Natural Language Interfaces for Databases with Deep Learning

George Katsogiannis-Meimarakis
Athena Research Center
Athens, Greece
katso@athenarc.gr

Mike Xydas
Athena Research Center
Athens, Greece
mxydas@athenarc.gr

Georgia Koutrika
Athena Research Center
Athens, Greece
georgia@athenarc.gr

ABSTRACT

In the age of the Digital Revolution, almost all human activities, from industrial and business operations to medical and academic research, are reliant on the constant integration and utilisation of ever-increasing volumes of data. However, the explosive volume and complexity of data makes data querying and exploration challenging even for experts, and makes the need to democratise the access to data, even for non-technical users, all the more evident. It is time to lift all technical barriers, by empowering users to access relational databases through conversation. We consider 3 main research areas that a natural language data interface is based on: Text-to-SQL, SQL-to-Text, and Data-to-Text. The purpose of this tutorial is a deep dive into these areas, covering state-of-the-art techniques and models, and explaining how the progress in the deep learning field has led to impressive advancements. We will present benchmarks that sparked research and competition, and discuss open problems and research opportunities with one of the most important challenges being the integration of these 3 research areas into one conversational system.

PVLDB Reference Format:
George Katsogiannis-Meimarakis, Mike Xydas, and Georgia Koutrika. Natural Language Interfaces for Databases with Deep Learning. PVLDB, 16(12): 3878 - 3881, 2023.
doi:10.14778/3611540.3611575

1 INTRODUCTION

The Web has democratized access to knowledge, and search engines have arguably played a paramount role in this by enabling users to search for information in web pages using keywords or simple natural language questions. However, a vast amount of data used in a wide range of tasks, from business operations, medical and scientific research, to activities in our everyday lives, lives in relational databases. To a great extent, this creates a technical barrier for users not familiar with a low-level query language such as SQL for formulating their queries and holds them back from leveraging the wealth of data. During the past decades, there has been an increasing research focus on data democratisation to lift this barrier by allowing users to query data using natural language.

Proceedings of the VLDB Endowment, Vol. 16, No. 12 ISSN 2150-8097.
doi:10.14778/3611540.3611575

To motivate the importance of a natural language data interface imagine a scientist retrieving information from a relational database. She first asks an NL question, which is translated to a SQL query (Text-to-SQL). How can the user confirm that the results they receive correctly matches their intention? An explanation of the SQL query may be returned to the user who can examine whether the system understood her NL question correctly (SQL-to-Text). Then, the user receives the query results in natural language (Data-to-Text), which is quickly and easily readable.

The recent advances in deep neural networks along with the creation of large and diverse datasets have led to an explosion of recent efforts, shaping some of the most exciting and fast-paced research fields, and making data democratisation seem more real than ever. While research on Text-to-SQL, SQL-to-Text and Data-to-Text systems thrives, many open challenges stand in our way. To understand the progress in the field, the barriers, and the opportunities for breaking these barriers, a systematic study of existing solutions is needed. This is critical as we are moving to conversational systems over the Web, where the user and the system can interact using natural language. Hence, this tutorial is important and timely.

Given the abundance of existing deep learning approaches, we organize them in a detailed taxonomy and highlight their differences and commonalities as well as their advantages and deficiencies. We will zoom in on the recent advances of deep learning techniques for each one of the three tasks, as well as the available evaluation metrics and techniques. Our analysis will highlight new research opportunities for researchers and practitioners, aiming at building a strong foundation for research on natural language data interfaces going forward, and it will discuss the open challenges for the broader area of conversational search and recommender systems.

2 TUTORIAL OUTLINE

2.1 The Text-to-SQL Problem

We will first introduce the problem at hand, present its main challenges, and analyze their impact on a Text-to-SQL system.

A Text-to-SQL system translates a Natural Language Query (NLQ) over a relational database (RDB) to an equivalent SQL query, which is valid for this RDB. In this way, the user is shielded from the particularities of the database, and can, in principle, ask any queries over the data using natural language.

