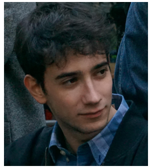
Natural Language Interfaces for Databases with Deep Learning

George Katsogiannis-Meimarakis, Mike Xydas, and Georgia Koutrika
Athena Research Center, Athens, Greece

49th International Conference on Very Large Data Bases, Vancouver, Canada, 2023

Presenters

George Katsogiannis



- **Research Associate** at Athena Research Center
  - Natural Language Data Interfaces
  - INODE and FC4EOSC projects
- Graduate of MSc Data Science
- Upcoming PhD in Fall of ‘23

Mike Xydas

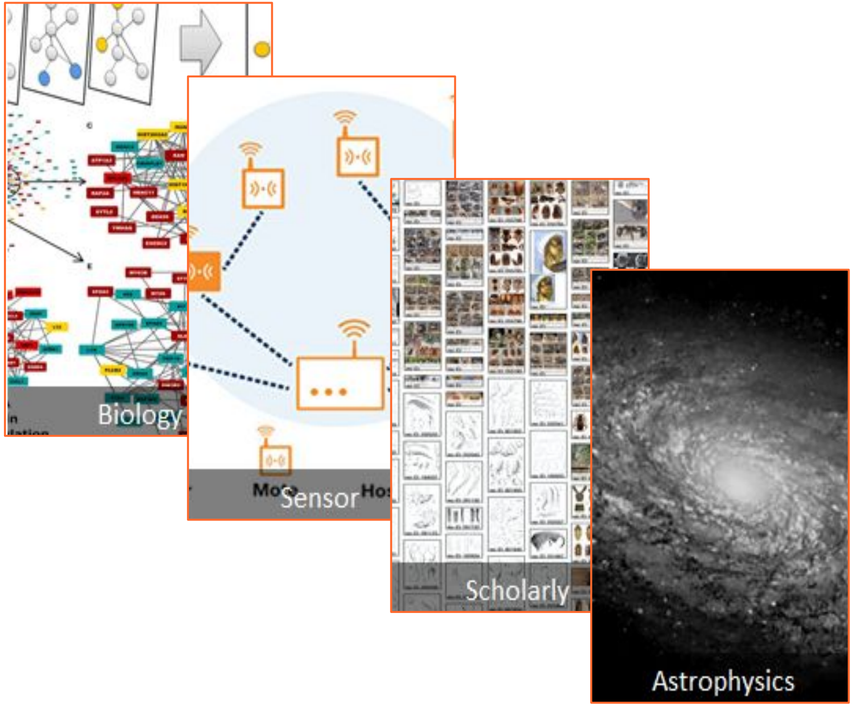
- **Research Associate** at Athena Research Center
  - Natural Language Data Interfaces
  - INODE and EOSCF projects
- Graduate of MSc Data Science

Georgia Koutrika

- **Research Director** at Athena Research Center
- Research interests:
  - Conversational data systems
  - Intelligent Data Exploration
  - User-driven data management

Why Natural Language Interfaces for DBs?

- The imminent age of information has made data an indispensable part of all human activities
- Many different data sets are being generated by users, systems and sensors
- Databases can benefit many types of users looking for insights, patterns, information, etc.
- However, not all users have equal access to data



Why Natural Language Interfaces for DBs?

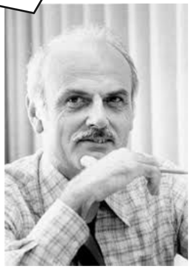
What if we could simplify the interaction and enable users to access DBs with Natural Language?

Which cities have year-round average temperature above 50 degrees?

Athens and Singapore have an average yearly temperature higher than 50°F

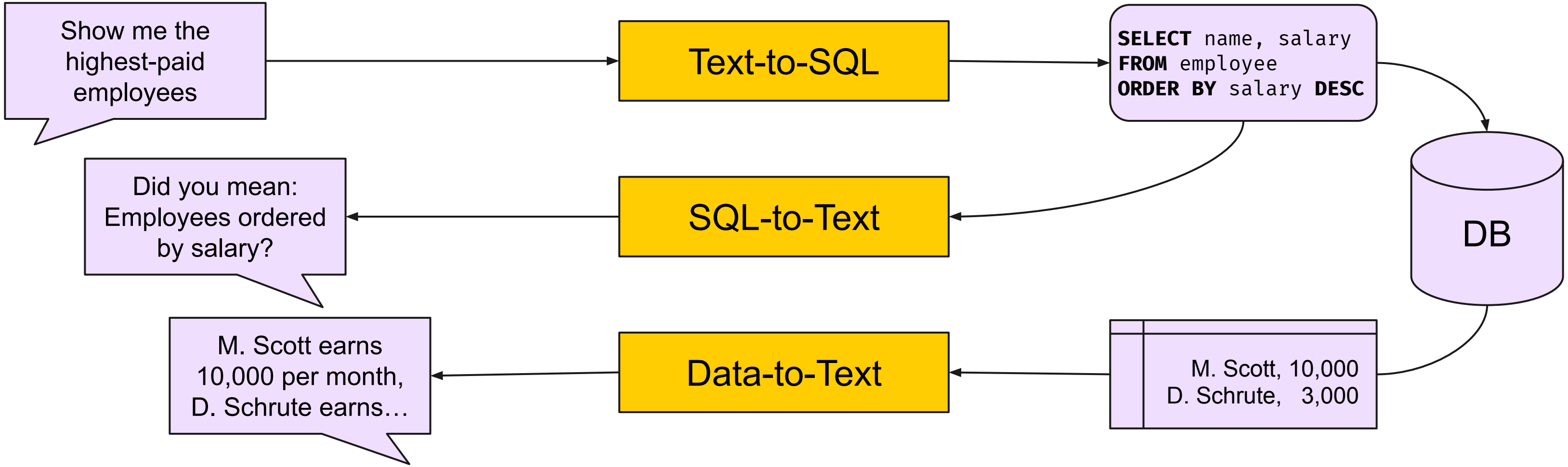
To satisfy the needs of casual users of databases, we must break through the barriers that presently prevent these users from freely **employing their native languages**

Ted Codd (circa: 1974)



Interacting with natural language can open up data access to everyone

[1] Seven steps to rendez-vous with the casual user (1974)



Natural Language Interfaces for DBs

Tutorial *Agenda*

01
Introduction

02
Text-to-SQL

03
SQL-to-Text

04
Data-to-Text

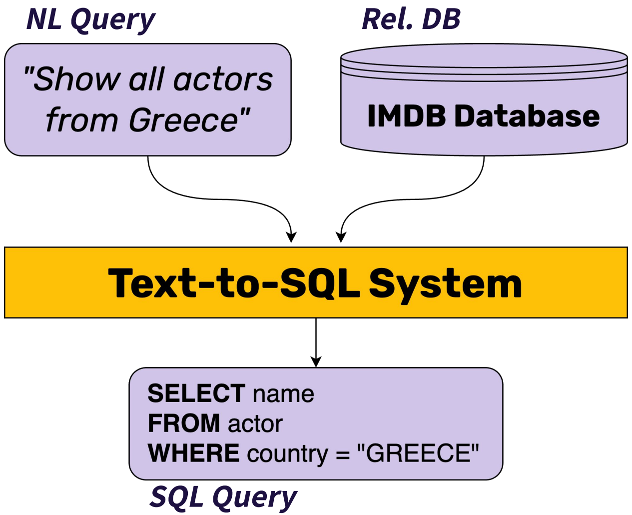
05
Bringing it all Together

06
QnA

Text-to-SQL

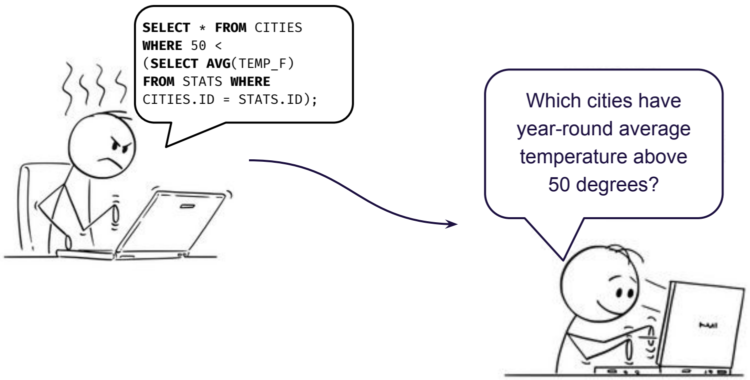
The Text-to-SQL Problem

“Given a **Natural Language Query (NLQ)** on a **Relational Database (RDB)**, produce a **SQL query** equivalent in meaning, which is **valid** for the said RDB and will return results that **match the user’s intent**.”



The Text-to-SQL Problem

- The text-to-SQL problem has long been a holy grail for the DB community
- It would allow users to query DBs without any technical skills
- There have been many efforts from the DB community during the past decades
- However this is a notoriously difficult problem



Users can query the DB with Natural Language instead of SQL

Challenges: From the NL side

- Complexity of Natural Language

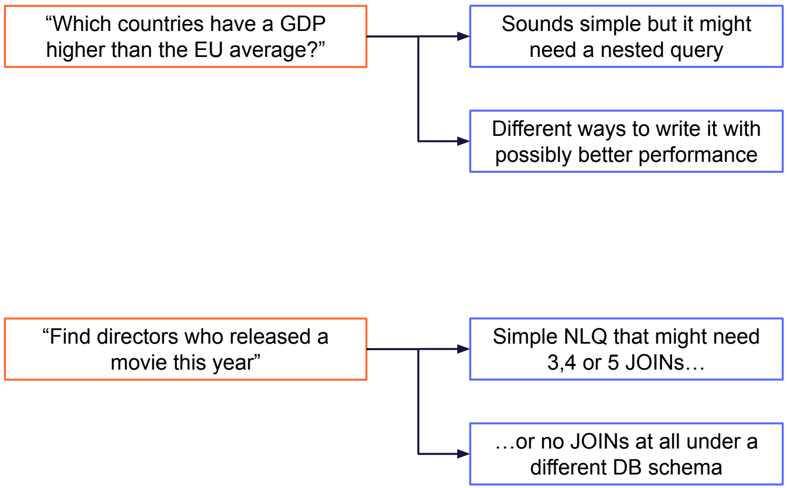
Ambiguity	"Show information about Paris"	Paris is a city or a person?
References - Schema Linking	"How many models are there?"	car.model OR engine.model?
Inferences	"Who was president before Obama?"	President of the USA is inferred
Vocabulary Gap	"Return all composers"	DB table is called songwriters

- User Mistakes

Spelling mistakes	"Which singer won the most Grammies?"	Grammys
Syntactical/Grammatical mistakes	"Show actor most played movies "	??!?

Challenges: From the SQL side

- Complex Syntax
  - SQL is a structured language with a strict grammar and limited expressivity
  - The same query can be written in different ways
- Database Structure
  - The user's data model may not match the data schema
  - The same query must be written differently for different schemas

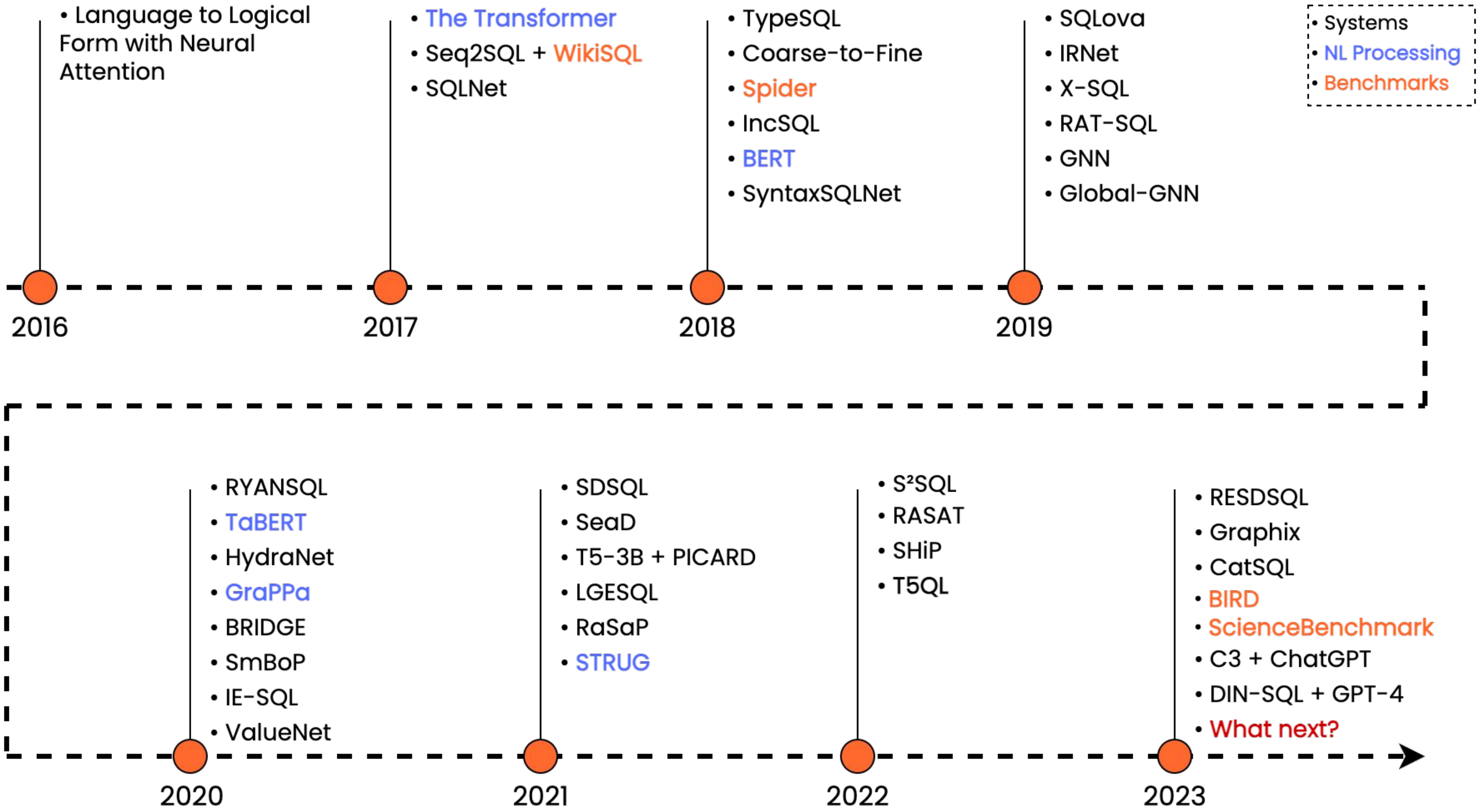


Text-to-SQL: Early Approaches

- Keyword Systems [2, 3, 4]
  - Search engine-like functionality, where NLQs contain just keywords
  - e.g., *“drama movies”*
- Enhanced Keyword systems [5, 6]
  - Queries with aggregate functions, comparison operators, and keywords that map to database metadata
  - Syntactic constraints on their input to make sure they can parse the user query
  - e.g., “*count movies actress “Priyanka Chopra””*
- Natural language systems [7, 8]
  - Allow queries in natural language
  - e.g., *“What is the number of movies of “Priyanka Chopra””*

Syntactic parsers, ontology mappings, knowledge bases, information retrieval…

Why not use deep learning and treat it like a translation problem from NLQ to SQL?

