**Qual o objetivo do meu trabalho?**

O objetivo do meu trabalho é…

**Natural language interfaces to databases–an introduction (1995)**

This paper is an introduction to natural language interfaces to databases (Nlidbs). A brief overview of the history of Nlidbs is first given. Some advantages and disadvantages of Nlidbs are then discussed, comparing Nlidbs to formal query languages, form-based interfaces, and graphical interfaces. An introduction to some of the linguistic problems Nlidbs have to confront follows, for the benefit of readers less familiar with computational linguistics. The discussion then moves on to Nlidb architectures, portability issues, restricted natural language input systems (including menu-basedNlidbs),andNlidbs with reasoning capabilities. Some less explored areas of Nlidb research are then presented, namely database updates, meta-knowledge questions, temporal questions, and multi-modal Nlidbs. The paper ends with reflections on the current state of the art.

**Natural Language Interfaces to Databases: An Analysis of the State of the Art (2013) - NOVO**

https://link.springer.com/chapter/10.1007/978-3-642-33021-6\_36

([Recent Advances on Hybrid Intelligent Systems](https://link.springer.com/book/10.1007/978-3-642-33021-6) pp 463-480)

Although since the 60s many NLIDBs have been developed, their performance has not been satisfactory, there still remain very difficult problems that have not been solved by NLIDB technology, and there does not yet exist a standardized method of evaluation that permits to compare the performance of different NLIDBs. This chapter presents an analysis of NLIDBs, which includes their classification, techniques, advantages, disadvantages, and a proposal for a proper evaluation of them.

**Natural language interfaces to databases (dissertação de mestrado) (2006) - NOVO**

Y Chandra, R Mihalcea - Masters Abstracts International, 2006 - Citeseer

Natural language interfaces to databases (NLIDB) are systems that aim to bridge the gap between the languages used by humans and computers, and automatically translate natural language sentences to database queries. This thesis proposes a novel approach to NLIDB, using graph-based models. The system starts by collecting as much information as possible from existing databases and sentences, and transforms this information into a knowledge base for the system. Given a new question, the system will use this knowledge to analyze and translate the sentence into its corresponding database query statement. The graph-based NLIDB system uses English as the natural language, a relational database model, and SQL as the formal query language. In experiments performed with natural language questions ran against a large database containing information about U.S. geography, the system showed good performance compared to the state-of-the-art in the field.

**Towards a Theory of Natural Language Interfaces to Databases (2003)**

In this paper, we introduce a theoretical framework for reliable NLIs, which is the foundation for the fully implemented PRECISE NLI. We prove that, for a broad class of *semantically tractable* natural language questions, PRECISE is guaranteed to map each question to the corresponding SQL query. We report on experiments testing PRECISE on several hundred questions drawn from user studies over three benchmark databases. We find that over 80% of the questions are semantically tractable questions, which PRECISE answers correctly. PRECISE automatically recognizes the 20% of questions that it cannot handle, and requests a paraphrase. Finally, we show that PRECISE compares favorably with Mooney’s learning NLI and with Microsoft’s English Query product.

**Modern natural language interfaces to databases: Composing statistical parsing with semantic tractability (2004)**

Natural Language Interfaces to Databases (NLIs) can benefit from the advances in statistical parsing over the last fifteen years or so.However, statistical parsers require training on a massive, labeled corpus, and manually creating such a corpus for each database is prohibitively expensive. To address this quandary, this paper reports on the PRECISE NLI, which uses a statistical parser as a “plugin”. The paper shows how a strong semantic model coupled with “light retraining” enables PRECISE to overcome parser errors, and correctly map from parsed questions to the corresponding SQL queries. We discuss the issues in using statistical parsers to build database-independent NLIs, and report on experimental results with the benchmark ATIS dataset where PRECISE achieves 94% accuracy.

**A survey on question answering technology from an information retrieval perspective (2011)**

This article provides a comprehensive and comparative overview of question answering technology. It presents the question-answering task from an information retrieval perspective and emphasizes the importance of retrieval models, i.e., representations of queries and information documents, and retrieval functions which are used for estimating the relevance between a query and an answer candidate. The survey suggests a general question answering architecture that steadily increases the complexity of the representation level of questions and information objects. On the one hand, natural language queries are reduced to keyword-based searches, on the other hand, knowledge bases are queried with structured or logical queries obtained from the natural language questions, and answers are obtained through reasoning. We discuss different levels of processing yielding bag-of-words-based and more complex representations integrating part-of-speech tags, classification of the expected answer type, semantic roles, discourse analysis, translation into a SQL-like language and logical representations.