The Text-to-SQL problem hides several challenges. One of the most important ones is the inherent *ambiguity* of Natural Language Queries (NLQs). For instance, *lexical ambiguity*, where a single word has multiple meanings (e.g., *"Paris"* can be a city or a person), and *syntactic ambiguity*, where a sentence has multiple interpretations (e.g., *"Find all German movie directors"* can mean *"directors that have*

directed German movies" or *"directors from Germany"*). On the other hand, several challenges arise from the DB and SQL part of the problem. A system must be robust to the *vocabulary gap* problem, where the DB might use different vocabulary than the one used by the user (e.g., a user might ask for "singers", but the DB might store them as "artists"). Furthermore, some user questions may require the system to generate complex SQL queries (e.g., the NLQ "Find the highest rated movie" might require a nested SQL query).

Two main datasets have enabled the bloom in research for this task during the past years. On the one hand, WikiSQL [19] contains 25,000 tables from Wikipedia pages and over 80,000 natural language and SQL question pairs. WikiSQL questions are simple since they are directed to a single table and not to a relational database. Hence the proposed task is simplified. On the other hand, the Spider dataset [18] contains 200 relational databases from 138 different domains along with over 10,000 natural language questions and over 5,000 complex SQL queries. Its queries cover a wide range of complexity, from very simple to very hard, using all the common SQL elements and nesting.

Given the abundance of existing deep learning approaches for the Text-to-SQL problem, we will present a fine-grained taxonomy of these systems, and highlight the main characteristics as well as the advantages and shortcomings of each approach. Below, we highlight some important dimensions of this taxonomy.

- *Schema Linking*: The process of discovering the connections between words in the NLQ and the DB elements (tables, columns, values) they refer to. This first step produces important information that can help the system create the correct SQL query.
- *Input Encoding*: This dimension examines how the various inputs to the task (e.g., NLQ, DB schema) are processed so that they can be effectively used by the neural part of the system.
- *Decoder Output*: This dimension refers to the different approaches that a neural network can use to generate a SQL query.
- *Neural Training*: Besides training a network from scratch, there are novel approaches that show better performance in some cases, such as the incorporation of transfer learning, the use of auxiliary training tasks, or pre-training specific components of the network before starting the standard part of the training.
- *Output Refinement*: Having trained a neural model, there are additional techniques to ensure that the system avoid producing error-yielding SQL queries. Such techniques are mostly based on designing rules that can restrict the model's output space, or executing queries while they are being constructed, to ensure their correctness.

Based on the presented taxonomy, we will study important Text-to-SQL systems in greater depth to offer a concrete understanding of the different approaches proposed. Seq2SQL [19] and SQLNet [17] were the first neural networks specifically created for the Text-to-SQL problem. The emergence of pre-trained language models (PLM) such as BERT [2] changed the landscape. We will present systems such as SQLova [4] and HydraNet [9] that heavily rely on BERT. We will also focus on complex systems such as RAT-SQL [14], and we will analyse how they generate complex SQL queries such as the ones in the Spider benchmark. Finally, we will present the latest innovations in the area, such as the PICARD [13] decoding technique, that has allowed the use of seq-to-seq PLM to achieve the highest score on Spider, overturning previous beliefs that seq-to-seq models are not adequate for the Text-to-SQL task. We will take advantage of the taxonomy to highlight the differences and commonalities between these systems, as well as the best design choices based on the requirements for a Text-to-SQL system.

2.2 The SQL-to-Text Problem

The SQL-to-Text problem hides several challenges. First and foremost, such a system should generate fluent and human-like explanations of SQL queries. Another challenge is correctly identifying the DB domain and using the appropriate vocabulary. For example, the `MAX` aggregation function must be translated in a different way, depending on the context and the attribute on which it is applied. In a DB containing sport data, the `MAX(lap_time)` refers to the *"slowest lap time"*, while in a database containing products, the `MAX(price)` refers to the *"highest product price"*. Finally, the complexity of SQL poses additional challenges in tackling this problem. As we discussed previously, simple NLQs might map to complex SQL queries. Similarly, a SQL-to-Text system must be able to understand the underlying meaning of complex queries in order to be able to express them with a simple,condensed NL explanation.