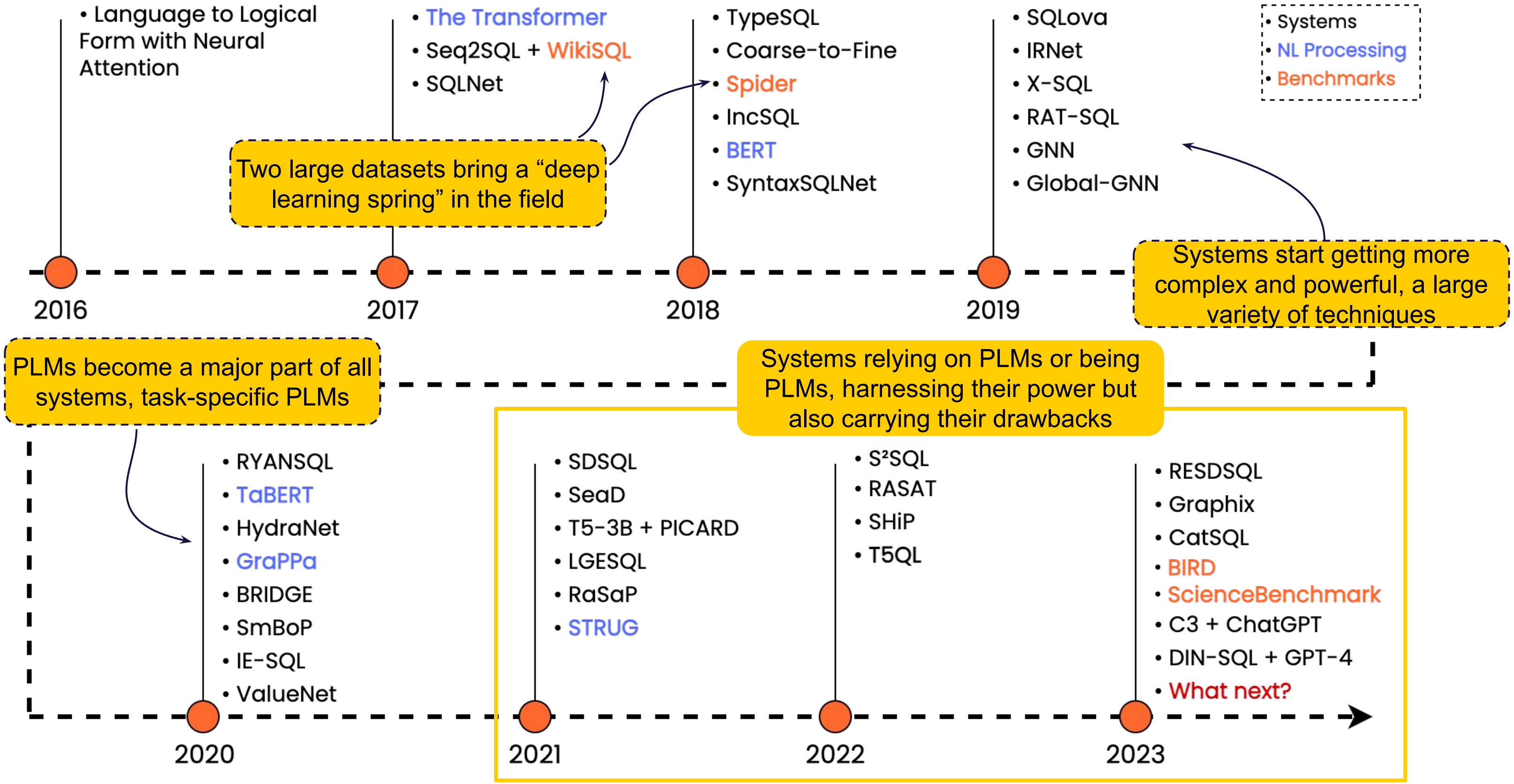


Timeline of Deep Learning Text-to-SQL

Text-to-SQL Datasets

- Several pain points in early evaluation
  - No common datasets
  - Small or proprietary datasets
  - Incomparable effectiveness evaluations
- Two new large cross-domain benchmarks
  - Revolutionise text-to-SQL research
  - Open the door to deep learning
- **WikiSQL** is simpler and serves as a starting point
- **Spider** introduces full DBs and more complex SQL and becomes the standard
  - But it is clear we need to keep moving
- New benchmarks have been proposed but are not yet widely adopted

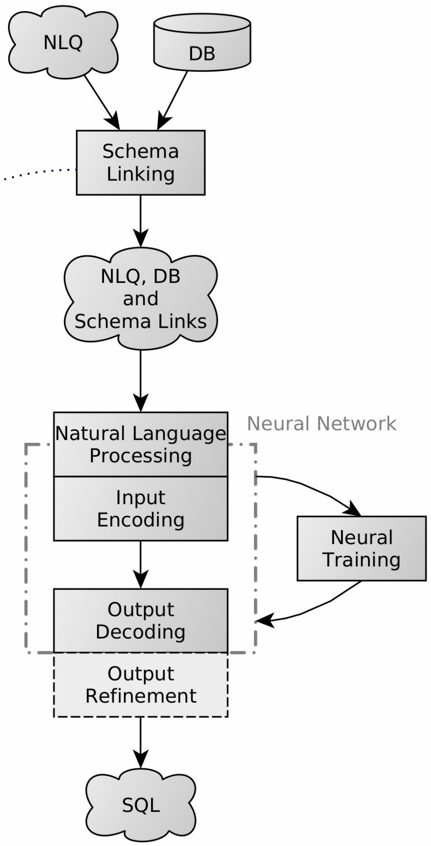
Year	Dataset	Examples	Databases
1994	ATIS	275	1
1996	GeoQuery	525	1
2003	Restaurants	39	1
2014	Academic	179	1
2017	IMDb	111	1
	Yelp	68	1
	Scholar	396	1
	WikiSQL	**80,654**	**24,241**
2018	Advising	281	1
	Spider	**10,181**	**200**
2020	MIMICSQL	10,000	1
	SQUALL	11,276	1,670
	FIBEN	300	1
2021	Spider-Syn	8,034	160
	Spider-DK	535	?
	KaggleDBQA	272	8
	SEDE	12,023	1
2023	ScienceBenchmark	4,985	3
	BIRD	12,751	95



Timeline of Deep Learning Text-to-SQL

Text-to-SQL Taxonomy

Discovery of possible mentions of database elements (tables, attributes and values) in the NLQ



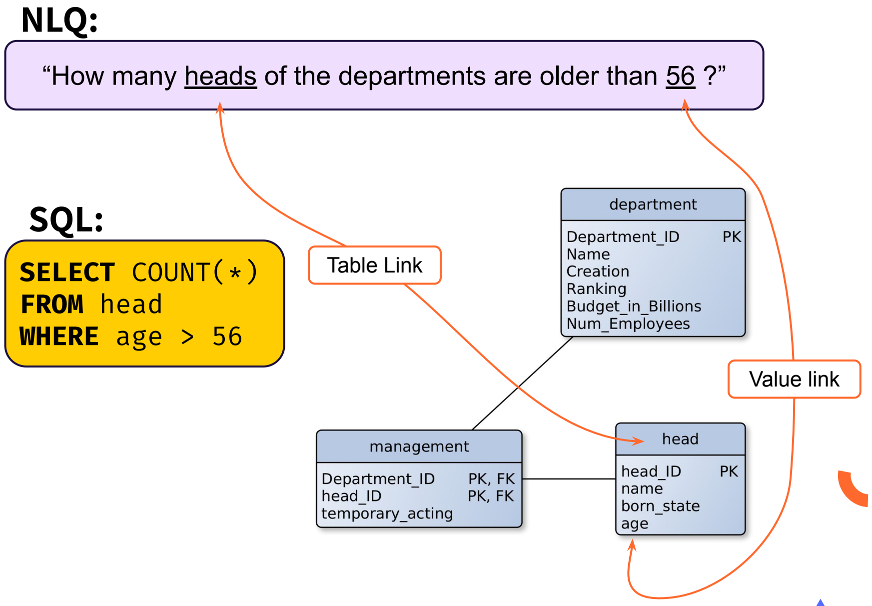
A Taxonomy of Text-to-SQL Deep Learning Systems

[9] Survey on Text-to-SQL (Katsogiannis and Koutrika, VLDBJ 2023)

Schema Linking

Finding connections between the NLQ and the DB

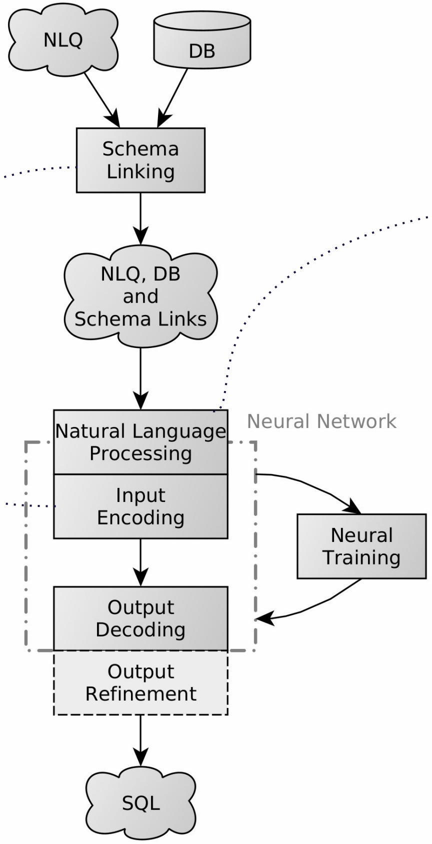
- Three main types of schema links:
  - Table links
  - Column links
  - Value links
- The three questions of schema linking:
  - Which parts of the NLQ to consider?
  - Which parts of the DB to consider?
  - How to decide on a match?
- A very difficult task that latest systems tackle with dedicated models or relying on PLMs



Examples
"track" links to table `song`
"composer" links to column `song.composer`

Discovery of possible mentions of database elements (tables, attributes and values) in the NLQ

Selection of the encoder and structuring the inputs so that they can be accepted by the encoder



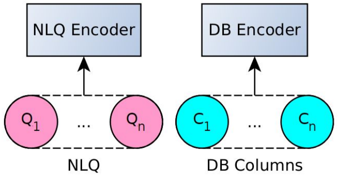
Example Choices
Word Embeddings (GloVe, Word2Vec),
Pre-trained Language Models (BERT, T5)