3. A short history of question answering

In the literature the first cited question answering systems are BASEBALL [42], built in 1961, and LUNAR [149], built in 1972, both of which interrogated a structured database using natural language questions. LUNAR provided an interface to data from analyses of rock samples during the Apollo moon missions. The BASEBALL system answered questions about baseball games played over a period of one year. Both systems analyzed questions based on a set of natural language patterns that were expected to occur in the input. Since the subject domain was restricted, an exhaustive set of analysis patterns embedded in a domain-specific vocabulary was built manually, so that the questions could be processed and successfully translated into a structured query form needed to interrogate the databases. BASEBALL and LUNAR were the first examples of natural language interfaces to databases, NLIDB. Today’s research in NLIDB focuses on generic approaches to the detection of objects and their attributes and relationships in natural language questions, and on the translation of lexical items into the string tokens that are used to describe the database entries. An overview of NLIDB research can be found in [23,8]. Recent work in this area includes [118,103,95].

[118] Vinitha Reddy, Kyle Neumeier, Joshua McFarlane, Jackson Cothren, Craig W. Thompson, Extending a natural language interface with geospatial queries, IEEE Internet Computing 11 (2007) 82–85.

[103] Majdi Owda, Zuhair Bandar, Keeley Crockett, Conversation-based natural language interface to relational databases, in: Proceedings of IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology Workshops, Los Alamitos, CA, USA, November 2007, pp. 363–367.

[95] Anh Kim Nguyen, Huong Thanh Le, Natural language interface construction using semantic grammars, in: Proceedings of the 10th Pacific Rim International Conference on Artificial Intelligence (PRICAI-08), 2008, pp. 728–739.

**[23] Ann Copestake, Karen Sparck Jones, Natural language interfaces to databases, Knowledge Engineering Review 5 (1990) 225–249. (NOVO)**

This paper reviews the current state of the art in natural language access to databases. This has been a long-standing area of work in natural language processing. But though some commercial systems are now available, providing front ends has proved much harder than was expected, and the necessary limitations on front ends have to be recognized. The paper discusses the issues, both general to language and task-specific, involved in front end design, and the way these have been addressed, concentrating on the work of the last decade. The focus is on the central process of translating a natural language question into a database query, but other supporting functions are also covered. The points are illustrated by the use of a single example application. The paper concludes with an evaluation of the current state, indicating that future progress will depend on the one hand on general advances in natural language processing, and on the other on expanding the capabilities of traditional databases.

**A Phrasal Approach to Natural Language Interfaces over Databases (2005)**

This short paper introduces the STEP system for natural language access to relational databases. In contrast to most work in the area, STEP adopts a phrasal approach; an administrator couples phrasal patterns to elementary expressions of tuple relational calculus. This ‘phrasal lexicon’ is used bi-directionally, enabling the generation of natural language from tuple relational calculus and the inverse parsing of natural language to tuple calculus. This ability to both understand and generate natural language enables STEP to engage the user in clarification dialogs when the parse of their query is of questionable quality or is open to multiple interpretations. An on-line demonstration of STEP is accessible at http://www.cs.umu.se/~mjm/step.

**A survey on question answering systems with classification (2016)**

Question answering systems (QASs) generate answers of questions asked in natural languages. Early QASs were developed for restricted domains and have limited capabilities. Current QASs focus on types of questions generally asked by users, characteristics of data sources consulted, and forms of correct answers generated. Research in the area of QASs began in 1960s and since then, a large number of QASs have been developed. To identify the future scope of research in this area, the need of a comprehensive survey on QASs arises naturally. This paper surveys QASs and classifies them based on different criteria. We identify the current status of the research in the each category of QASs, and suggest future scope of the research.

**Question answering systems: survey and trends (2015)**

The need to query information content available in various formats including structured and unstructured data (text in natural language, semi-structured Web documents, structured RDF data in the semantic Web, etc.) has become increasingly important. Thus, Question Answering Systems (QAS) are essential to satisfy this need. QAS aim at satisfying users who are looking to answer a specific question in natural language. In this paper we survey various QAS. We give also statistics and analysis. This can clear the way and help researchers to choose the appropriate solution to their issue. They can see the insufficiency, so that they can propose new systems for complex queries. They can also adapt or reuse QAS techniques for specific research issues.

