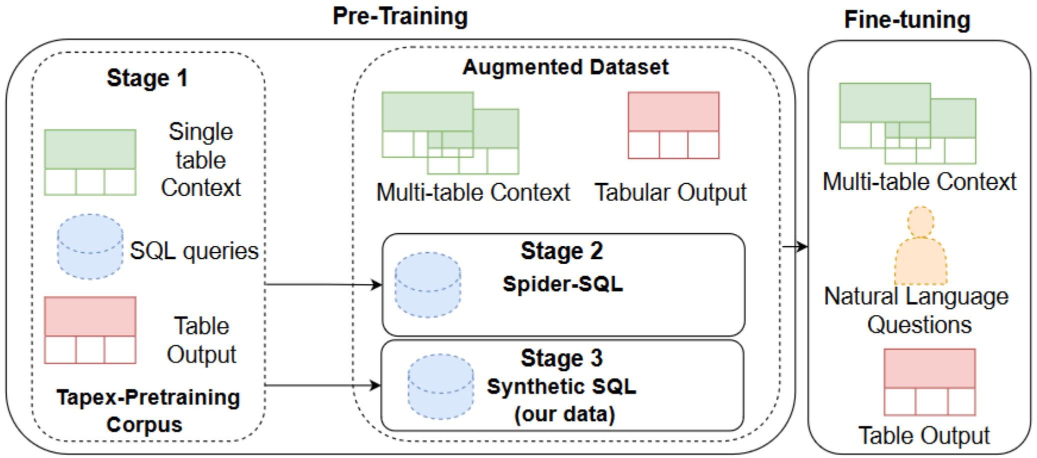
Q&A on (multi-)tabular data: MultiTabQA

- Given a question Q and a set of N tables T_n the goal of the multi-table QA model is to perform chains of operations over T_n, constrained by Q, and generate a table T_{out}.
- This model always generate a table as output, thus it does not generate text.



Q&A on (multi-)tabular data: MultiTabQA [cont.]

- Pre-training phase follows a curriculum learning approach:
    1. Stage 1 is single table QA where the model learns to generate tabular answers from simple SQL queries.
    2. Stage 2 is multi-table QA where the model trained in Stage 1 is further tuned for multi-table SQL QA.
- Final stage of training is fine-tuning the pretrained model on natural language questions.



Open problems

Open problems

- In the approaches we considered, entities constituting a knowledge graph are represented by names.
- How could we properly represent documents as entities?
- What is the best method for generating text from structured data?
- Is there a way to adapt the pre-training phase used for KG to document-based KG?

Open problem - documents

- A short textual context can be represented as an embedding.
- Documents may contain several pages. How could we represent them?
  - Summarizing and then embedding?
  - Are metadata useful for this scope?
  - How much information is lost?

- Current methods linearize the entities and relations of a KG, how could we do the same with documents?
  - This problem has an impact both on the text generation and on the pre-training phase.

Open problem – data disambiguation

- The knowledge acquired by an LM is crucial for the generation of the text, especially for the disambiguation of the entities and relations.
- How much performance do we lose when working on niche topics or with new terms? (zero-shot)
- An LM may need to be trained with a continual learning technique, both to learn new terms and to remember those previously learned.
- Multi-hop reasoning: how can the relationship between far-away entities be captured efficiently?

Thank you for your attention!