Creating an efficient numerical representation that can be read by the encoder

A Taxonomy of Text-to-SQL Deep Learning Systems

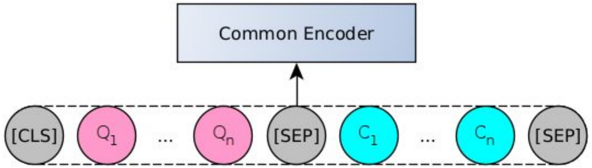
Input Encoding

Separate Encoding



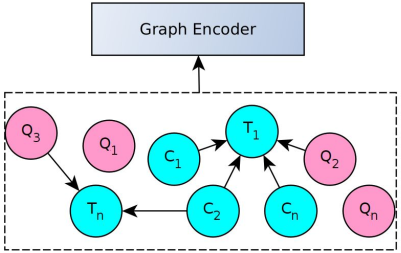
Mostly used by earlier systems

Serialised Encoding

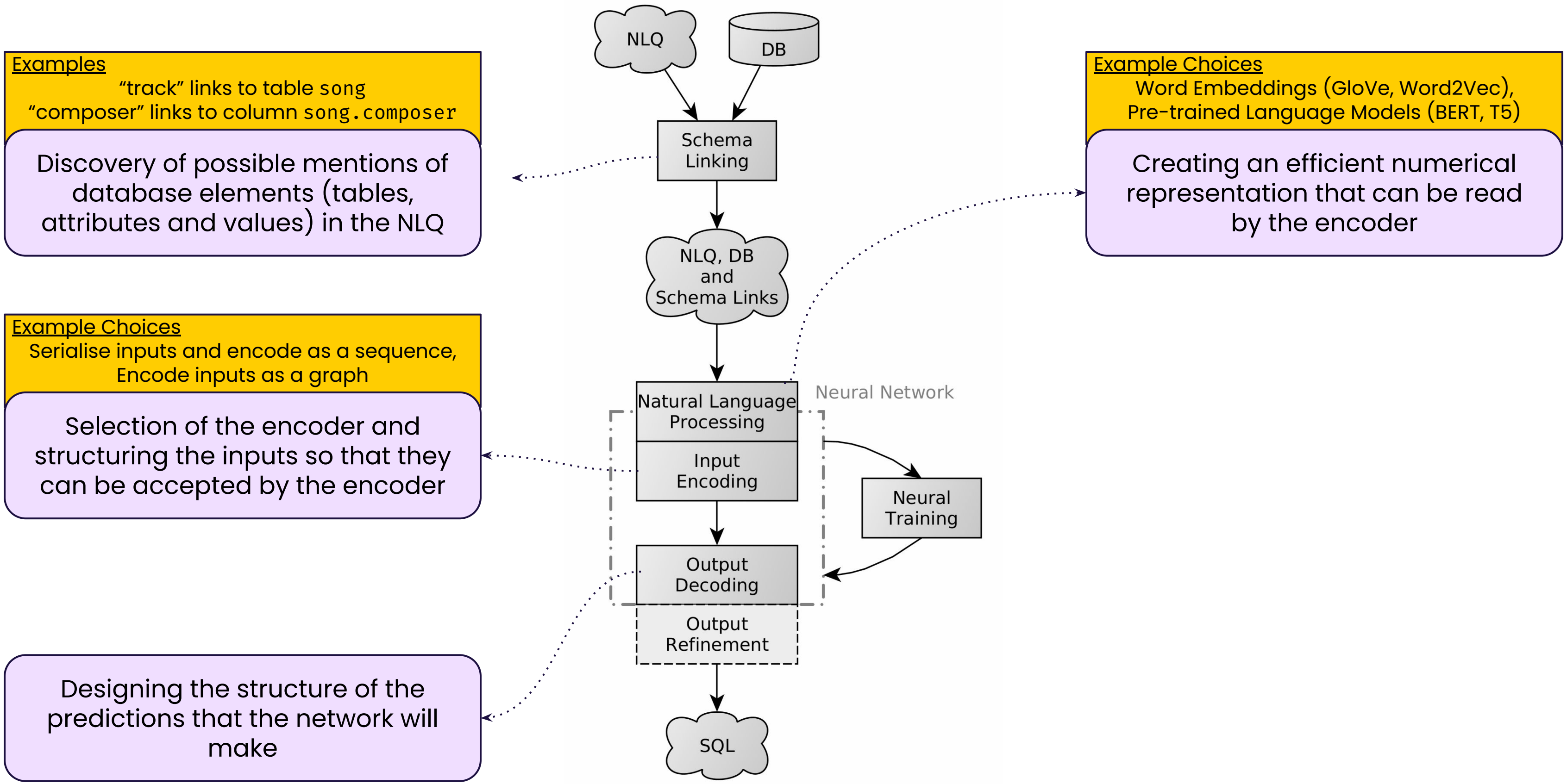


Very easy to use with PLMs

Graph Encoding



More intricate but can retain a lot of structural information



A Taxonomy of Text-to-SQL Deep Learning Systems

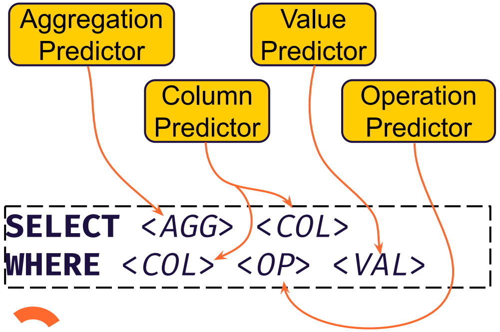
[9] Survey on Text-to-SQL (Katsogiannis and Koutrika, VLDBJ 2023)

Output Decoding

Sketch-based

✅ Simplifies the problem by breaking it down

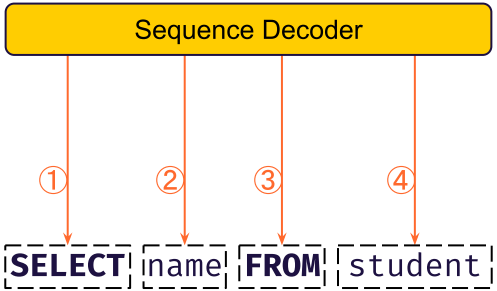
❌ Extending to more complex queries is far from trivial



Sequence-based

✅ Simplest approach, works with off-the-shelf models

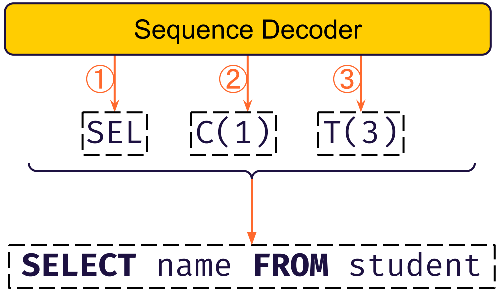
❌ Nothing prevents the decoder from generating SQL queries with errors



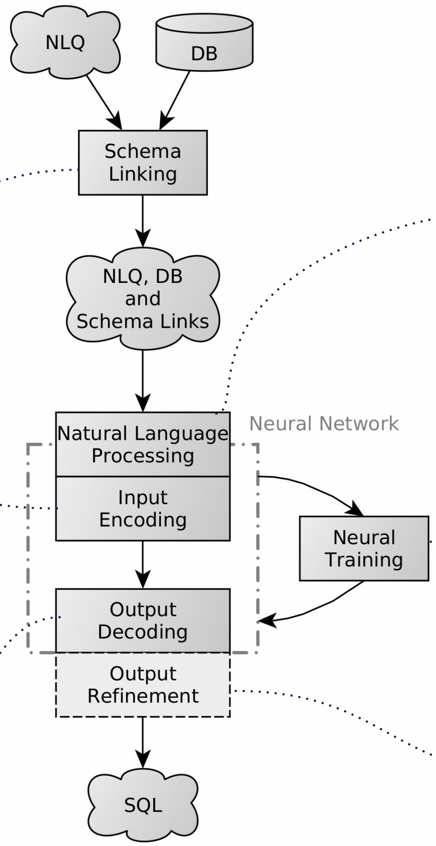
Grammar-based

✅ Grammar guarantees the correctness of SQL

❌ Requires specifically designed decoder, not easy to use pre-trained



A Taxonomy of Text-to-SQL Deep Learning Systems



Examples
"track" links to table `song`
"composer" links to column `song.composer`

Discovery of possible mentions of database elements (tables, attributes and values) in the NLQ

Example Choices
Serialise inputs and encode as a sequence,
Encode inputs as a graph

Selection of the encoder and structuring the inputs so that they can be accepted by the encoder

Example Choices
Generate a SQL query as a text sequence,
Generate SQL using grammar rules

Designing the structure of the predictions that the network will make

Example Choices
Word Embeddings (GloVe, Word2Vec),
Pre-trained Language Models (BERT, T5)

Creating an efficient numerical representation that can be read by the encoder

Example Choices
Training from scratch, Transfer Learning (e.g., fine-tuned PLMs), Prompt-engineering

How to train the network for the best possible performance

Example Choices
Re-ranking predictions, constrained decoding, correcting predictions

Additional techniques that can be added after training to improve performance

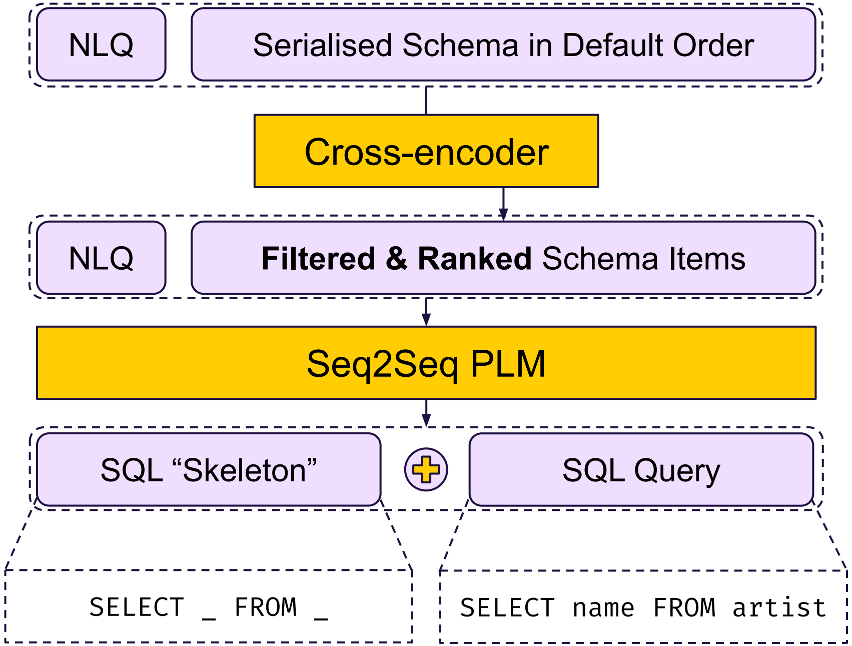
[9] Survey on Text-to-SQL (Katsogiannis and Koutrika, VLDBJ 2023)

Key Approaches

RESDSQL

Introduces two techniques to improve the performance of PLMs in Text-to-SQL:

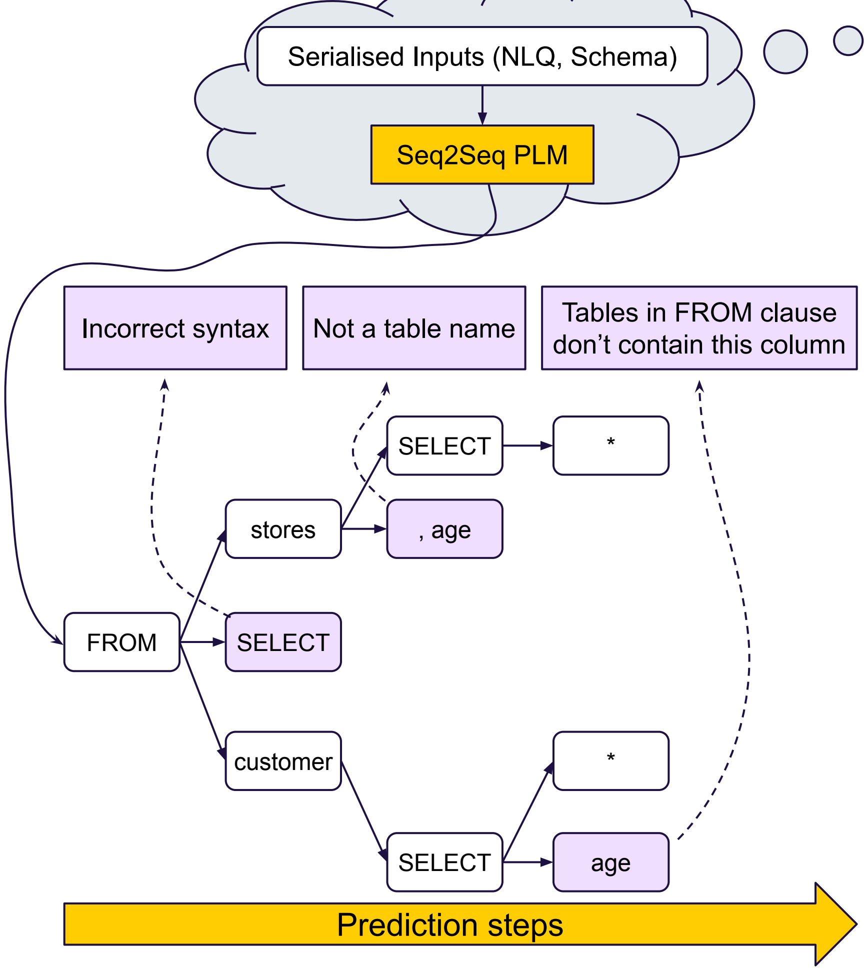
- Filtering and ranking schema items
  - The PLM only sees the most relevant columns and tables
- Separate skeleton and query prediction
  - The PLM's decoder starts by predicting the skeleton of the query
  - The actual query is generated after the skeleton



Schema Linking	Nat. Language Representation	Input Encoding	Output Decoding	Neural Training	Output Refinement
Filtering and Ranking	Encoder - Decoder PLM	Serialised	Sequence	Fine-Tuning	None

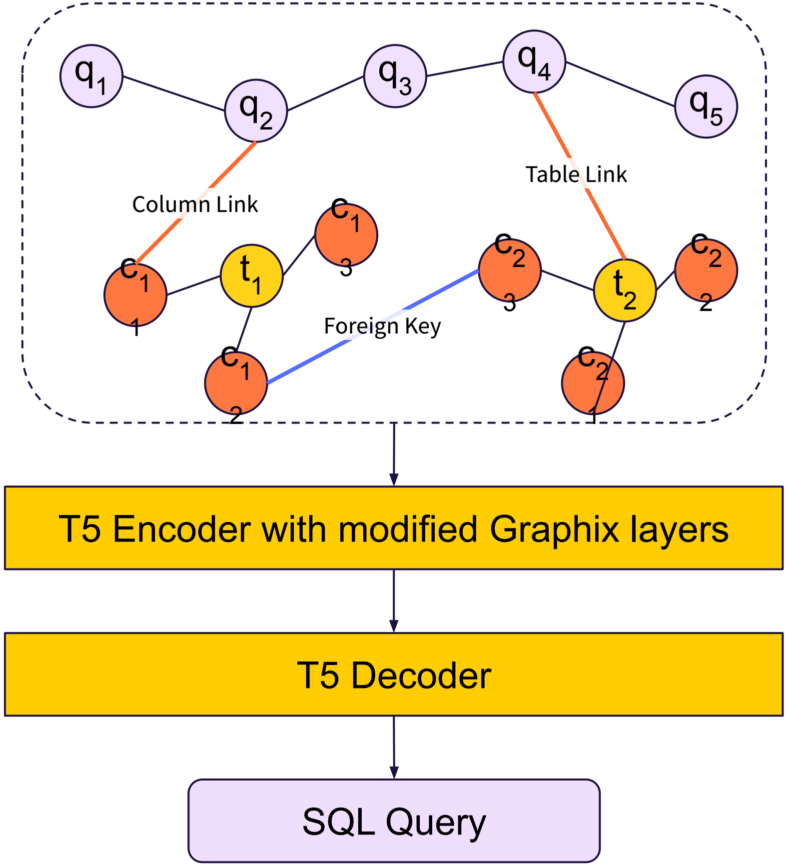
PICARD

- PICARD is a **constraining technique** for autoregressive decoders of PLMs
- Tackles the **drawbacks of sequence-based decoders**
    - Grammar and syntax errors
    - Hallucinations of non-existent attributes
- Blocks predictions that create errors
    - Checks for spelling, syntax and grammar errors
    - Checks for availability of used attributes
    - Checks the use of correct aliases
- Is frequently used by SOTA systems, but can add considerable overhead!