[**A survey of natural language interface to database management system**](https://pdfs.semanticscholar.org/5b8e/98c1ed296e6b748c063ab3e12865706cb7de.pdf) **(2012)**

This paper has attempted to serve two purposes: to introduce the reader to the field of Nlidbs by describing some of the central issues, and to indicate the current situation in this area by outlining the facilities and methods of typical implemented systems. The goal of surveying the field can be achieved only incompletely at any given moment. Research is done from the last few decades on Natural Language Interfaces. With the advancement in hardware processing power, many NLIDBs mentioned in historical background got promising results. Though several NLIDB systems have also been developed so far for commercial use but the use of NLIDB systems is not wide-spread and it is not a standard option for interfacing to a database. This lack of acceptance is mainly due to the large number of deficiencies in the NLIDB system in order to understand a natural language.

**A Novel Approach Towards Incorporating Context Processing Capabilities in NLIDB System (2013)**

This paper presents a novel approach to categorize, model and identify contextual information in natural language interface to database (NLIDB) systems. The interactions between user and system are categorized and modeled based on the way in which the contextual information is utilized in the interactions. A relationship schema among the responses (user and system responses) is proposed. We present a novel method to identify contextual information in one specific type of user system interaction. We report on results of experiments with the university related queries.

One of the main reasons for the less acceptance of these systems in realtime applications is that they lack robust context processing capabilities (Bertomeu et al., 2006). Currently there is very little work which explicitly aims to investigate the role of context processing capabilities in NLIDB systems. However, the importance of context processing capabilities has been explored extensively in Question Answering systems (Chai and Jin, 2004; Kato et al., 2004; Kirschner and Bernardi, 2007; Negri and Kouylekov, 2007; Kirschner and Bernardi, 2010). Users often fail to express their intention (information need) in a single NL query (user re- sponse) (Bertomeu et al., 2006). Hence to answer a sequence of related NL queries, NLIDB systems should keep track of contextual information. NLIDB systems which do not use contextual information (non-contextual NLIDB) fail to completely capture the user’s intention.

[**An Overview of NLIDB Approaches and Implementation for Airline Reservation System**](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.686.4401&rep=rep1&type=pdf) **(2014)**

Relational databases are queried using database query languages such as SQL. Natural language interfaces to databases (NLIDB) are systems that translate a natural language sentence into a database query. In this modern techno-crazy world, as more and more laymen access various systems and applications through their smart phones and tablets, the need for Natural Language Interfaces (NLIs) has increased manifold. The challenges in Natural language Query processing are interpreting the sentence correctly, removal of various ambiguity and mapping to the appropriate context. Natural language access problem is actually composed of two stages - Linguistic processing and Database processing. NLIDB techniques encompass a wide variety of approaches. The approaches include traditional methods such as Pattern Matching, Syntactic Parsing and Semantic Grammar to modern systems such as Intermediate Query Generation, Machine Learning and Ontologies. In this report, various approaches to build NLIDB systems have been analyzed and compared along with their advantages, disadvantages and application areas. Also, a natural language interface to a flight reservation system has been implemented comprising of flight and booking inquiry systems.

**A Novel Approach Towards Building a Generic, Portable and Contextual NLIDB System (master degree - 2015)**

Natural Language Interface to Database (NLIDB) systems allow the users to query databases in a natural language. A NLIDB system takes a natural language (NL) query as input from the user and converts it to a database query language like SQL (structured query language). In this work, we present a novel approach to design a generic, portable and contextual NLIDB system. We have designed a novel computational paninian grammar (CPG) framework based semantic frames (i.e. domain knowledge) to identify the domain terms in the NL query. We propose a novel graph based method to identify join conditions of the SQL query. By using our approach, the domain of a NLIDB system can be easily changed without disturbing the implementation details of the system. Our approach greatly reduces both the size of the domain knowledge required to develop a NLIDB system and the time to create the domain knowledge. We propose a novel approach to identify contextual information in NLIDB systems.