In contrast to Text-to-SQL, this inverse problem has seen relatively little attention from the research community. More specifically, template-based [6] can produce very accurate explanations of SQL queries, but require a lot of manual effort in order to create new templates for a new DB that the system must work on. The biggest caveat of template-based systems is that they often generate "robotic" and unnatural explanations. A simple example can be seen in the following SQL query: `SELECT p.title FROM projects p WHERE p.start_year>=2014 AND p.start_year<=2018`, which is translated to *"Find projects whose start year is greater than or equal to 2014 and start year is less than or equal to 2018."* by a template-based system, but could be explained much more fluently as *"Get the names of projects started between 2014 and 2018."*

Only a handful of solutions using deep neural networks already exist. Deep learning solutions (e.g., [16]) offer better generalisation to unseen databases, but are not guaranteed to generate accurate explanations every time. We will discuss the existing solutions for this problem, compare their advantages and drawbacks, while also paving the path for future research, by addressing the opportunities that arise from the use of novel NLP techniques that have taken other research areas by storm, such as the the Transformer architecture and Pre-trained Language Models (PLMs).

2.3 The Data-to-Text Problem

We will introduce the Data-to-Text problem and we will showcase the connection with our goal of a natural language data interface. Data-to-Text aims at generating fluent and fact-based verbalisations of a given structured input. The problem requires careful encoding of the input allowing the model to understand the underlying input structure. Also, the task of generating the verbalisation (decoding) has its intricacies since our input tends to have many entities and unseen content. Data-to-Text can be distinguished based on the type of input into: Table-to-Text and Graph-to-Text.

On Table-to-Text, the first datasets were fairly small and domain-specific. Wikibio [7] gathers 728,321 biography info-boxes from English Wikipedia along with the first paragraph of the page. ROTOWIRE [15] consists of (human-written) 4,853 NBA basketball game summaries aligned with their corresponding box- and line-scores. More recently, a large and domain-diverse dataset was released named ToTTo [11], which proposes a controlled generation task: given a Wikipedia table and a set of highlighted table cells, produce a one-sentence description.

In Graph-to-Text, one of the most influential datasets is the WebNLG [3] corpus, which comprises sets of triplets describing facts in the form of a graph and natural language.

2.3.1 Table-to-Text Systems. In Table-to-Text, the input of the model is a table and the goal is to output its description in natural language focusing both on coverage (all/important parts of the table are described) and fluency (avoiding syntactical errors).

Field-gating Seq2Seq [8] focused on making the model have both a global and local view of the verbalised table simultaneously. Using the global view, the model will decide the order and contents of the verbalisation, while using the local view, it will choose words to copy or paraphrase.

On the other hand, NCP [12] prefers a two-staged approach. First, a content plan is created, which decides the order in which the records of the table should be generated. Second, an LSTM network takes into account the encoded plan along with a copy mechanism, and it generates its verbalisation. This stage separation allows for intermediate rewards leading to more stable training.

As in the other two fields of Text-to-SQL and SQL-to-Text, the pre-training of huge language models has greatly impacted both Table-to-Text and Graph-to-Text. So far, the solutions focus on straightforward application of models like GPT-2 or T5, managing to outperform previous approaches. However, these are just the first steps in using pre-trained models on Data-to-Text and a lot of research is needed to successfully harness their full power.

2.3.2 Graph-to-Text Systems. In Graph-to-Text, we have as input a graph but the goal remains the same, i.e., to generate a text description of the contents of the graph. The main challenge is a meaningful representation of the relations between nodes which are crucial for correct verbalisation.

In a similar fashion as in schema encoding of RAT-SQL, Zhu et al. [20] extend the attention mechanism to include information about the relationships between two nodes that are not necessarily directly connected. A similar approach is followed by Cai et al. [1] but they use the transformer architecture for generating the path and a bi-directional GRU for encoding the information of the path.

