Grounding Large Language Models in Reaction Knowledge Graphs for Synthesis Retrieval

Olga Bunkova, Lorenzo Di Fruscia*, Sophia Rupprecht,
Artur M. Schweidtmann, Marcel Reinders and Jana M. Weber
Technical University of Delft, Delft, The Netherlands
*E-mail: **l.difruscia@tudelft.nl**

Chemical synthesis in the era of LLMs

Chemical synthesis is the process of planning and executing reactions to produce target molecules (medicines, materials, etc.)[1]

LLMs could aid in identifying chemical pathways from language prompts

Shortcomings of *out-of-the-box* LLM prompting include:

- Hallucination
- Limited domain knowledge
- Weak multi-step reasoning

Can we ground LLMs in chemical reactions databases?

Data representation

Chemical reactions can be represented as graphs

SMILES notation[2]

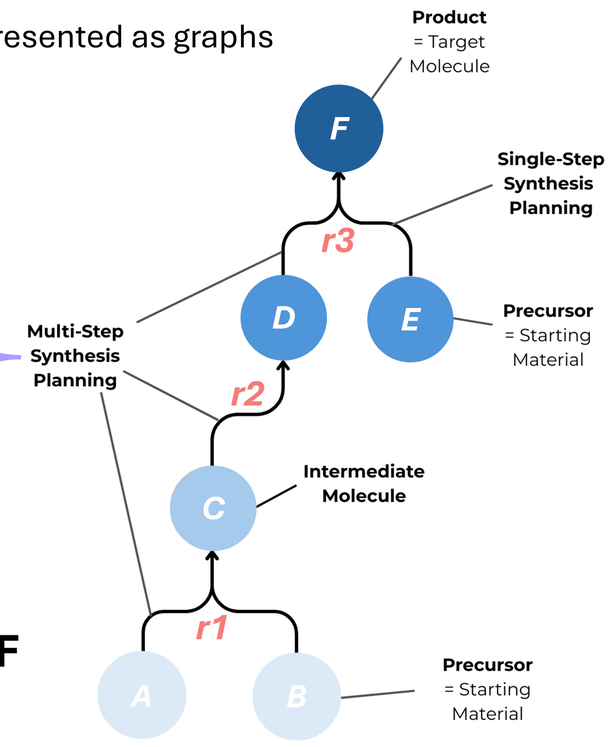
CO . CC(=O)O >> CC(=O)OC

r1 : A + B → C
r2 : C → D
r3 : D + E → F

Reaction Equations

Directed Bipartite Graph

Good choice for reaction pathway encoding



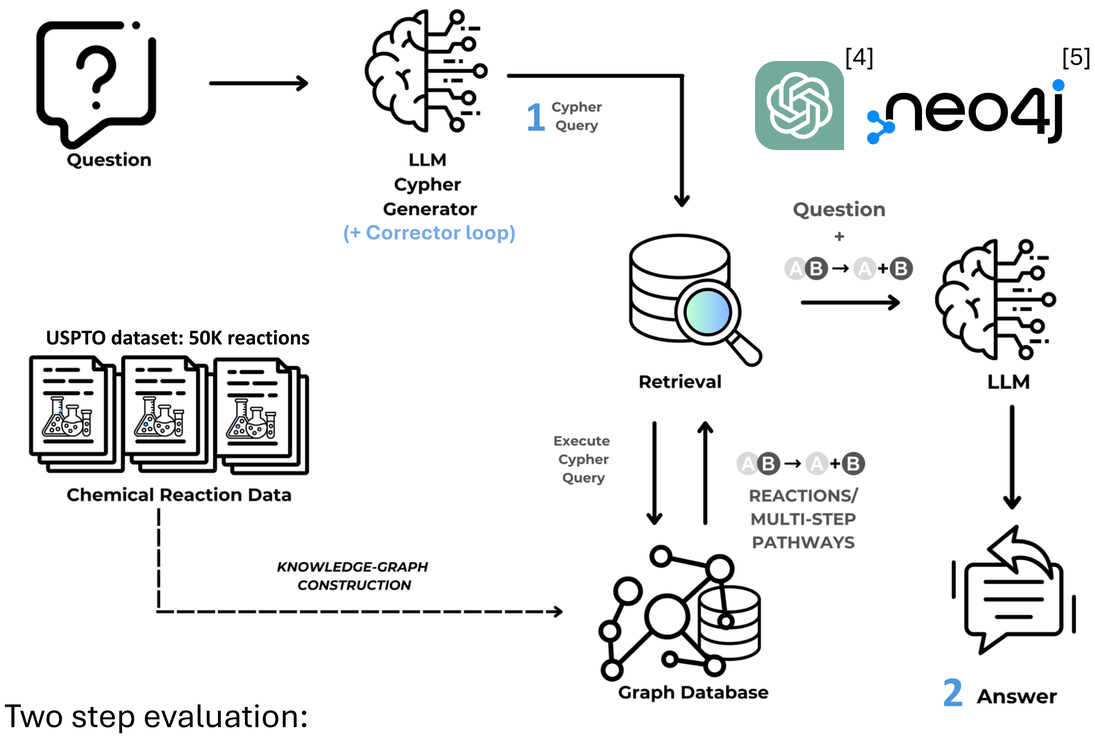