Graphix-T5

- Input is structured as a graph:
  - **Nodes** can be NLQ tokens, column names, and table names
  - **Edges** can store information such as foreign keys, schema links, column appearing in a table, etc.
- Encoder’s architecture is modified to use Graphix layers instead of Transformers
  - Graphix layers can process the structural information of the graph
  - Initialised with T5 weights



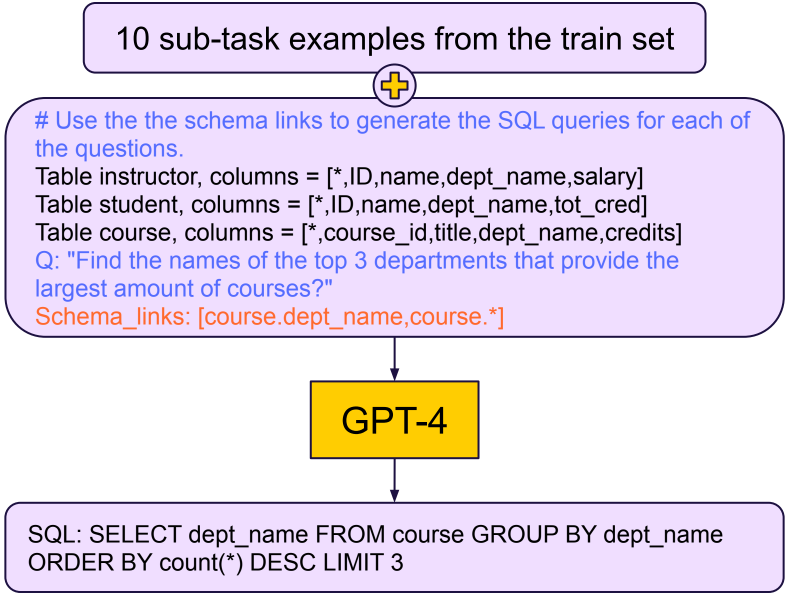
Schema Linking	Nat. Language Representation	Input Encoding	Output Decoding	Neural Training	Output Refinement
Yes	Encoder - Decoder PLM	Graph	Sequence	Fine-Tuning	None

[12] Graphix-T5 (2023)

DIN-SQL

- Using PLMs with **few-shot learning** and **decomposing** the text-to-SQL task
- Decompose the task in **4 sub-tasks**:
  - Schema Linking
  - Query Structure Prediction
  - SQL Generation
  - Self-Correction
- For each query, **prompt the model 4 times**, for each of the 4 sub-tasks
- Currently the **SOTA for the Spider** dataset, when paired with GPT-4

Schema Linking	Nat. Language Representation	Input Encoding	Output Decoding	Neural Training	Output Refinement
PLM	Encoder - Decoder PLM	Serialised	Sequence	Prompt-Tuning	PLM (Self Correction)



Research Opportunities

Better Text-to-SQL Evaluation

Researchers tend to rely only on the Spider benchmark for evaluating their systems, ignoring its drawbacks:

❌ Databases and queries created specifically for evaluating text-to-SQL systems
- They do not have the complexity of real-life databases
- They contain very little data

❌ Small number of examples for training and evaluation

- Newer systems can currently reach up to 85% accuracy on Spider
- It's high time we set new standards:

✅ Create benchmarks using real-world use cases and DBs

✅ Ask real users to provide the queries that they would want to ask the DB

✅ Include fine-grained categories to enable detailed evaluation

e.g., robustness on synonyms, misspellings, missing info, etc.

Technical Feasibility in Text-to-SQL

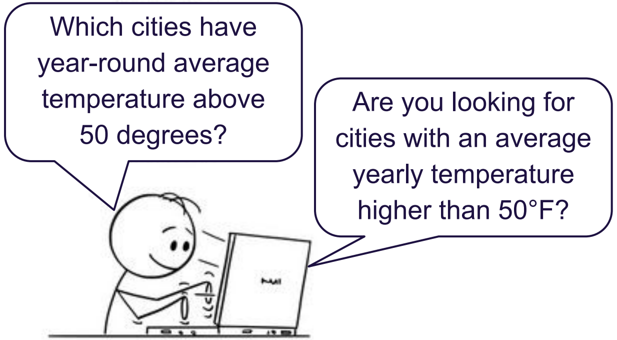
- A lot of breakthroughs have been made by using more and more intricate methods
- However, these techniques are often unrealistic for real-life applications
  - ❌ Large PLMs → Expensive infrastructure, long training, and slow predictions
  - ❌ Some DBs might contain sensitive data that prohibit the use of GPT-x models

- A lot of room for contributions in making existing techniques more robust
  - Better performance without very large PLMs
  - Optimised schema linking techniques
- It is necessary to evaluate models not only by their accuracy, but also:
  - Their size
  - Their computing requirements/cost
  - Their prediction latency/throughput

SQL-to-Text

The SQL-to-Text Problem

- Essential for explaining queries to non-technical users in a NLIDB
    - To verify the prediction of a Text-to-SQL system
    - To allow the user to choose between multiple predictions of a Text-to-SQL system
- Also useful for:
    - Automatic comment generation
    - Helping technical users understand complex queries faster
    - Data augmentation for Text-to-SQL



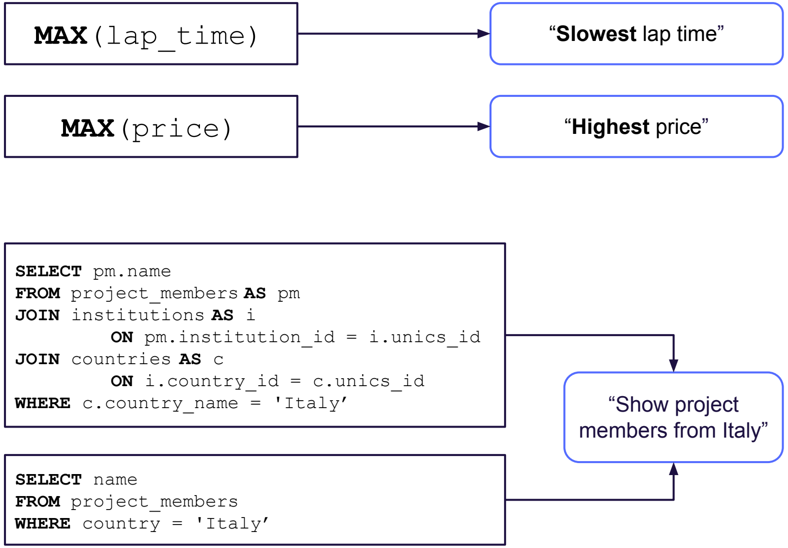
Challenges: From the NL side

- Generated NL explanations must:

Be fluent, coherent and human-like	"Show yearly average city temperatures Europe"	"Show the average yearly temperatures of European cities"
Avoid unnecessary repetitions	"Show projects that start after 2014 and before 2020"	"Show projects starting between 2014 and 2020"
Avoid over-complications	"Show me actors that play in a movie and other actors who also play in the same movie"	"Show actors that have played in the same movie"

Challenges: From the SQL side

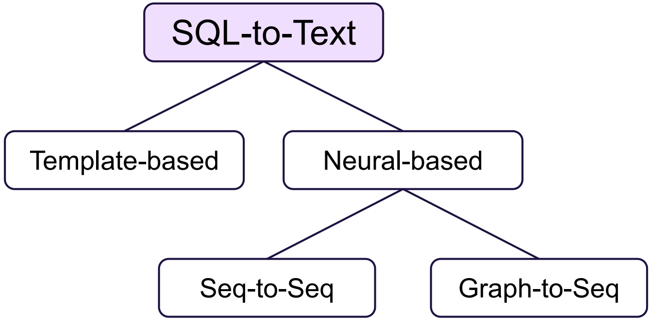
- Using the correct vocabulary based on the DB domain
- Capturing the semantics of complex SQL queries
  - Some parts of the query might not need to be explicitly verbalised
  - The same semantics might be expressed differently, in DBs with different schemas



Key Approaches

SQL-to-Text Approaches

- Has seen less attention compared to the fast-paced Text-to-SQL field
    - Only a handful of deep learning systems
    - No established benchmark or metric
- Earlier approaches used templates and rules to construct query explanations
- Recently, a few deep learning approaches have sprung, mostly motivated by data augmentation for Text-to-SQL



SQL-to-Text Approaches: Template-based

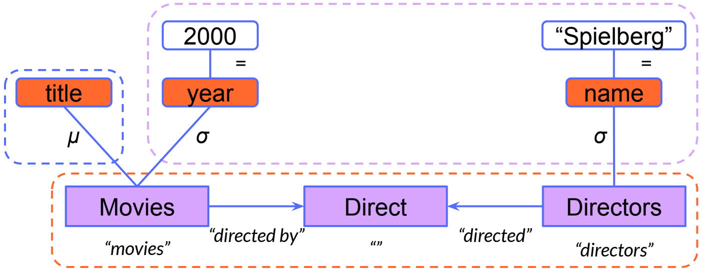
- A query graph is created based on the input query
- A set of templates for each part of DB is provided
- The query explanation is created by traversing the query graph and using the appropriate templates

✅ Very precise, since they verbalise all parts of the query

❌ A new set of templates is needed when moving to a new DB

❌ The query explanations are not fluent and realistic

```sql
SELECT m.title
FROM Movies m
JOIN Direct r
    ON m.id=r.movie_id
JOIN Director d
    ON r.director_id=d.id
WHERE d.name='Spielberg'
    AND m.year=2000
```



"Find the titles of movies that have been directed by directors. Return results only for movies whose release year is 2000 and directors whose name is Spielberg."

SQL-to-Text Approaches: Neural-based

- Two main categories of deep learning SQL-to-Text, based on **input format**:
  - Sequence-to-Sequence
  - Graph-to-Sequence
- A relatively unexplored field

✅ Can produce much more fluent and natural explanations

✅ Are easier to generalise to unseen DBs, even without human labour

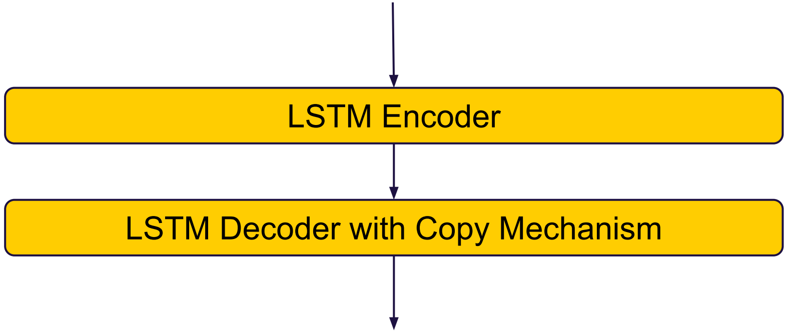
❌ Can not guarantee the precision of their explanations

Model	**WikiSQL** (BLEU)	**Spider** (BLEU)
Seq-to-Seq [15]	18.40	-
Graph-to-Seq [16] (GNN)	28.70	-
Graph-to-Seq [17] (RGT)	31.20	28.84

SQL-to-Text: Sequence-to-Sequence

- The SQL query is decoded as a text sequence
- The explanation is generated using an RNN or Transformer decoder
- Similarly to any other translation task
- Does not take advantage of the inherent structure of SQL

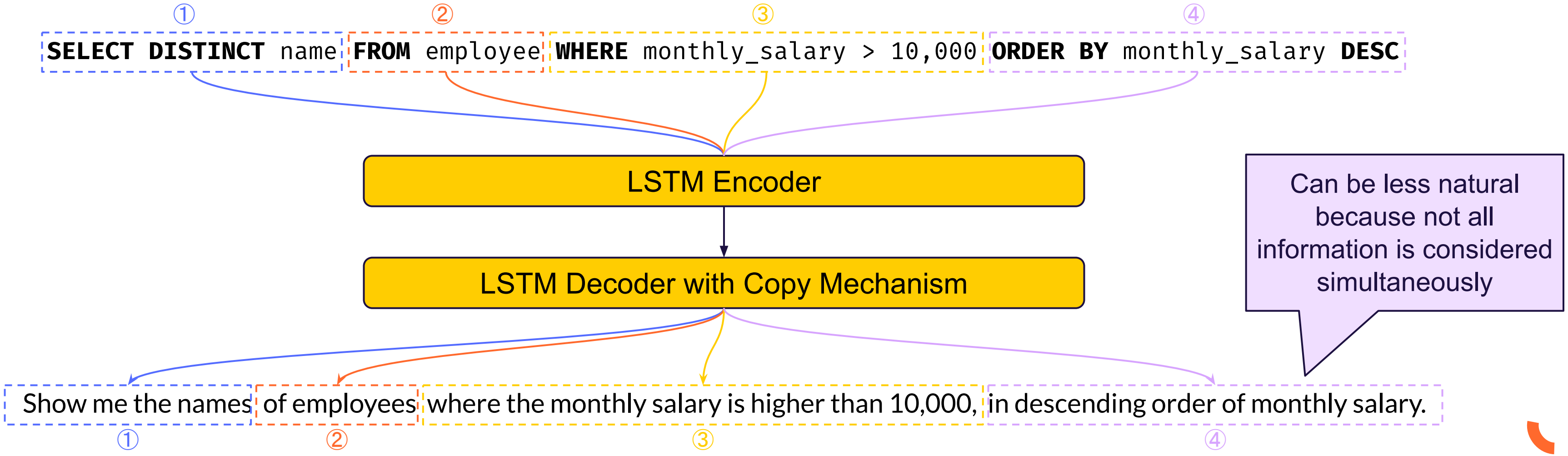
“**SELECT DISTINCT** `name` **FROM** `employee` **WHERE** `monthly_salary > 10,000` **ORDER BY** `monthly_salary` **DESC**”



“Show me all employees with a monthly salary higher than 10,000, in descending order of monthly salary.”