Most recent approaches focus on leveraging the power of self-training with the teacher-student architecture. In this direction, the main challenge is processing and filtering the synthetic labels created. CSBT [5] chooses to train the student model on a pseudo-labeled dataset of increasing curriculum difficulty. BLEURT self-training [10] leverages the BLEURT metric for filtering low-quality synthetic labels.

2.4 Challenges and Research Opportunities

We will present challenges that are unique to each of the discussed problems, areas where one problem could benefit by the recent advances in the other domains, as well as challenges of integrating all three tasks into a single conversational system.

Firstly, evaluation is a big hurdle for all three problems. In some cases there are no perfect automatic metrics, and systems often resort to human evaluation. Another common problem is that evaluation is often limited to accuracy metrics, overlooking time and computational costs. There is a constant need for new benchmarks that can stress these systems and test them in difficulties they would encounter when working on a real database. For example, how to handle domain specific knowledge, large amounts of data, many users that interact simultaneously, and so forth.

Another challenge that troubles all three areas is how structured data (e.g., tables and databases) can be efficiently represented in a format that can be processed by a neural network.

Also, generalizing the problem beyond relational databases is another domain that will also enjoy the attention of researchers in the near future, given that the advancements of Knowledge Graphs, the Resource Description Framework (RDF) and query languages such as SPARQL, point to the need for similar interfaces that can go beyond SQL.

One of the most interesting research and engineering challenges is to create a unified system that combines solutions to the problems we presented, creating a complete natural language data interface. Simply combining existing models is destined to fail, because most systems proposed in all three domains are not designed and tested for real-world databases. Additionally, since this will be a system for casual users, latency, usability, and accessibility, all become important factors, that require specific optimizations and evaluation studies in order to achieve enjoyable user experience. However, the feat of implementing such a system will be a massive step forward for data democratisation, and a remarkable scientific and engineering achievement.

3 PRESENTERS

George Katsogiannis-Meimarakis is a research assistant at Athena Research Center in Greece, and a graduate of the Department of Informatics and Telecommunications of the National and Kapodistrian University of Athens. He holds a MSc of Data Science and will be starting a PhD in September of 2023. *Prior tutorials:* Deep Learning for Text-to-SQL [EDBT'21, SIGMOD'21, TWC'22, WSDM'23].

Mike Xydas is a research assistant at Athena Research Center in Greece, where he works on the EOSC Future project on creating a recommender system for the EOSC portal and the INODE project focusing on the Data2Text problem. He is a graduate of the Department of Informatics and Telecommunications and is currently attending the MSc programme on Data Science and Information Technologies with a specialisation on Artificial Intelligence and Big Data, completing his thesis with title "QR2T: Verbalising Query Results". *Prior tutorials:* Deep Learning for Data Democratisation [WSDM'23].

Georgia Koutrika is a Research Director at Athena Research Center in Greece. She has worked in multiple roles at HP Labs, IBM

Almaden, and Stanford. Her work focuses on intelligent and interactive data exploration, conversational data systems, and user-driven data management, and it has been incorporated in commercial products, described in 14 granted patents and 26 patent applications in the US and worldwide, and published in more than 100 papers in top-tier conferences and journals. She is a member of the VLDB Endowment Board of Trustees, member of the PVLDB Advisory Board, member of the ACM-RAISE Working Group, co-Editor-in-chief for VLDB Journal, PC co-chair for VLDB 2023, co-EiC of Proceedings of VLDB (PVLDB). She has been associate editor in top-tier conferences (such as ACM SIGMOD, VLDB) and journals (VLDB Journal, IEEE TKDE), and she has been in the organizing committee of several conferences including SIGMOD, ICDE, EDBT, among others. *Prior tutorials:* She has given several tutorials in conferencs and summer schools, including: Deep Learning for Text-to-SQL [EDBT'21, SIGMOD'21, TWC'22, WSDM'23], Fairness in Rankings and Recommenders [EDBT20, MDM21], Recommender Systems [SIGMOD'18, EDBT'18, ICDE'15], Personalization [ICDE'10, ICDE'07, VLDB'05].