Text-to-Pathway pipeline

- **Text2Cypher:** LLM translates natural language to a graph query language (*Cypher*[3])
  - **Corrector loop:** LLM checks and self-corrects graph queries over multiple attempts
- **Execution:** LLM executes the graph query on the chemical bipartite graph
- **Extraction:** LLM extracts molecules and reactions `IDs` and provides an answer

What are the direct precursors used to synthesize {molecule}?

```
MATCH (target:Molecule
{name: '{molecule}'})<-
[:PRODUCES]-(r:Reaction)
```

The direct precursors are: {id1: {mol}, id2: …}



Two step evaluation:

1. Text-to-Text generation → Metrics: **Bleu, Meteor, Rouge-L** ∈ [0,1] [6]
2. Retrieval from executed queries

Single-Step retrieval format

```
{
  "reaction_id": id,
  "products":   [SMILES1],
  "reactants":  [SMILES2, SMILES3],
  "agents": [SMILES4, SMILES5],
  ...
}
```

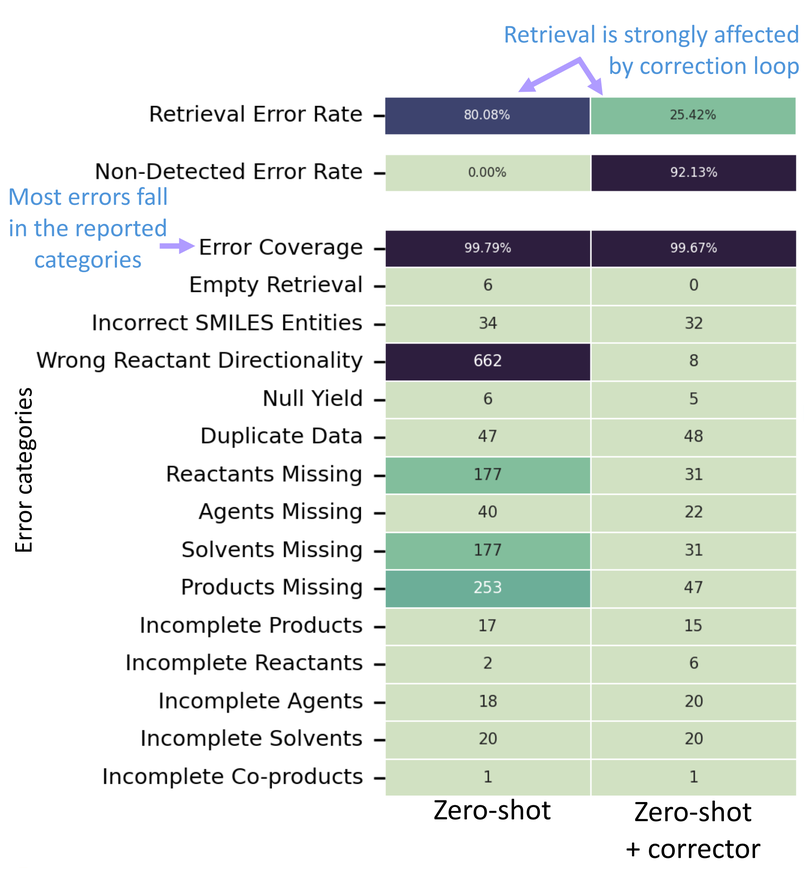
Multi-Step retrieval format

```
{
  "reaction_nodes": [
  {"id": id1, "type": "Reaction"}
  {"id": id2, "type": "Reaction"}
   ...]
}
```