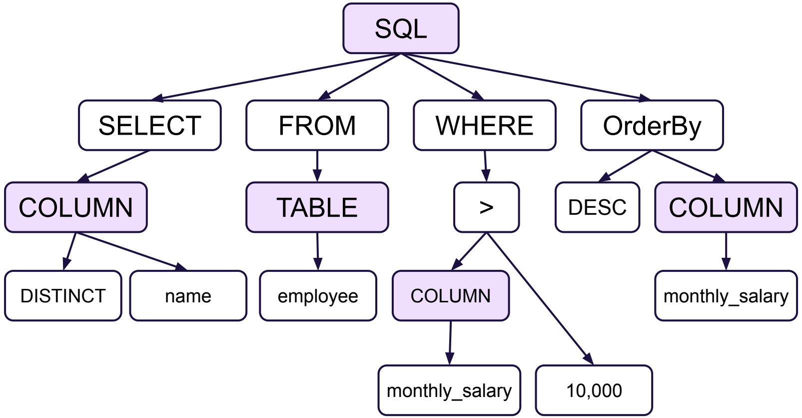
[15] Question generation from SQL queries (2018)

Hierarchical Sequence-to-Sequence



Non-hierarchical: *Show me all employees with a monthly salary higher than 10,000, in descending order of monthly salary.*

SQL-to-Text: Graph-to-Sequence



"Show me all employees with a monthly salary higher than 10,000, in descending order of monthly salary."

- The SQL query is encoded as a graph or as a tree
  - Using GNNs, or Graph Transformers
- The explanation is generated using a RNN, or Transformer-based decoder
- Different graph representation compared to the one used by template-based approaches

[17] Relation-Aware Graph Transformers for SQL-to-Text (2022)

Research Opportunities

A Metric for Query Explanations

There are no tailor-made metrics for query explanations

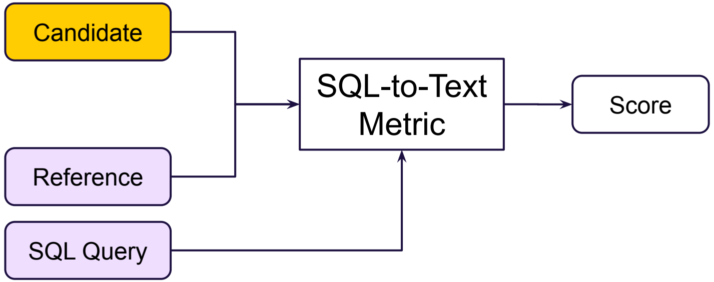
Automatic translation metrics are not robust to vocabulary differences and do not take the query into account

Ground Truth	Prediction		BLEU	chrF	METEOR
How many singers do we have?	How many songs do we have?	✗	48.89	68.49	80.66
	What is the number of singers?	✓	7.80	24.15	0.00
Tell me the age of the oldest dog.	Tell me the age of the youngest dog.	✗	66.06	72.83	86.47
	How old is the eldest dog?	✓	5.66	37.42	16.12

[19, 20, 21] BLEU (2002), chrF (2015), METEOR (2005)

Creating a Metric for Query Explanations

- It is evident that a robust metric for query explanations should:
  - Take semantic similarity into account, not just common words and n-grams
  - Work well for short text inputs



- Inspiration from learned metrics:
  - Use cosine similarity on sentence embeddings produced by a PLM (e.g., BERTScore)
  - Train a PLM to predict a score on its own (e.g., BLEURT)
- Design a new model using a PLM
  - Take advantage of the SQL query as well

[22, 23] BERTScore (2020), BLEURT (2020)

Creating a SQL-to-Text Dataset

- Currently no dataset/benchmark created specifically for SQL-to-Text
  - All proposed systems use Text-to-SQL datasets such as Spider
- Create a dedicated SQL-to-Text dataset
  - Improve evaluation of systems and comparison of different approaches
  - Higher quality data helps train better systems

An SQL-to-Text benchmark should provide:

- Multiple NL explanations for each SQL
- Variations in style and detail for the NL
- Fine-grained categories for analytical system evaluation
- Realistic DBs and queries that would appear in real life use-cases
- A metric and evaluation script to make scores fair and comparable

What is a Query Explanation?

- There are many different ways to translate a single SQL query to NL
- Different explanations can be required based on the user or the use-case
- There can be different **expression types**:
  - **Statement:** “Employees with a monthly salary higher than 10,000.”
  - **Question:** “Which employees earn a monthly salary higher than 10,000?”
  - **Command:** “Show me all employees with a monthly salary higher than 10,000.”

- Different **levels of detail**:

```
SELECT name, location, district
FROM shop
ORDER BY number_products DESC
```

“Show me the name, location and district of all shops, in descending order of number of products”

“Show me the shops, ordered by their number of products”

- Different ways to verbalise certain **operators**:

```
SELECT *
FROM project
WHERE start_year > 2014
```

“Show projects that started after 2014”

“Show information about projects starting after 2014”

“Show everything about projects that start after 2014”

Data-to-Text

What is Data-to-Text?

Definition: Translating information from a structured form to natural language

Carl Friedrich Gauss	
Born	Johann Carl Friedrich Gauss 30 April 1777 Brunswick, Principality of Brunswick-Wolfenbüttel, Holy Roman Empire
Died	23 February 1855 (aged 77)

Carl Friedrich Gauss was born on the 30th of April 1777 in Brunswick and died on 23th of February 1855 at the age of 77.

Why Data-to-Text?

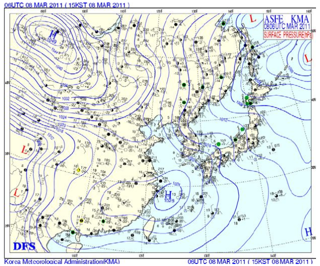
TEAM	WIN	LOSS	PTS	FG_PCT	RB	AS …
Heat	11	12	103	49	47	27
Hawks	7	15	95	43	33	20

PLAYER	AS	RB	PT	FG	FGA	CITY …
Tyler Johnson	5	2	27	8	16	Miami
Dwight Howard	4	17	23	9	11	Atlanta
Paul Millsap	2	9	21	8	12	Atlanta
Goran Dragic	4	2	21	8	17	Miami
Wayne Ellington	2	3	19	7	15	Miami
Dennis Schroder	7	4	17	8	15	Atlanta
Rodney McGruder	5	5	11	3	8	Miami
Thabo Sefolosha	5	5	10	5	11	Atlanta
Kyle Korver	5	3	9	3	9	Atlanta
…						

Box-score statistics of a basketball game

The Atlanta Hawks defeated the Miami Heat , 103 - 95 , at Philips Arena on Wednesday . Atlanta was in desperate need of a win and they were able to take care of a shorthanded Miami team here . Defense was key for

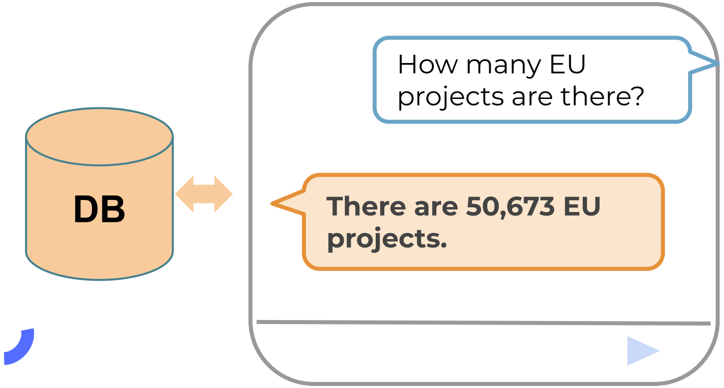
- ➔ Automating and assisting tedious report creation
- ➔ Explanation of data that need expertise to understand
- ➔ Create insights of large amounts of data, not interpretable by a human



Tomorrow expect strong SW winds on the coasts of South Korea

Why Data-to-Text in a Natural Language Database Interface?

Chat-based applications



Quick insight on results

Question: What is the weather in Athens?

Air Temp	Ground Temp	Wind	Humidity	Precipitation
40°C	43°C	2 kts	15%	0%

The weather in Athens is hot with an air temperature of 40°C.

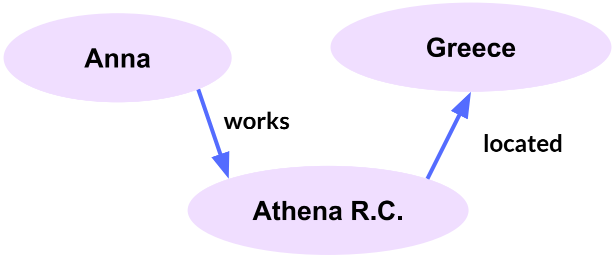
Data-to-Text Sub-fields

Table-to-Text

Name	Age	Height
Anna	26	1.90

“Anna is 26 years old and has a height of 1.90”

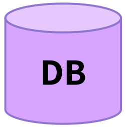
Graph-to-Text



“Anna is employed by Athena R.C. which is located in Greece.”

Query Results-to-Text

What is the average student age?



AVG(age)
21

“The average student age is 21.”

(Relatively less research)

Table-to-Text

Table-to-Text Datasets

Year	Dataset	Domain	Examples
2009	WEATHERGOV	Weather	29,528
2016	WIKIBIO	Wikipedia Bios	728,357
2017	E2E	Restaurants	51,426
2017	ROTOWIRE	Basketball	4,826
2018	ESPN	Basketball	15,054
2018	Wikiperson	Wikipedia Bios	310,655
2019	ROTOWIRE-MODIFIED	Basketball	3,734
2019	MLB	Basketball	26,304
2019	Rotowire-FG	Basketball	7,476
2020	LOGICNLG	Wikipedia	37,015
2020	ToTTo	Wikipedia	136,161
2021	WIKITABLET	Wikipedia	1.5M
2021	SciGen	Scientific	1,300
2021	TWT	Wikipedia	128,268 and 49,417
2022	Hitab	Wikipedia	10,686

Single domain

Cross domain

Influential datasets, which their challenges caused many important innovations in Table-to-Text.

ROTOWIRE – 2017

➔ **Size:** 4.9K statistics-report pairs in total

A dataset of NBA basketball game statistics paired with their human-written reports.

TEAM	WIN	LOSS	PTS	FG_PCT	RB	AST	...
Pacers	4	6	99	42	40	17	...
Celtics	5	4	105	44	47	22	...

PLAYER	H/V	AST	RB	PTS	FG	CITY	...
Jeff Teague	H	4	3	20	4	Indiana	...
Miles Turner	H	1	8	17	6	Indiana	...
Isaiah Thomas	V	5	0	23	4	Boston	...

The **Boston Celtics** defeated the host **Indiana Pacers 105-99** at Bankers Life Fieldhouse on Saturday. In a battle between two injury-riddled teams, the Celtics were able to prevail...

Average number of statistics per game: 628

Average generated text length: 805 words

ToTTo – 2020

➔ **Size:** 135K highlighted tables

Given a Wikipedia table and a set of highlighted table cells, produce a one-sentence description

Table Title: Gabriele Becker
Section Title: International Competitions
Table Description: None

Year	Competition	Venue	Position	Event	Notes
Representing Germany					
1992	World Junior Championships	Seoul, South Korea	10th (semis)	100 m	11.83
1993	European Junior Championships	San Sebastián, Spain	7th	100 m	11.74
			3rd	4x100 m relay	44.60
1994	World Junior Championships	Lisbon, Portugal	12th (semis)	100 m	11.66 (wind: +1.3 m/s)
			2nd	4x100 m relay	44.78
1995	World Championships	Gothenburg, Sweden	7th (q-finals)	100 m	11.54
			3rd	4x100 m relay	43.01

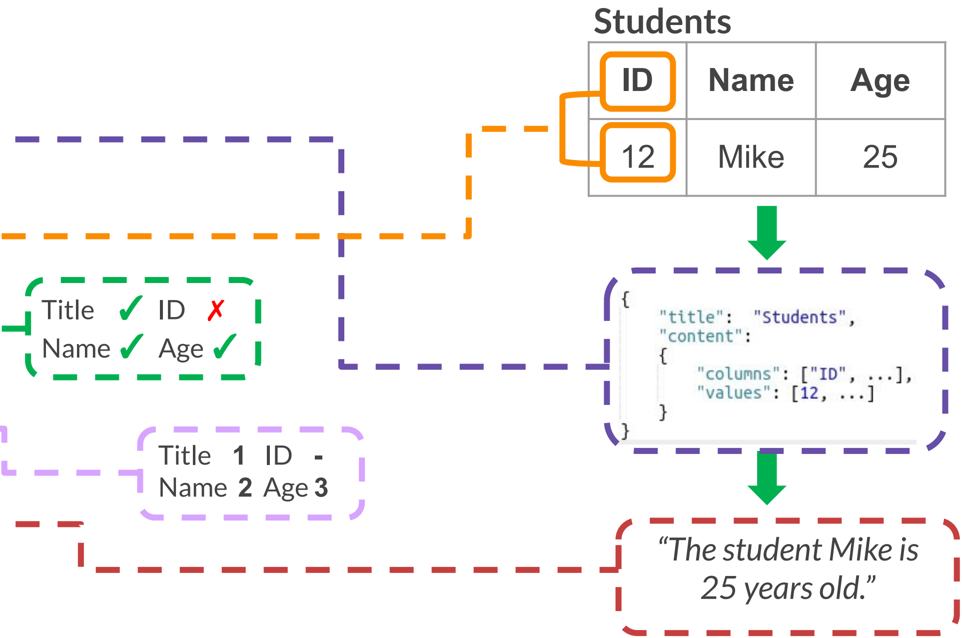
“Gabrielle Becker competed at the 1995 World Championships both individually and on the relay.”

