State of the Art and Open Challenges in Natural Language Interfaces to Data

Fatma Özcan
fozcan@us.ibm.com
IBM Research - Almaden

Abdul Quamar
ahquamar@us.ibm.com
IBM Research - Almaden

Jaydeep Sen
jaydesen@in.ibm.com
IBM Research - India

Chuan Lei
chuan.lei@ibm.com
IBM Research - Almaden

Vasilis Efthymiou
vasilis.efthymiou@ibm.com
IBM Research - Almaden

ABSTRACT

Recent advances in natural language understanding and processing resulted in renewed interest in natural language based interfaces to data, which provide an easy mechanism for non-technical users to access and query the data. While early systems only allowed simple selection queries over a single table, some recent work supports complex BI queries, with many joins and aggregation, and even nested queries. There are various approaches in the literature for interpreting user’s natural language query. Rule-based systems try to identify the entities in the query, and understand the intended relationships between those entities. Recent years have seen the emergence and popularity of neural network based approaches which try to interpret the query holistically, by learning the patterns. In this tutorial, we will review these natural language interface solutions in terms of their interpretation approach, as well as the complexity of the queries they can generate. We will also discuss open research challenges.

ACM Reference Format:

Fatma Özcan, Abdul Quamar, Jaydeep Sen, Chuan Lei, and Vasilis Efthymiou. 2020. State of the Art and Open Challenges in Natural Language Interfaces to Data. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data (SIGMOD’20), June 14–19, 2020, Portland, OR, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3318464.3383128

SIGMOD’20, June 14–19, 2020, Portland, OR, USA

ACM ISBN 978-1-4503-6735-6/20/06…$15.00
https://doi.org/10.1145/3318464.3383128

1 INTRODUCTION

Many business users and line of business owners rely on technical people to query and gain insights from their business data. These technical people are experts on using complex query languages such as SQL or SPARQL. Today, it is vital for non-technical business owners to derive insights from their data as quickly as possible to make effective business decisions. Natural language interfaces enable such non-technical users to explore their business data in a more natural way and without relying on technical users’ help.

There are several challenges in building natural language interfaces to data [2]. Ambiguity in natural language is a big challenge, making it difficult to understand the semantics of the query and hence the user intent. The complexity of queries that one can ask over the data has been increasing over time. Some systems [1, 51] only allowed a set of keywords, with very limited expressive power. Some of the early work [15, 66] on natural language interfaces (NLI) to data mostly focused on simple queries that access a single table using some selection criteria. Later works allow a full-blown English statement and try to disambiguate among the multiple meanings of the words and their relationships. With the recent advances in NLP [61], both the complexity of input natural language statements, as well as the generated SQL and SPARQL queries have increased over time, and renewed the interest in NLI to data (NLID). To interpret the user’s natural language query, many systems try to identify the entities in the query, and understand the intended relationships between those entities. These entity-based interpretation techniques [15, 29–32, 44, 46, 49] vary widely in terms of the complexity of the queries that they generate. Some of the recent works are also using machine learning and deep learning based [9, 22, 56, 59, 62, 67, 69] techniques, and require good training sets, which are hard to obtain. Others require user feedback [27, 31, 34] to disambiguate. One natural extension to one-shot NLQ approaches is the extension to dialogue, allowing continuous context and a well-defined user feedback.

In this tutorial, we will review these natural language interface solutions in terms of their interpretation approach, as well as the complexity of the queries they can generate. We will also discuss how to extend the one-shot query approaches to dialogue, taking advantage of the context for disambiguation. Finally, we will conclude with a discussion on challenges that need to be addressed before these systems can be widely adapted.

2 TARGET AUDIENCE AND OUTLINE

Target Audience. The target audience groups of this tutorial are the following:

- Researchers and developers who would like to learn the challenges that natural language interfaces pose while building systems, as well as the recent industry trends and offerings in the NLID,
- PhD students who are seeking a high-impact research topic in this area.

Prerequisites. No prior knowledge is required to understand the concepts in the tutorial, but we assume some familiarity with databases and basic machine learning terminology.