We present a novel classification of the interactions between user and system: Strongly Coherent interaction, Coherent interaction and Weakly Coherent interaction. We proposed three models depending on the way in which the contextual information can be utilized in the interactions: Linear Disjoint Model, Linear Coincident Model and Non-Linear Model. We propose a relationship schema among the responses (user and system responses). Further, we also present a novel method to identify contextual information in Linear Disjoint interactions. We evaluated our approach on university related queries and the results confirm the viability of the proposed approach.

**A Survey: Natural Language Interface to Databases (2015 - Patel) - Only two references**

Information plays an important role in our everyday life and databases are widely used for storing and retrieving information. Database technology is having major impact in the world of computing. To access the information from database one need to have knowledge of database query language such as SQL. Because the naïve user may not be aware of the syntax of SQL and structure of database, s/he may not be able to write the SQL queries. Non -technical users may query relational databases in their natural language (i.e. English) instead of using SQL. This idea of using a Natural Language instead of SQL has lead to an approach of building Natural Language Interface to Relational Database. This paper is an introduction to the natural language interface to databases (NLIDB).

**Natural Language Interface to Database (2013- Gaikwad) – Only one reference**

The need for natural language interfaces to database has become increasingly acute as more and more people access information from web browsers. Yet NLI (Natural Language Interface) is only usable if they map natural language questions to SQL queries correctly. Natural Language processing is becoming one of the most active areas in Human-Computer Interaction. It is a branch of AI (Artificial Intelligence) which includes in information retrieval, machine translation and Language analysis. The goal of NLP (Natural Language Processing) is to enable communication between people and computers without requiring to memorization of complex commands and procedures. In other words NLP (Natural Language Processing) is techniques which can make the computer understand the languages naturally used by humans. The main purpose of natural Language Query Processing is for an English sentence to be interpreted by the computer and appropriate action taken. Asking questions to databases in natural language is a very convenient and easy method of data access, especially for casual users who do not understand complicated database query languages such as SQL.

[**A Dynamic Form-Based Natural Language Interface to Data Warehouse Question Answering**](http://ijccer.org/index.php/ojs/article/view/61) **(2013) – Zero references**

Form-based query interfaces are widely used to access databases as they provide selectable options for building query. The data warehouses are used for decision-making by executives of enterprises who are commonly non-technical. Form-based interface can support in decision -making due to ease of use. In this work, we present such a form-based approach that exhibits an interface to take a natural language query from the user. The natural language query is then parsed and mapped with data warehouse semantics. Finally, a form is dynamically generated based on mapped information that user fills up to complete their query and to change the selection at that time. This system supports easy decision-making ability that ultimately increases revenue.

[**Natural Language Interface To Database**](https://irjet.net/archives/V3/i10/IRJET-V3I1032.pdf) **(2016 - Chaitali) – Zero references**

Information is playing an important role in our lives. One of the major sources of information is databases. Databases and database technology are having major impact on the growing use of computers. Almost all IT applications are storing and retrieving information from databases. There are various interfaces available to retrieve data such as form based, natural language and keyword based. Data retrieval from the database requires knowledge of database language like SQL. In this paper we have proposed architecture of a Natural Language and Keyword Based Interface for Database (NLKBIDB) which provides solution for syntactically correct and incorrect natural language input query. Our partial experiment of Lexical Analyzer and Keyword based interface on agriculture survey database solves 53% of syntactically incorrect query which will not be solved by natural language interface resulting in increase of rate of SQL query conversion.

**NATURAL LANGUAGE INTERFACE TO DATABASES (2017- Kuthandi) – Zero references**

In the world of computing, information plays an important role in our lives. One of the major sources of information is database. Database and Database technology are having major impact on the growing use of computers. Almost all IT applications are storing and retrieving the information or data from the database. Database Management Systems (DBMS) have been widely used for storing and retrieving data. However, databases are often hard to use since their interface is quite rigid in cooperating with users. For storing and retrieving the information from database requires the knowledge of database language like SQL. Structured Query Language (SQL) is an ANSI standard for accessing and manipulating the information stored in database. However, everyone may not be able to write the SQL query as they may not be aware of the syntax and structure of SQL and database respectively. The purpose of Natural Language Interface is to allow users to compose questions in Natural Language and receive the response also in Natural Language. The idea of using Natural Language instead of SQL has promoted the development of new type of processing called Natural Language Interface to Database (NLIDB). This paper discuss about an introduction of Intelligent Database System, Natural Language Processing and Natural Language Interface to Database. It also gives a brief overview of subcomponent of NLIDB, techniques used to development of NLIDB along with its architecture.