ToTTo is:

- ✓ Big (135K)
- ✓ Diverse
  - Sports
  - Countries
  - Politics
  - …
- ✓ High quality

Challenges of Table-to-Text Systems

- ➔ Table representation
- ➔ Table understanding
- ➔ Content selection
- ➔ Content planning
- ➔ Text generation



Solution Frameworks

Non-pretrained architectures

Based on RNNs, CNNs, transformers, and FCNNs trained from scratch.

- ✅ Flexible and adaptable
- ✅ Low latency
- ❌ No pretraining

Field-Gating Seq-to-Seq (2017) [27]
NCP (2018) [28]
DATA-TRANS (2019) [29]
DUV (2020) [30]

PLMs (i.e. T5, BERT)

Utilising pretrained language models along with other **components**.

- ✅ Language knowledge
- ✅ Only need finetuning
- ❌ Hard to modify

T5 (2019) [31]
TableGPT (2020) [32]
Plan-then-Generate (2021) [33]
LATTICE (2022) [34]
TabT5 (2022) [35]

Large Language Models

Huge models built with human feedback.

- ✅ Great in language-based tasks
- ❌ Not modifiable
- ❌ High cost
- ❌ Privacy

ChatGPT, Bard
Llama 2
Huggingchat
Alpaca

Time

Utilizing PLMs

the current trend

🔗 [36] Text-to-Text Pre-Training for Data-to-Text Tasks (2020)

Representation	Understanding	Selection
XML-like	PLM	PLM

Planning	Generation
PLM	PLM

PLM-only Solution

Table Title: Cristhian Stuani
Section Title: International goals

No.	Date	Venue	Opponent	Result
2	13 November 2013	Amman International Stadium, Amman, Jordan	Jordan	5-0

<page_title> Christian Stuani </page_title>
<section_title> International goals </section_title>
<table>
<cell>
2. <col_header> No. </col_header>
</cell>
...
</table>

PLM

eg. T5, BART, GPT3

"On 13 November 2013 Christian Stuani netted the second in a 5-0 win in Jordan."

Most straightforward PLM solution.

But these are Text-to-Text models, not Table-to-Text.

Structured data

Row/column order invariance

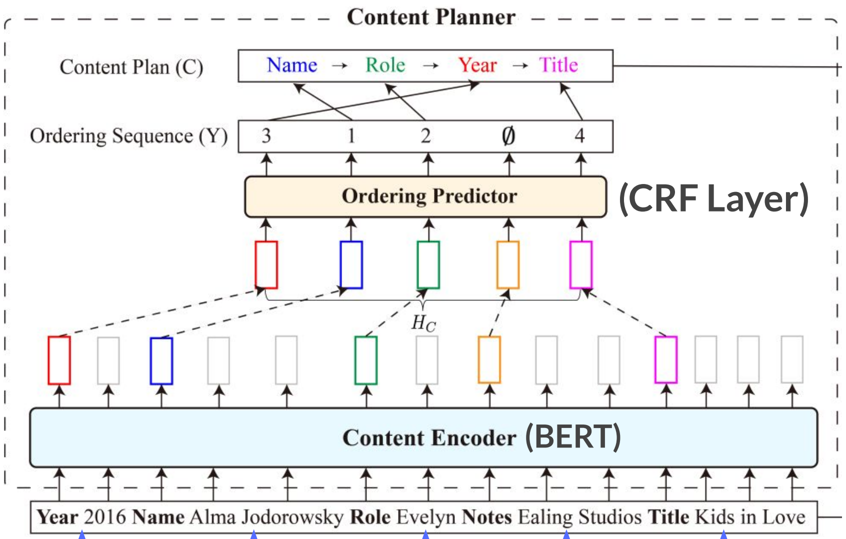
Can we do better?

[33] Plan-then-Generate: Controlled Data-to-Text Generation via Planning (2021)

Representation	Understanding	Selection
Serialised String	PLM	BERT & CRF

Planning	Generation
BERT & CRF	PLM

Plan-then-Generate



Year	Name	Role	Notes	Title
2016	Alma Jodorowsky	Evelyn	Ealing Studios	Kids in Love

BART

To teach the model to remain faithful to the plan they employ RL-training.

[34] Robust (Controlled) Table-to-Text Generation with Structure-Aware Equivariance Learning (2022)

LATTICE

Representation	Understanding	Selection
Serialised String	Pruned Attention Flows	PLM

Planning	Generation
PLM	PLM

Seq-to-seq PLMs (i.e. T5) will **capture the table as a linear structure.**

However tables have content invariant properties

- Shuffling the order of rows.
- Shuffling the order of columns.

ID	Name	Age
9	Anna	26
12	Mike	25

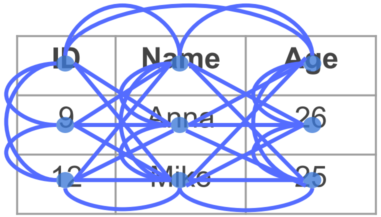
≡

ID	Age	Name
9	26	Anna
12	25	Mike

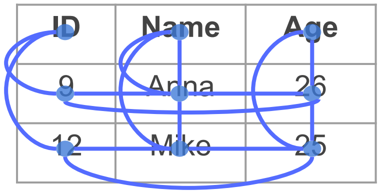
≡

ID	Name	Age
12	Mike	25
9	Anna	26

Modify the transformer encoder module by **pruning not needed attention flows.**



Original Attention Flows



Structural Attention

✓ Taking into account the table structure

✓ Transformation invariance

✓ Applicable to any transformer based architecture

[35] Table-To-Text generation and pre-training with TABT5 (2022)

Representation	Understanding	Selection
Serialised String	Table Pretraining	PLM

Planning	Generation
PLM	PLM

TabT5- Pre-training

Goal: Pre-train T5, a text-to-text model, in understanding table structure.

Datasources

- Wikipedia Infoboxes (3.3M)

Frederick Parker-Rhodes	
Born	21 November 1914 Newington, Yorkshire
Died	2 March 1987 (aged 72)
Residence	UK
Nationality	British

- WikiTable (2.9M)

Year	City	Country	Nations
1896	Athens	Greece	14
1900	Paris	France	24
1904	St. Louis	USA	12

Pre-training tasks

Denoising

Name	Age
X	26
Mike	Y

<X> Anna <Y> 25

ToTTification

Name	Age
Chris	26

Chris is 26 years old.

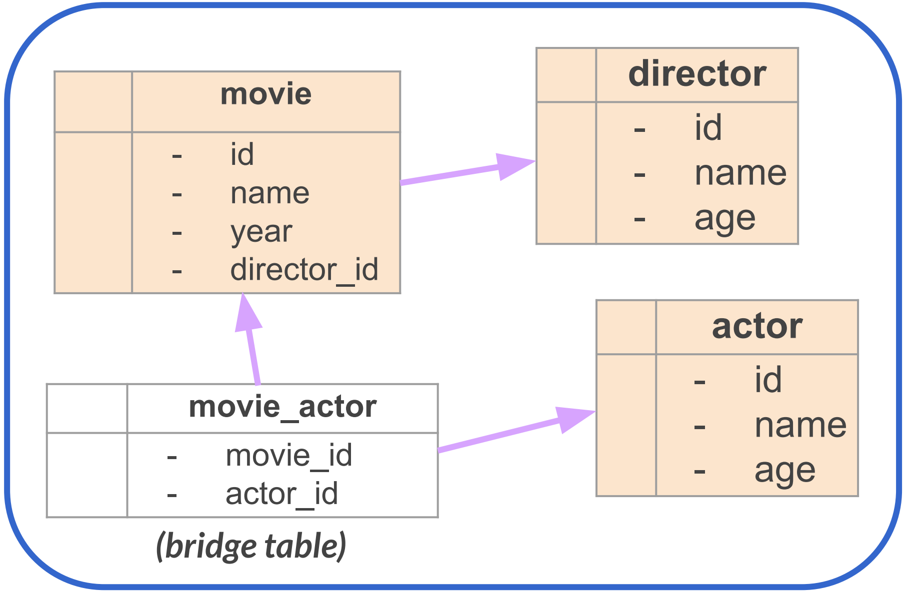
Obtained from **passages** close to the Wikipedia table with **matching entities**.

Similar approaches are followed by **TaBERT** (2020) [42] and **TaPas** (2020) [43]. However, they pretrain an **encoder-only** model (BERT).

Graph-to-Text

Graph-to-Text

Given a graph generate text that expresses the information of the whole graph or parts of it.



*"The database has information about the **movie domain**. For each movie it contains its **name** and **release year** along with the **directors** and **actors** that participated."*

Graph-to-Text Datasets

Year	Dataset	Type/Domain	Examples
2017-20	WebNLG (v3)	DBPedia	16,905
2017-20	LDC2020	Who did what to whom?	59,255
2020	AGENDA	Knowledge Graph	40,720
2020	LOGIC2TEXT	Wikipedia	10,753
2020	WITA	Wikipedia	55,400
2020	GenWiki	DBPedia	1.3mil
2020	ENT-DESC	Knowledge Graphs	110,000
2021	WikiGraphs	Wikipedia	23,522
2021	Map2Seq	OpenStreetMap	7,772
2021	DART	Wikipedia+Restaurant	82,191
2021	EventNarrative	EventKG+Wikidata	224,428

🔗 LDC2020 (2020)

LDC – 2020

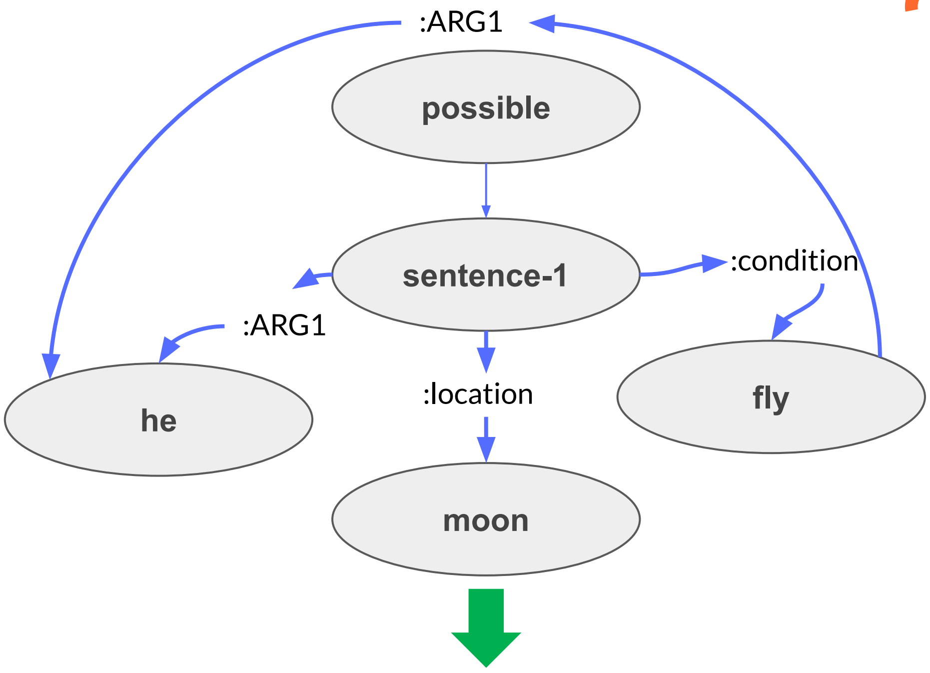
➔ **Size:** 59K triplets

A dataset facilitating the **Abstract Meaning Representation (AMR) to text task.**

AMR captures "who is doing what to whom" in a sentence. Each sentence is paired with a graph that represents its meaning in a tree-structure.

➔ **Data sources:**
- Forum discussions
- Journals
- Blogs
- News texts

Still getting updated every ~3 years

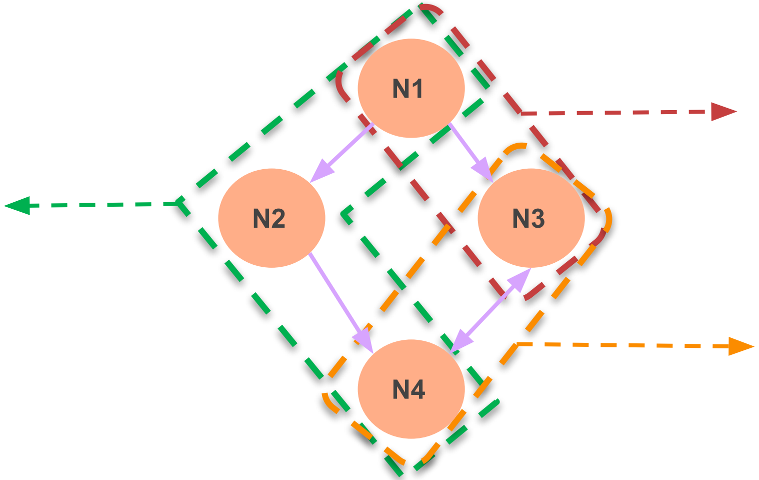


"If he flies he could go to the moon."