Outline. The 1.5-hour tutorial is tentatively structured as follows:

(1) **Introduction and overview**
(2) **Complexity of generated queries**
  - Simple queries (simple selection on a single table)
  - Moderate queries (join across multiple tables)
  - Complex (BI) queries (nested sub-queries)
(3) **Natural language query interpretation**
  - Entity-based approaches
  - Machine learning-based approaches
  - Hybrid approaches
(4) **Extension to dialogue**
(5) **Open challenges**

There was a tutorial on natural language interfaces at SIGMOD 2017 [33]. In our tutorial, we cover a wider scope of topics (i.e., query complexity and extension to dialogue) and more recent approaches, especially machine learning and deep learning based methods. We also focus more on building natural language querying systems, rather than managing natural language text data, with particular emphasis on the wide spectrum of query complexity and conversational NLID.

3 COMPLEXITY OF GENERATED QUERY

The query complexity can be categorized into 4 groups:

- simple selection queries on a single table
- aggregation queries on a single table involving `GROUP BY` and `ORDER BY`
- queries involving multiple tables (`JOIN`), and
- complex Business Intelligence (BI) or analytic queries with nested sub-queries.

We will classify the existing NLID systems and their associated methodologies based on these categories.

Selection Queries on Single Table. Early NLID systems [15, 66] mostly provide keyword-based entity extraction. They only consider each individual word for a possible match in meta data or data instances. Such systems can only handle simple filter queries but cannot detect other clauses like `GROUP BY` and `ORDER BY` in natural language queries. Seq2SQL [69] uses reinforcement learning to train end-to-end translation model from NLQ to SQL on single table based on WikiSQL data [69]. SQLNet [59] extends from Seq2SQL by avoiding the sequence-to-sequence structure when ordering does not matter in SQL query conditions. It also targets single table queries from WikiSQL.

Aggregation Queries on Single Table. Pattern-based NLID systems [16, 51, 68] introduce the use of natural language patterns for detecting more SQL clauses like aggregation, `GROUP BY`, `ORDER BY`, etc. Exploiting fixed patterns in natural language enables such systems to overcome the limitations of keyword-based systems, but they are limited to those fixed patterns. For example, simple natural language patterns like "by", "total/average" enable such systems to detect `GROUP BY` and aggregation, respectively.

Queries involving Multiple Tables. To generate SQL queries with joins across multiple tables, NLID systems [9, 31, 44] evolve towards using natural language parsing with additional information from the backend schema to detect relationships. The parse tree is mainly used to analyze the dependency among tokens, and to infer the join paths between matched database elements. ATHENA [44] further utilizes domain ontologies as an abstraction of the backend database to facilitate intelligent domain reasoning. TEMPLAR [7] leverages information from the SQL query log to improve keyword mapping and join path inference. DBPal [9] aims to generate join queries based on the information learnt from domain-specific training data, and requires many training examples with different join paths.

BI Queries with Nested Sub-queries. Recently, there has been a trend to deploy NLID in commercial systems [23, 38, 45, 48, 50, 57] for business intelligence (BI) use cases. Typical BI queries often involve nested sub-queries with all other query clauses. However, none of the NLID systems so far has the capability of generating such complex BI queries.

Instead, they heavily rely on explicit user actions via interactive tools and visualizations in combination of natural language queries for creating analytic dashboards. Some preliminary systems [9, 31, 46] handle a collection of BI queries with nesting, but a full-fledged solution remains an open challenge.

4 NATURAL LANGUAGE QUERY INTERPRETATION

At the core of a NLID system lies its ability to understand/interpret a user query expressed in natural language. In this section, we classify works according to their interpretation method: (1) works that follow an entity-based approach, recognizing the different entities involved in a query, (2) machine learning-based approaches, classifying a user query into one of the possible query templates, and (3) hybrid approaches that combine entity-based and machine learning-based approaches.

4.1 Entity-Based Approaches

Entity-based approaches recognize the constituent entities mentioned in a query, as well as their relationships, based on an internal representation of the underlying data, e.g., an index structure, a taxonomy, or an ontology.