**WikiSQL**

**Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning (08/2017)**

A significant amount of the world's knowledge is stored in relational databases. However, the ability for users to retrieve facts from a database is limited due to a lack of understanding of query languages such as SQL. We propose Seq2SQL, a deep neural network for translating natural language questions to corresponding SQL queries. Our model leverages the structure of SQL queries to significantly reduce the output space of generated queries. Moreover, we use rewards from in-the-loop query execution over the database to learn a policy to generate unordered parts of the query, which we show are less suitable for optimization via cross entropy loss. In addition, we will publish WikiSQL, a dataset of 80654 hand-annotated examples of questions and SQL queries distributed across 24241 tables from Wikipedia. This dataset is required to train our model and is an order of magnitude larger than comparable datasets. By applying policy-based reinforcement learning with a query execution environment to WikiSQL, our model Seq2SQL outperforms attentional sequence to sequence models, improving execution accuracy from 35.9% to 59.4% and logical form accuracy from 23.4% to 48.3%.

**Pointing Out SQL Queries From Text (11/2017)**

The digitization of data has resulted in making datasets available to millions of users in the form of relational databases and spreadsheet tables. However, a majority of these users come from diverse backgrounds and lack the programming expertise to query and analyze such tables. We present a system that allows for querying data tables using natural language questions, where the system translates the question into an executable SQL query. We use a deep sequence to sequence model in wich the decoder uses a simple type system of SQL expressions to structure the output prediction. Based on the type, the decoder either copies an output token from the input question using an attention-based copying mechanism or generates it from a fixed vocabulary. We also introduce a value-based loss function that transforms a distribution over locations to copy from into a distribution over the set of input tokens to improve training of our model. We evaluate our model on the recently released WikiSQL dataset and show that our model trained using only supervised learning significantly outperforms the current state-of-the-art Seq2SQL model that uses reinforcement learning.

**SQLNet: Generating Structured Queries From Natural Language Without Reinforcement Learning**

**Xiaojun Xu, Chang Liu, Dawn Song (Submitted on 13 Nov 2017)**

Synthesizing SQL queries from natural language is a long-standing open problem and has been attracting considerable interest recently. Toward solving the problem, the de facto approach is to employ a sequence-to-sequence-style model. Such an approach will necessarily require the SQL queries to be serialized. Since the same SQL query may have multiple equivalent serializations, training a sequence-to-sequence-style model is sensitive to the choice from one of them. This phenomenon is documented as the "order-matters" problem. Existing state-of-the-art approaches rely on reinforcement learning to reward the decoder when it generates any of the equivalent serializations. However, we observe that the improvement from reinforcement learning is limited.

In this paper, we propose a novel approach, i.e., SQLNet, to fundamentally solve this problem by avoiding the sequence-to-sequence structure when the order does not matter. In particular, we employ a sketch-based approach where the sketch contains a dependency graph so that one prediction can be done by taking into consideration only the previous predictions that it depends on. In addition, we propose a sequence-to-set model as well as the column attention mechanism to synthesize the query based on the sketch. By combining all these novel techniques, we show that SQLNet can outperform the prior art by 9% to 13% on the WikiSQL task.

**Natural Language to Structured Query Generation via Meta-Learning**

Po-Sen Huang, Chenglong Wang, Rishabh Singh, Wen-tau Yih, Xiaodong He

(Submitted on 2 Mar 2018 (v1), last revised 18 Jul 2018 (this version, v4))

In conventional supervised training, a model is trained to fit all the training examples. However, having a monolithic model may not always be the best strategy, as examples could vary widely. In this work, we explore a different learning protocol that treats each example as a unique pseudo-task, by reducing the original learning problem to a few-shot meta-learning scenario with the help of a domain-dependent relevance function. When evaluated on the WikiSQL dataset, our approach leads to faster convergence and achieves 1.1%-5.4% absolute accuracy gains over the non-meta-learning counterparts.