Unique Challenge of Graph-to-Text

Encoding the graph structure and the **information we get from it** into a meaningful representation

How do you encode that N1 is connected to N4 through N2?



How do you encode an edge connection between N1 and N3?

How do you encode the bidirectionality of N3 and N4?

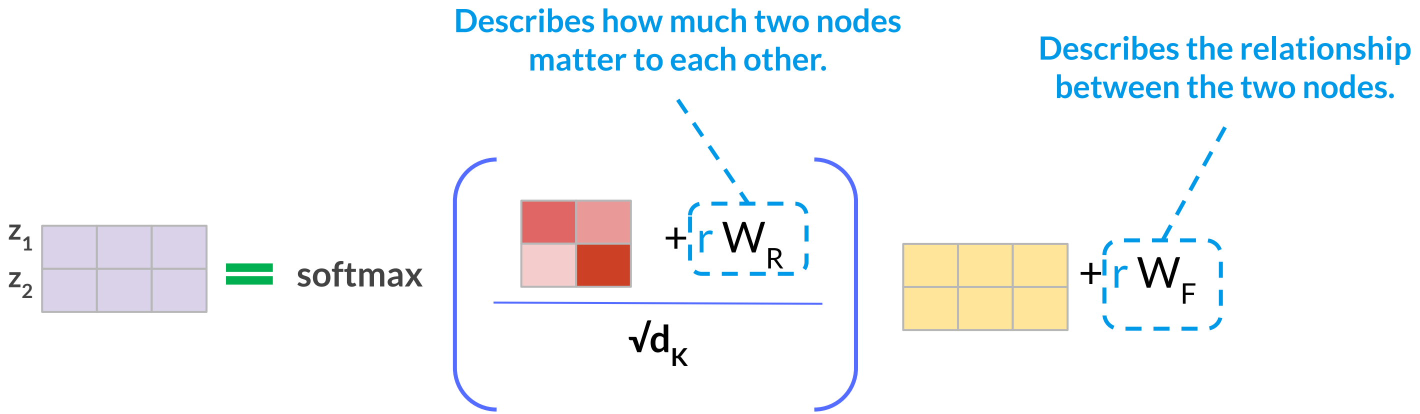
[36] Graph Transformer for Graph-to-Sequence Learning (2019)
[37] Modeling Graph Structure in Transformer for Better AMR-to-Text Generation (2019)

Graph Transformer

Representation	Understanding	Selection
Serialised String	Modified Self-attention	Transformer

Planning	Generation
Transformer	Transformer

The transformer self-attention mechanism was proposed initially for text-to-text problems, meaning that it expects a **sequence of tokens.**



But how can we generate r which describes the relationship between every node combination?

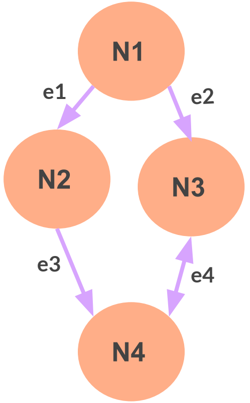
[36] Graph Transformer for Graph-to-Sequence Learning (2019)

[37] Modeling Graph Structure in Transformer for Better AMR-to-Text Generation (2019)

Graph Transformer

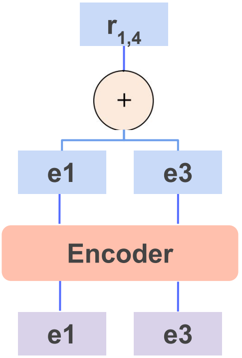
Representation	Understanding	Selection
Serialised String	Modified Self-attention	Transformer

Planning	Generation
Transformer	Transformer



		to			
		N1	N2	N3	N4
from	N1	-	e1	e2	e1,e3
	N2	NO	-	e3, e4	e3
	N3	NO	NO	-	e4
	N4	NO	NO	e4	-

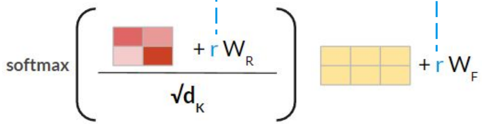
Shortest paths between every node



Encode each path independently to get an embedding

$r_{1,1}$	$r_{1,2}$	$r_{1,3}$	$r_{1,4}$
$r_{2,1}$	$r_{2,2}$	$r_{2,3}$	$r_{2,4}$
$r_{3,1}$	$r_{3,2}$	$r_{3,3}$	$r_{3,4}$
$r_{4,1}$	$r_{4,2}$	$r_{4,3}$	$r_{4,4}$

Our final relationship representation r



There are other methods to encode each path:

- SUM of embeddings
- AVG of embeddings
- 1D Convolution
- RNNs

🔗 [44] Investigating Pretrained Language Models for Graph-to-Text Generation (2020)

Representation	Understanding	Selection
Serialised String	PLM	PLM

Planning	Generation
PLM	PLM

PLM-only Solution

As in Table-to-Text we can simply define a way of serializing our graph to text and then simply feed it to a pretrained PLM.

Input Graph | Serialized Graph | Target Verbalisation

Assumption: The model will be able to catch and embed the relationships that exist between nodes.

Can we help the model to have a better understanding of what a graph is?

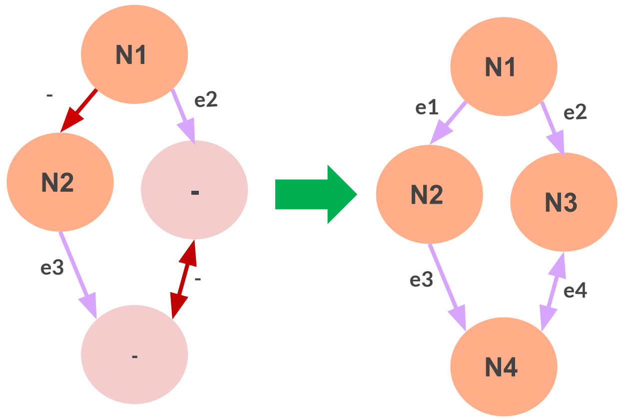
[38] Graph Pre-training for AMR Parsing and Generation (2022)

Representation	Understanding	Selection
Serialised String	Pretraining	PLM

Planning	Generation
PLM	PLM

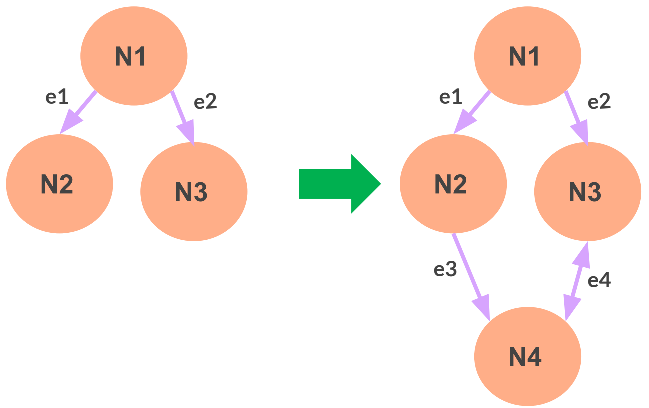
AMRBART - Graph Pre-training

The pretraining tasks aim at improving the graph awareness of PLMs.



Node and edge denoising

Goal: Capturing knowledge about node and edge values.



Sub-graph denoising

Goal: Enforces the model to predict a sub-graph, thus facilitating the graph-level learning.

[19] BLEU (2022) [39] PARENT (2022)

Results

N-gram overlap of different sizes.

Same as BLEU but takes into account the contents of the table.

Model	Representation	Understanding	Selection	Planning	Generation	Dataset	BLEU	PARENT
T5	XML-like	PLM	PLM	PLM	PLM	ToTTo	**47.7**	**57.1**
LATTICE	Serialised String	Pruned Attention Flows	PLM	PLM	PLM	ToTTo	**48.4**	**58.1**
TabT5	Serialised String	Pretraining	PLM	PLM	PLM	ToTTo	**49.2**	**57.2**
Plan-then-Generate	Serialised String	PLM	BERT & CRF	BERT & CRF	PLM	ToTTo	**49.2**	**58.7**
Graph Transformer	Serialised String	Modified Self-attention	Transformer	Transformer	Transformer	LDC2017	**29.8**	-
T5	Serialised String	PLM	PLM	PLM	PLM	LDC2017	**45.8**	-
AMRBART	Serialised String	Pretraining	PLM	PLM	PLM	LDC2017	**49.8**	-

Best solutions utilize PLMs and introduce a way for the model to understand tables.

Huge improvement by using a PLM (T5).

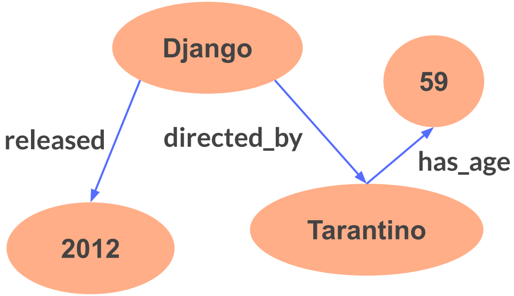
Pretraining for graph understanding achieved significant improvements.

🔗 [45] Large Language Models are Effective Table-to-Text Generators, Evaluators, and Feedback Providers (2023)

LLMs on Data2Text

Air Temp	Ground Temp
40°C	43°C
Wind	**Precipitation**
2 kts	0%

The table shows a substantial 3ºC difference between scorching air at 40º C and even hotter ground at 43ºC, alongside gentle wind (2 kts), and no precipitation (0%).



In 2012, "Django" was released, directed by "Tarantino," who is 59 years old.

Both examples generated with **GPT3.5**

There have been proposals that utilize ChatGPT for structured data:

- StructGPT [40]
- GPT4Graph [41]
- TabLLM [46]

Preliminary benchmarking on the LogicNLG dataset offers a ~15% improvement compared to T5.

Research Opportunities

Purpose may not be clear

In an NLIDB the user has a purpose stated by the NL query.
Eg. *How good is the cheapest laptop?*

Many ways to verbalise a table or graph, especially the bigger they are.

Model	CPU	Display	Price
Laptop 1	i3	17''	1050
Laptop 2	i7	15''	950

Laptop 1 has an i3 CPU, a display of 17'' and is priced at 1050. Laptop 2 has an i7 CPU, a display of 15'' and a price of 950.

Laptop 2 is the cheapest laptop with the fastest CPU but the smallest display.

The most expensive laptop is Laptop 1.

Query Results to Text Challenges

1. Query Result Understanding

A text-to-text model will not understand the structure of a results table.

Name	Job
Tarantino	director

2. Result ambiguities

```
SELECT director.name, movie.name
FROM director INNER JOIN movie
        ON movie.director_id=director.id
WHERE movie.name = 'Dune';
```

name	name
Villeneuve	Dune

3. Incorporating query semantics

Question: How old is Chris?

```
SELECT age
FROM members
WHERE name = 'Chris'
```

age
25

Verbalisation

The age is 25.

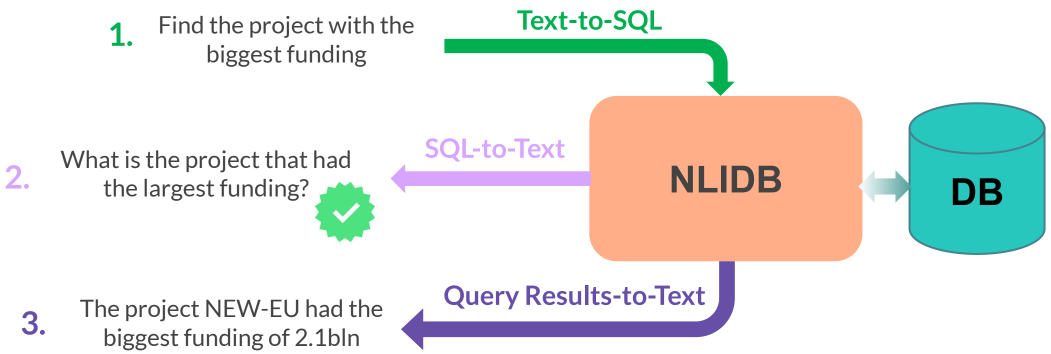
4. Existing Table-to-Text datasets are not suitable

- **Difference in goal**
  Describing a table vs. answering a query
- **No underlying database**
  Great source of information

Bringing it all together

NLIDB – The big challenge

Goal: Combining all these different domains into one system that can be considered a **Natural Language Interface of Databases (NLIDB)**.



The simple solution:

1. Get a well performing model from each field.
2. Train them on their respective datasets.
3. The rest is a technical challenge of how to serve these 3 models.

But many challenges arise.

Challenges: Database Generalizability

For a system to be useful it must be able to work correctly on databases that no training data exist.
Current datasets (eg. Spider) are of high quality but are not able to cover the diversity and difficulties of real-world databases.

Databases are used in every domain from life sciences to e-commerce

Table names and columns might not be "LLM friendly" eg. *usr*

Production databases tend to be much bigger than the ones in existing datasets
(eg. Spider databases have 4 tables on average)

Challenges

Error Propagation

For a user interaction with the NLIDB to be considered successful all 3 components must work correctly.

Performance analysis and evaluation becomes harder since we need a common Benchmark for all the components.

Latency

All of our components' best solutions utilize PLMs (mostly T5).

$T5_{\text{Text-to-SQL}}$ $T5_{\text{SQL-to-Text}}$ $T5_{\text{Data-to-Text}}$

T5 variations (10^7 - 10^9 parameters) have a significant latency when producing inferences. The **latency is 3x** since all the components use T5 or similar PLMs.