Earlier works (e.g., BELA [53], QUICK [66], Précis [26, 47]) first perform some parsing of the natural language query to transform it to a machine-readable format, and later look up the slots that correspond to entities using a simple index structure. For example, Précis [26, 47] first transforms a given query to disjunctive normal form (DNF) using well-studied DNF transformation algorithms [36], and then consults an inverted index over the contents of the underlying database to retrieve candidate interpretations for each disjunct. BELA [53] uses a lexical tree adjoining grammar [52] to parse the input queries based on part-of-speech tags. This parsing results in a set of SPARQL query templates, each corresponding to a possible interpretation of the given query. For filling the unknown slots in the SPARQL queries, an inverted index, built from DBpedia [6] entity names, is consulted. Similarly, QUICK [66] binds a keyword-based query to the lookup results from an inverted index that is built on the instances, concepts, and properties of the underlying data. In addition to the steps described before, QUICK employs an additional step in which users can interactively select one of the suggested query interpretations that best fits their query.

Later works extend the query interpretation capability by further employing taxonomies to capture the semantics of the underlying data. NaLIR [30–32] uses Stanford NLP Parser [17] to obtain a linguistic understanding of the input query in the form of a parse tree. Tree nodes corresponding to entities are mapped to the underlying data using a WordNet-based similarity function [58]. This may provide multiple mappings per tree node, which are then clarified by users. Recently, Duoqest [8] leverages guided partial query enumeration to efficiently explore the space of possible queries for a given NL query.

SODA [15] is one of the earliest works to employ ontologies for query interpretation. It looks up each query keyword in two different indices: one for the data in a database, and one for the meta-data in ontologies. This leads to multiple interpretations per query, which are ranked based on an aggregation of the scores associated with each lookup result. Query interpretations are extended using ontologies, e.g., super-classes of the lookup results add new interpretations.

USI Answers [54] employs syntactic rules to identify query parts that refer to entities. In a lookup step, it produces the candidate entities mentioned in the query, generating different query interpretations. An ontology models the underlying data and is used to determine if there is a relationship between the identified entities mentioned in the query.

TR Discover [49] uses a feature-based context-free grammar for parsing natural language queries, also providing query auto-completion. When a user starts typing a query segment and selects one of the suggested lexical entries (i.e., an entity, an object, or a property) for this segment, TR Discover suggests the next lexical entries that are reachable from the selected query part, based on the rules of the context-free grammar. The ranking of these suggestions is based on the nodes centrality in an RDF graph, in which each node represents a different lexical entry.

ATHENA [29, 44, 46] maps parts of the natural language query to concepts and relationships in an ontology that captures the semantics of a relational database. The ontology and the mappings to the underlying data can be either provided manually, or generated automatically from the database information [24]. ATHENA uses an intermediate query language before translating the input query into SQL. Lei et al. [28] introduce a query relaxation technique to improve the query understanding capability of ATHENA by leveraging external knowledge sources, with a focus on medical KBs. The proposed technique fills the gap between the terms stored in the KBs and the colloquial and imprecise terminology used in user queries. Recently, Quamar et al. [42] adopt ATHENA's ontology-driven approach to further build a conversation system for domain-specific knowledge bases.

Since entity-based approaches try to understand the relationship between different entities in the user query, they can handle complex input queries and generate complex structured queries. Moreover, it is easier to incorporate domain knowledge, as these systems use semantics rich domain taxonomies and ontologies. However, they are highly sensitive to variations and paraphrasing of the user query.

4.2 Machine Learning-Based Approaches

The recent success of artificial intelligence and in particular deep learning triggered a new trend of building NLID systems. The basic idea is to apply supervised machine learning techniques on a set of question/answer pairs where the questions are the natural language queries and the answers are the respective SQL or SPARQL statements. These questions and answers are first transformed into a vector by applying word embedding techniques. Then, these vectors are consumed by a deep neural network [69].