→ Metrics: Keys and path matching

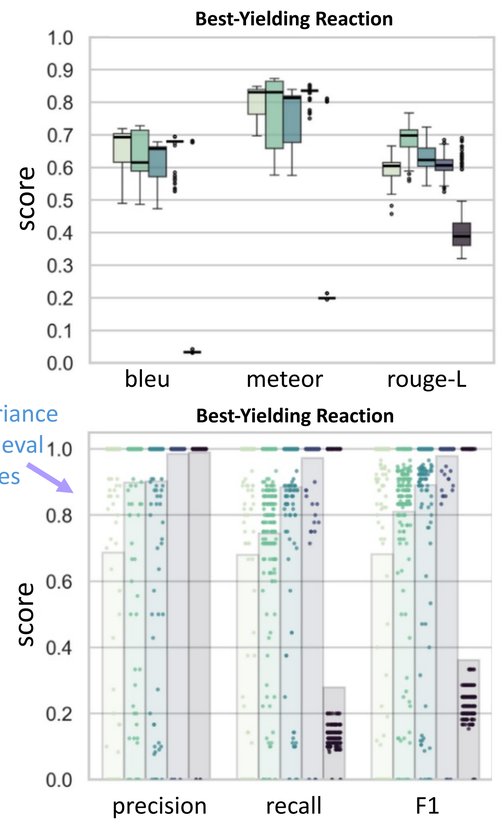
Results

Self-correction recovers missing molecules



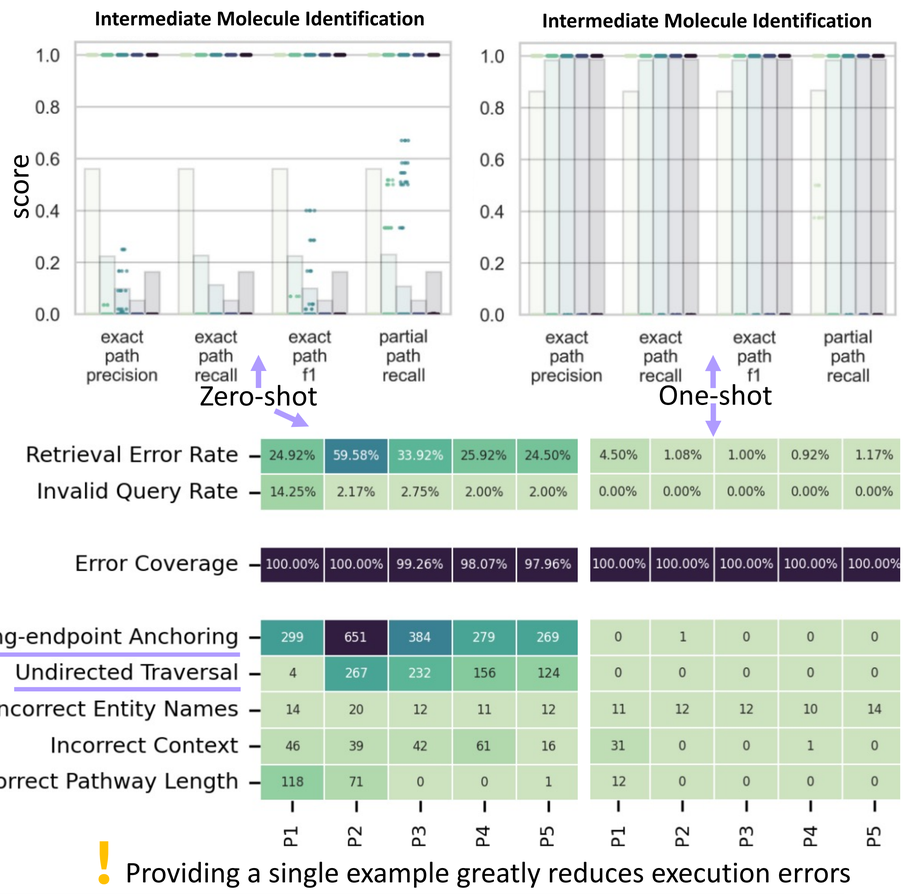
! The corrector focuses on directionality and missing molecules errors

Query similarity is not a strong retrieval proxy



! Cypher queries can be semantically similar while altering the retrieval

One shot prompting helps multi-step graph traversal



! Providing a single example greatly reduces execution errors

Prompt version (increasing complexity)
1 – Base Contextual | 2 – 1 + Explicit MATCH Order | 3 – 2 + Direction | 4 – 3 + Yield | 5 – 4 + Full Context Rules

Conclusions

- **Our results provide practical guidelines for reaction planning and database retrieval with LLMs**
- **Self-correction** is beneficial, however it mostly helps with generic errors: we suggest to make the corrector task-aware
- **Evaluate by execution**: high text similarity can coexist with wrong retrieved answers, thus pipeline that goes beyond query semantics is crucial
- **Exemplar-guided prompting** alone solves most graph traversal errors

References

[1] E. J. Corey, "*Computer-Assisted Design of Complex Organic Syntheses*", Science, 1969
[2] D. Weininger, "*SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules*", J. Chem. Inf. Comput. Sci., 1988
[3] N. Francis *et al.*, "*Cypher: An Evolving Query Language for Property Graphs*", SIGMOD '18, 2018
[4] Brown et al. *"Language Models Are Few-Shot Learners." arXiv, 2020*
[5] Neo4J, https://neo4j.com/
[6] M. Evtikhiev et al., "*Out of the BLEU: how should we assess quality of the Code Generation models?*," Journal of Systems and Software, 2023

The Delft Bioinformatics Lab