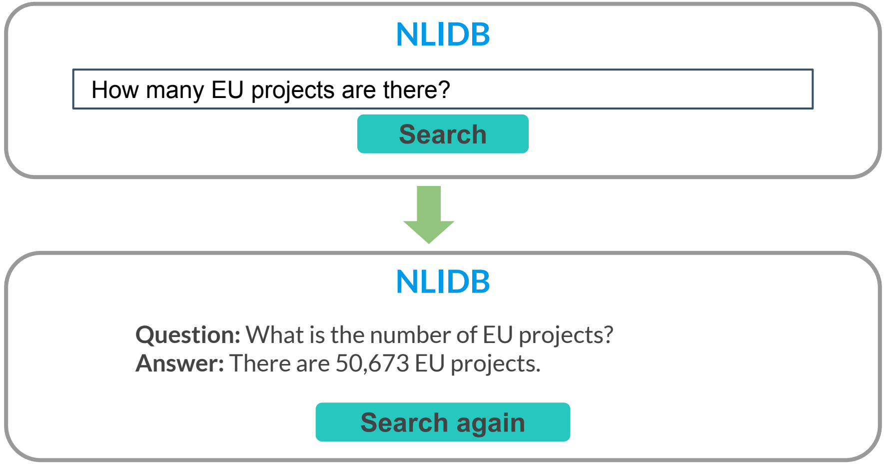
NLIDB solutions must address this issue by focusing on:

- **Efficiency**
- **Model size**

Challenges: User Interface

Designing an intuitive and easy to understand interface, which will not overwhelm the user.

Conversational Approach



"Search Engine" Approach

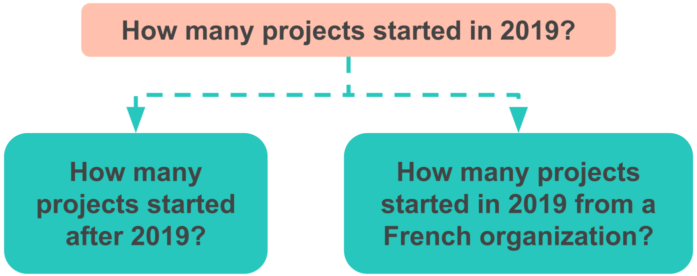
[39] Cluster-Driven Navigation of the Query Space (2016)
[40] Overview of Data Exploration Techniques (2015)

Challenges: Incorporating other fields

In this tutorial we explored the 3 main components of a NLIDB. However, there are more fields and by incorporating them we can improve the user experience.

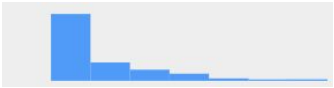
Query Recommendation

Query recommendations aim at aiding the user build a query that cover their needs based on previous interactions or data insights.



Data Exploration

Data exploration addresses query results that have a massive number of rows they are difficult to interpret and end up overwhelming the user.



"In month May of 2021 more projects started compared to month May of the past 10 years."

Median/Avg/Max/Min

Demo

DatAgent

https://bit.ly/datagent

Not mobile friendly 💻

DatAgent is a smart data assistant, developed by the DARELAB team, which works as a NLIDB, integrating solutions for

✓ Text-to-SQL ✓ SQL-to-Text ✓ Data-to-Text

✓ Query Recommendations ✓ Data Exploration

The online version by default runs on the CORDIS database:

A real-world production database used by the European Commission to store information about EU-funded programs such as projects, participants, institutions, etc.

Some example queries that succeed

- Find the number of projects that started in 2015
- Which are the institutions in France?
- Find the projects of the institutions in France *(requires 4 JOINs)*

But still a lot of work needs to be done!

- Which project had the **biggest** cost?
- When is the end date of project with acronym ALFRED?

Noticed something interesting or you have a question?
Feel free to talk to us or contact us "offline".

References (1/8)

[1] E. F. Codd. **Seven Steps to Rendezvous with the Casual User**. 1974. IFIP Working Conference Data Base Management.
[2] V. Hristidis and Y. Papakonstantinou. 2002. **Discover: Keyword Search in Relational Databases**. In VLDB.
[3] V. Hristidis, L. Gravano, and Y. Papakonstantinou. 2003. **Efficient IR-style Keyword Search over Relational Databases**. In VLDB.
[4] Y. Luo, X. Lin, W. Wang, and X. Zhou. 2007. **Spark: Top-k Keyword Query in Relational Databases**. In ACM SIGMOD.
[5] Z. Zeng, M. L. Lee, and T. Wang Ling. 2016. **Answering Keyword Queries involving Aggregates and GROUPBY on Relational Databases**. EDBT.
[6] L. Blunschi, C. Jossen, D. Kossmann, M. Mori, and K. Stockinger. 2012. **SODA: Generating SQL for Business Users**. PVLDB 5, 10 (2012), 932–943.
[7] F. Li and H. V. Jagadish. 2014. **Constructing an Interactive Natural Language Interface for Relational Databases**. PVLDB 8, 1 (Sept. 2014), 73–84.
[8] D. Saha, A. Floratou, K. Sankaranarayanan, U. F. Minhas, A. R. Mittal, and F. Özcan. 2016. **ATHENA: An Ontology-Driven System for Natural Language Querying over Relational Data Stores**. VLDB.
[9] G. Katsogiannis-Meimarakis, G. Koutrika. **A survey on deep learning approaches for text-to-SQL**. The VLDB Journal 32, 905–936 (2023).
[10] H. Li, J. Zhang, C. Li, and H. Chen. 2023. **RESDSQL: Decoupling Schema Linking and Skeleton Parsing for Text-to-SQL**. Proceedings of the AAAI Conference on Artificial Intelligence.

References (2/8)

[11] T. Scholak, N. Schucher, D. Bahdanau. 2021. **PICARD: Parsing Incrementally for Constrained Auto-Regressive Decoding from Language Models**. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing.

[12] L. Jinyang, H. Binyuan, C. Reynold, Q. Bowen, M. Chenhao, H. Nan, H. Fei, D. Wenyu, S. Luo, L. Yongbin. 2023. **Graphix-T5: Mixing Pre-Trained Transformers with Graph-Aware Layers for Text-to-SQL Parsing**.

[13] M. Pourreza, D. Rafiei. 2023. **DIN-SQL: Decomposed In-Context Learning of Text-to-SQL with Self-Correction**

[14] G. Koutrika, A. Simitsis, and Y. E. Ioannidis. 2010. **Explaining structured queries in natural language**. IEEE 26th International Conference on Data Engineering.

[15] D. Guo, Y. Sun, D. Tang, N. Duan, J. Yin, H. Chi, J. Cao, P. Chen, and M. Zhou. 2018. **Question generation from SQL queries improves neural semantic parsing**. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing.

[16] K. Xu, L. Wu, Z. Wang, Y. Feng, and V. Sheinin. 2018. **SQL-to-text generation with graph-to-sequence model**. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing.

[17] D. Ma, X. Chen, R. Cao, Z. Chen, L. Chen, and K. Yu. 2022. **Relation-aware graph transformer for sql-to-text generation**. Applied Sciences.

[18] K. Wu, L. Wang, Z. Li, A. Zhang, X. Xiao, H. Wu, M. Zhang, and H. Wang. 2021. **Data augmentation with hierarchical SQL-to-question generation for cross-domain text-to-SQL parsing**. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing.

[19] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. 2002. **Bleu: a method for automatic evaluation of machine translation**. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics

References (3/8)

[20] M. Popovic. 2015. **chrF: character n-gram F-score for automatic MT evaluation**. In Proceedings of the Tenth Workshop on Statistical Machine Translation.
[21] S. Banerjee and A. Lavie. 2005. **METEOR: An automatic metric for MT evaluation with improved correlation with human judgments**. In Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization
[22] T. Sellam, D. Das, and A. Parikh. 2020. **BLEURT: Learning Robust Metrics for Text Generation**. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.
[23] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. 2020. **BERTScore: Evaluating Text Generation with BERT**. In Eighth International Conference on Learning Representations.
[24] Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. **Challenges in Data-to-Document Generation**. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics.
[25] Craig Thomson, Ehud Reiter, and Somayajulu Sripada. 2020. **SportSett:Basketball - A robust and maintainable data-set for Natural Language Generation.** In Proceedings of the Workshop on Intelligent Information Processing and Natural Language Generation
[26] Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. **ToTTo: A Controlled Table-To-Text Generation Dataset.** In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)

References (4/8)

[27] Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. 2018. **Table-to-text generation by structure-aware seq2seq learning.** In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence (AAAI'18/IAAI'18/EAAI'18)

[28] Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. **Data-to-text generation with content selection and planning.** In Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence (AAAI'19/IAAI'19/EAAI'19)

[29] Li Gong, Josep Crego, and Jean Senellart. 2019. **Enhanced Transformer Model for Data-to-Text Generation.** In Proceedings of the 3rd Workshop on Neural Generation and Translation, pages 148–156, Hong Kong. Association for Computational Linguistics.

[30] Heng Gong, Wei Bi, Xiaocheng Feng, Bing Qin, Xiaojiang Liu, and Ting Liu. 2020. **Enhancing Content Planning for Table-to-Text Generation with Data Understanding and Verification.** In Findings of the Association for Computational Linguistics: EMNLP 2020

[31] Mihir Kale and Abhinav Rastogi. 2020. **Text-to-Text Pre-Training for Data-to-Text Tasks.** In Proceedings of the 13th International Conference on Natural Language Generation, pages 97–102, Dublin, Ireland. Association for Computational Linguistics.

References (5/8)

[32] Heng Gong, Yawei Sun, Xiaocheng Feng, Bing Qin, Wei Bi, Xiaojiang Liu, and Ting Liu. 2020. **TableGPT: Few-shot Table-to-Text Generation with Table Structure Reconstruction and Content Matching.** In Proceedings of the 28th International Conference on Computational Linguistics.

[33] Yixuan Su, David Vandyke, Sihui Wang, Yimai Fang, and Nigel Collier. 2021. **Plan-then-Generate: Controlled Data-to-Text Generation via Planning.** In Findings of the Association for Computational Linguistics: EMNLP 2021

[34] Wang, F., Xu, Z., Szekely, P., & Chen, M. (2022). **Robust (controlled) table-to-text generation with structure-aware equivariance learning.** NAACL22

[35] Andrejczuk, E., Eisenschlos, J. M., Piccinno, F., Krichene, S., & Altun, Y. (2022). **Table-to-text generation and pre-training with tabt5.** EMNLP22.

[36] Li Gong, Josep Crego, and Jean Senellart. 2019. **Enhanced Transformer Model for Data-to-Text Generation.** In Proceedings of the 3rd Workshop on Neural Generation and Translation. Association for Computational Linguistics.

[37] Jie Zhu, Junhui Li, Muhua Zhu, Longhua Qian, Min Zhang, and Guodong Zhou. 2019. **Modeling Graph Structure in Transformer for Better AMR-to-Text Generation.** In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)

References (6/8)

[38] Xuefeng Bai, Yulong Chen, and Yue Zhang. 2022. **Graph Pre-training for AMR Parsing and Generation.** In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics.

[39] Dhingra, Bhuwan, Manaal Faruqui, Ankur Parikh, Ming-Wei Chang, Dipanjan Das, and William W. Cohen. **Handling divergent reference texts when evaluating table-to-text generation.** arXiv preprint arXiv:1906.01081 (2019).

[40] Jiang, J., Zhou, K., Dong, Z., Ye, K., Zhao, W. X., & Wen, J. R. (2023). **Structgpt: A general framework for large language model to reason over structured data.** arXiv preprint arXiv:2305.09645.

[41] Guo, J., Du, L., & Liu, H. (2023). **GPT4Graph: Can Large Language Models Understand Graph Structured Data?** An Empirical Evaluation and Benchmarking. arXiv preprint arXiv:2305.15066.

[42] Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. **TaBERT: Pretraining for Joint Understanding of Textual and Tabular Data.** In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics.

[43] Herzig, J., Nowak, P. K., Müller, T., Piccinno, F., & Eisenschlos, J. M. (2020). **TaPas: Weakly supervised table parsing via pre-training.** arXiv preprint arXiv:2004.02349.

[44] Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2021. **Investigating Pretrained Language Models for Graph-to-Text Generation.** In Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI, pages 211–227, Online. Association for Computational Linguistics.

References (7/8)

[38] Xuefeng Bai, Yulong Chen, and Yue Zhang. 2022. **Graph Pre-training for AMR Parsing and Generation.** In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics.
[39] Dhingra, Bhuwan, Manaal Faruqui, Ankur Parikh, Ming-Wei Chang, Dipanjan Das, and William W. Cohen. **Handling divergent reference texts when evaluating table-to-text generation.** arXiv preprint arXiv:1906.01081 (2019).
[40] Jiang, J., Zhou, K., Dong, Z., Ye, K., Zhao, W. X., & Wen, J. R. (2023). **Structgpt: A general framework for large language model to reason over structured data.** arXiv preprint arXiv:2305.09645.
[41] Guo, J., Du, L., & Liu, H. (2023). **GPT4Graph: Can Large Language Models Understand Graph Structured Data?** An Empirical Evaluation and Benchmarking. arXiv preprint arXiv:2305.15066.
[42] Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. **TaBERT: Pretraining for Joint Understanding of Textual and Tabular Data.** In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics.
[43] Herzig, J., Nowak, P. K., Müller, T., Piccinno, F., & Eisenschlos, J. M. (2020). **TaPas: Weakly supervised table parsing via pre-training.** arXiv preprint arXiv:2004.02349.
[44] Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2021. **Investigating Pretrained Language Models for Graph-to-Text Generation.** In Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI, pages 211–227, Online. Association for Computational Linguistics.

References (8/8)

[45] Zhao, Y., Zhang, H., Si, S., Nan, L., Tang, X., & Cohan, A. (2023). **Large Language Models are Effective Table-to-Text Generators, Evaluators, and Feedback Providers**. arXiv preprint arXiv:2305.14987.
[46] Hegselmann, Stefan, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. **Tabllm: Few-shot classification of tabular data with large language models.** In International Conference on Artificial Intelligence and Statistics, PMLR, 2023.