Seq2SQL [69] uses a deep neural network architecture with reinforcement learning to translate natural language to SQL. This approach was demonstrated to work on simple single-table queries without joins. DBPal [9, 56] avoids manually labeling large training data sets by synthetically generating a training set that only requires minimal annotations in the database. DBPal uses the database schema and query templates to describe NL/SQL-pairs. The results show that on a single-table data set DBPal performs better than the semantic parsing approach. SQLNet [59] uses column attention and employs a sketch-based method and generates SQL as a slot-filling task. This fundamentally avoids the sequence-to-sequence structure when ordering does not matter in SQL query conditions. TypeSQL [62] improves upon SQLNet by proposing a different training procedure and utilizing types extracted from either knowledge graph or table content to help model better understand entities and numbers in the question. DialSQL [22] is a dialogue-based structured query generation framework that leverages human intelligence to boost the performance of existing algorithms via user interaction. DialSQL is capable of identifying potential errors in a generated SQL query and asking users for validation via simple multi-choice questions. User feedback is then leveraged to revise the query. Zhang et al. [67] propose SQL query generation by editing the query in the previous turn. The previous query is first encoded as a sequence of tokens, and the decoder computes a switch to change it at the token level. This sequence editing mechanism models token-level changes and is thus robust to error propagation. Furthermore, to capture the user utterance and the complex database schemas in different domains, an utterance-table encoder is used based on BERT [18] to jointly encode the user utterance and column headers with co-attention, and a table-aware decoder is adopted to perform SQL generation with attentions over both the user utterance and column headers. Most recently, Wang et al. [55] introduce a general purpose transfer-learnable NLID system. The proposed system adopts the data management principle of separating data and its schema, but with the additional support for the idiosyncrasy and complexity of natural languages.

Machine learning-based approaches have shown promising results in terms of robustness to NL variations. However, these systems still have limited capability of handling complex queries involving multiple tables with aggregations, and nested queries. In addition, they require large amounts of training data, which makes the domain adaption challenging.

4.3 Hybrid Approaches

Hybrid approaches [10–12] combine entity- and learning-based query understanding in a multi-step strategy, using one of the two approaches as a filtering mechanism. For example, QUEST [12] first chooses the entities that are relevant to the keywords in the query based on Hidden Markov Models (HMM), trained on a data set of previous searches, validated by the user. The relationships between the entities extracted from the query are then computed based on heuristic rules that consider the relationships of those entities in the database. The candidate interpretations are ranked based on the aggregate confidence scores returned by the HMM. However, these systems are still not capable of covering a full spectrum of the complexity of generated queries. Hence, more research is needed on hybrid approach that leverages the best from both worlds.

5 EXTENSION TO DIALOGUE

Dialogue is an extension of natural language querying to a two-way conversation between the user and the system (or service). Conversational interfaces, which provide the ability to interact with business applications and data using a two-way dialogue, are rapidly gaining popularity [20] because of their ability to (1) understand, respond and clarify ambiguity in natural language, and (2) persist the context of conversation across multiple turns. One possible solution to handling complex queries is to express them as a sequence of simpler questions. This is in line with machine learning-based approaches for query translation which allow for a richer linguistic variability in query expressions and user flexibility while restricting their applicability to simpler individual queries [2]. However, some of the complex BI queries cannot be easily broken down into simpler questions, and even some simple BI queries require complex nested SQL.

Today's chatbot platforms (e.g., Google Dialogflow, Facebook Wit.ai, Microsoft Bot Framework, IBM Watson Assistant, etc.) allow users to interact through natural language (e.g., English, Spanish, Mandarin, etc.) speech or text. Using these platforms, developers can create many kinds of natural language interfaces for any kind of domain (e.g., weather, music, finance, travel, healthcare, etc.). In this tutorial, our focus is on chatbots for data exploration. This is different from social chatbots and task-oriented chatbots supported by open-domain agents, such as Microsoft Cortana [39], Apple Siri [4], Amazon Alexa [3], which are useful for accomplishing day-to-day tasks, such as checking weather forecasts, playing music, setting device timers, etc.

The set of all possible interactions with a conversational interface is defined in terms of three main components that enable its natural language understanding and ability to interact with users: intents, entities, and dialogue [20]. Intents are goals/actions that are expressed in the user utterances, while entities represent information that is relevant to the user's purpose. These entities would typically consist of elements from the domain schema well as actual data instances that are relevant to conversation. The dialogue provides a response to a user conditioned on the identified intents, entities in the user's input and the current context of the conversation. The dialogue structure is therefore especially relevant as it defines the space of all possible natural language interactions that the conversational interface is required to support.