**TypeSQL: Knowledge-based Type-Aware Neural Text-to-SQL Generation**

Tao Yu, Zifan Li, Zilin Zhang, Rui Zhang, Dragomir Radev (Submitted on 25 Apr 2018)

Interacting with relational databases through natural language helps users of any background easily query and analyze a vast amount of data. This requires a system that understands users' questions and converts them to SQL queries automatically. In this paper we present a novel approach, TypeSQL, which views this problem as a slot filling task. Additionally, TypeSQL utilizes type information to better understand rare entities and numbers in natural language questions. We test this idea on the WikiSQL dataset and outperform the prior state-of-the-art by 5.5% in much less time. We also show that accessing the content of databases can significantly improve the performance when users' queries are not well-formed. TypeSQL gets 82.6% accuracy, a 17.5% absolute improvement compared to the previous content-sensitive model.

**Coarse-to-Fine Decoding for Neural Semantic Parsing**

Li Dong, Mirella Lapata (Submitted on 12 May 2018)

Semantic parsing aims at mapping natural language utterances into structured meaning representations. In this work, we propose a structure-aware neural architecture which decomposes the semantic parsing process into two stages. Given an input utterance, we first generate a rough sketch of its meaning, where low-level information (such as variable names and arguments) is glossed over. Then, we fill in missing details by taking into account the natural language input and the sketch itself. Experimental results on four datasets characteristic of different domains and meaning representations show that our approach consistently improves performance, achieving competitive results despite the use of relatively simple decoders.

**The Natural Language Decathlon: Multitask Learning as Question Answering**

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, Richard Socher (Submitted on 20 Jun 2018)

Deep learning has improved performance on many natural language processing (NLP) tasks individually. However, general NLP models cannot emerge within a paradigm that focuses on the particularities of a single metric, dataset, and task. We introduce the Natural Language Decathlon (decaNLP), a challenge that spans ten tasks: question answering, machine translation, summarization, natural language inference, sentiment analysis, semantic role labeling, zero-shot relation extraction, goal-oriented dialogue, semantic parsing, and commonsense pronoun resolution. We cast all tasks as question answering over a context. Furthermore, we present a new Multitask Question Answering Network (MQAN) jointly learns all tasks in decaNLP without any task-specific modules or parameters in the multitask setting. MQAN shows improvements in transfer learning for machine translation and named entity recognition, domain adaptation for sentiment analysis and natural language inference, and zero-shot capabilities for text classification. We demonstrate that the MQAN's multi-pointer-generator decoder is key to this success and performance further improves with an anti-curriculum training strategy. Though designed for decaNLP, MQAN also achieves state of the art results on the WikiSQL semantic parsing task in the single-task setting. We also release code for procuring and processing data, training and evaluating models, and reproducing all experiments for decaNLP.

**Robust Text-to-SQL Generation with Execution-Guided Decoding**

Chenglong Wang, Kedar Tatwawadi, Marc Brockschmidt, Po-Sen Huang, Yi Mao, Oleksandr Polozov, Rishabh Singh (Submitted on 9 Jul 2018 (v1), last revised 13 Sep 2018 (this version, v3))

We consider the problem of neural semantic parsing, which translates natural language questions into executable SQL queries. We introduce a new mechanism, execution guidance, to leverage the semantics of SQL. It detects and excludes faulty programs during the decoding procedure by conditioning on the execution of partially generated program. The mechanism can be used with any autoregressive generative model, which we demonstrate on four state-of-the-art recurrent or template-based semantic parsing models. We demonstrate that execution guidance universally improves model performance on various text-to-SQL datasets with different scales and query complexity: WikiSQL, ATIS, and GeoQuery. As a result, we achieve new state-of-the-art execution accuracy of 83.8% on WikiSQL.

**IncSQL: Training Incremental Text-to-SQL Parsers with Non-Deterministic Oracles (Submitted on 13 Sep 2018)**

We present a sequence-to-action parsing approach for the natural language to SQL task that incrementally fills the slots of a SQL query with feasible actions from a pre-defined inventory. To account for the fact that typically there are multiple correct SQL queries with the same or very similar semantics, we draw inspiration from syntactic parsing techniques and propose to train our sequence-to-action models with non-deterministic oracles. We evaluate our models on the WikiSQL dataset and achieve an execution accuracy of 83.7% on the test set, a 2.1% absolute improvement over the model trained with traditional static oracles assuming a single correct target SQL query. When further combined with the execution-guided decoding strategy, our model sets a new state-of-the-art performance at an execution accuracy of 87.1%.

This is a work-in-progress technical report.