ACKNOWLEDGMENTS

This work has been partially funded by the European Union's Horizon 2020 research and innovation program (grant agreement No 863410) and the European Union Horizon Programme call INFRAEOSC-03-2020 (grant agreement No 101017536).

REFERENCES

[1] Deng Cai and Wai Lam. 2020. Graph Transformer for Graph-to-Sequence Learning. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 05 (Apr. 2020), 7464–7471. https://doi.org/10.1609/aaai.v34i05.6243

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs.CL]

[3] Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for nlg micro-planning. In *55th annual meeting of the Association for Computational Linguistics (ACL)*.

[4] Wonseok Hwang, Jinyeong Yim, Seunghyun Park, and Minjoon Seo. 2019. A Comprehensive Exploration on WikiSQL with Table-Aware Word Contextualization. arXiv:1902.01069 [cs.CL]

[5] Pei Ke, Haozhe Ji, Zhenyu Yang, Yi Huang, Junlan Feng, Xiaoyan Zhu, and Minlie Huang. 2022. Curriculum-Based Self-Training Makes Better Few-Shot Learners for Data-to-Text Generation. *arXiv:2206.02712* (2022).

[6] Andreas Kokkalis, Panagiotis Vagenas, Alexandros Zervakis, Alkis Simitsis, Georgia Koutrika, and Yannis Ioannidis. 2012. Logos: A System for Translating Queries into Narratives. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data* (Scottsdale, Arizona, USA) *(SIGMOD '12)*. Association for Computing Machinery, New York, NY, USA, 673–676. https://doi.org/10.1145/2213836.2213929

[7] Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. *arXiv:1603.07771* (2016).

[8] Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. 2018. Table-to-text generation by structure-aware seq2seq learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[9] Qin Lyu, Kaushik Chakrabarti, Shobhit Hathi, Souvik Kundu, Jianwen Zhang, and Zheng Chen. 2020. Hybrid Ranking Network for Text-to-SQL. arXiv:2008.04759 [cs.CL]

[10] Sanket Vaibhav Mehta, Jinfeng Rao, Yi Tay, Mihir Kale, Ankur Parikh, Hongtao Zhong, and Emma Strubell. 2021. Improving Compositional Generalization with Self-Training for Data-to-Text Generation. *arXiv:2110.08467* (2021).

[11] Ankur P Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. ToTTo: A controlled table-to-text generation dataset. *arXiv:2004.14373* (2020).

[12] Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with content selection and planning. In *Proceedings of the AAAI conference on artificial intelligence*.

[13] Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. 2021. PICARD: Parsing Incrementally for Constrained Auto-Regressive Decoding from Language Models. arXiv:2109.05093 [cs.CL]

[14] Bailin Wang, Richard Shin, Xiaodong Liu, Oleksandr Polozov, and Matthew Richardson. 2020. RAT-SQL: Relation-Aware Schema Encoding and Linking for Text-to-SQL Parsers. arXiv:1911.04942 [cs.CL]

[15] Sam Wiseman, Stuart M Shieber, and Alexander M Rush. 2017. Challenges in data-to-document generation. *arXiv:1707.08052* (2017).

[16] Kun Xu, Lingfei Wu, Zhiguo Wang, Yansong Feng, and Vadim Sheinin. 2018. SQL-to-Text Generation with Graph-to-Sequence Model. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium. https://doi.org/10.18653/v1/D18-1112

[17] Xiaojun Xu, Chang Liu, and Dawn Song. 2017. SQLNet: Generating Structured Queries From Natural Language Without Reinforcement Learning. arXiv:1711.04436 [cs.CL]

[18] Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2019. Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task. arXiv:1809.08887 [cs.CL]

[19] Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning. arXiv:1709.00103 [cs.CL]

[20] 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. *arXiv:1909.00136* (2019).