Recent advances in machine learning, particularly in neural networks, have allowed for complex dialogue management methods and conversation flexibility for conversational interfaces. Three approaches are commonly used in building the dialogue structure for a conversational interface. Rule-based approaches [35, 37] used in finite-state dialogue management systems are simple to construct for tasks that are straightforward and well-structured, but have the disadvantage of restricting user input to predetermined words and phrases. Frame-based systems [13, 19, 21] address some of the limitations of finite state dialogue management by enabling a more flexible dialogue. Frame-based systems enable the user to provide more information than required by the system's question, while the conversation system keeps track of what information is required and asks questions accordingly. Unlike finite-state and frame-based systems, agent-based systems [14, 40, 43, 60] are able to manage complex dialogues, where the user can initiate and lead the conversation. Agent-based methods for dialogue management are typically statistical models trained on corpora of real human computer dialogue, offering more robust speech recognition and performance, as well as better scalability, and greater scope for adaptation. Among the different approaches, agent based systems are the most flexible form of dialogue management, and hence suitable for iterative data exploration driven by the user.

In addition to dialogue design, there are many other challenges in building effective conversation services to explore data. First, building these systems requires a lot of domain knowledge and manual setup. Specifically, designing effective conversational interfaces for iterative data exploration for supporting enterprise applications requires deep domain understanding. This includes semantic understanding of the underlying data in terms of the entities and relationships it represents as well as the expected workload against the data. Together, such deep domain knowledge understanding facilitates designing for possible user intents and the required dialogue interactions. Ontologies provide a powerful abstraction for representing domain knowledge in terms of relevant entities, data properties and relationships. This can be used to bootstrap conversation systems to minimize the required manual labor and expediting their instantiation across different domains. Quamar et. al [42] follow this ontology-driven approach for building a conversational interface to domain-specific knowledge bases (KBs). They demonstrate the effectiveness of capturing patterns in the expected workload, mapping these patterns against the domain ontology to generate artifacts (i.e., intents, training examples, entities), and supporting dialogue for building a conversational interface.

Another challenge is the infusion of semantic domain knowledge into the intent classification. Ontologies can help in this regard as well. In particular, ontologies can augment the intent classifiers with greater linguistic variability and entity recognition capabilities through the provision of domain-specific synonyms and relaxation techniques [28].

6 OPEN CHALLENGES

Despite the recent increase in the number of research efforts focusing on this area, there are still many open challenges. In the following, we identify and briefly discuss the ones that we consider to be the most important.

Sub-queries. Handling natural language queries composed of one and multiple sub-queries is still one of the most challenging problems for NLIs. First, detecting whether a natural language query requires a nested query is non-trivial due to non-obvious linguistic patterns from the natural language queries. Second, building a nested query requires identifying proper sub-queries and figuring out the correct conditions to join the sub-queries to produce the correct query results.

Hybrid Approach. Neither the entity-based approaches nor the machine learning-based approaches can tackle all challenges in natural language querying. In general, the entity-based approaches provide better accuracy while the machine learning-based approaches offer greater flexibility (recall) in terms of the natural language queries, as they are more robust to variations in linguistic patterns. Therefore, more research is needed on hybrid approach that leverages the best from both worlds.

Conversation. There are many challenges for developing domain-specific conversational interfaces. Capturing the domain semantics and incorporating that into the conversation is non-trivial. This includes understanding the entities of the domain and their relationships, as well as the domain vocabularies and their synonyms. Providing sufficient training samples is yet another challenge.

Benchmarks. Evaluating NLID is a non-trivial task [5, 25]. With the current abundance of solutions that target this problem, a systematic evaluation of existing approaches becomes more and more a necessity. The first steps towards this goal (WikiSQL [69] and Spider [64]) have been very well-received by the community, focusing mostly on the learning-based approaches that target queries of lower complexity (Section 3). WikiSQL contains 80,654 pairs of NL questions and SQL queries which are manually annotated and distributed across 24, 241 tables from Wikipedia. The large volume of data enables machine learning based systems to train their model.

Similar to WikiSQL, WikiTableQuestions [41] is a popular benchmark for question answering on semi-structured HTML tables. The data set contains 2, 108 tables from a large variety of topics (more breadth) and 22, 033 NL questions with different complexity (more depth). Each question comes with a table from Wikipedia. Given the question and the table, the task is to answer the question based on the table.

Recent efforts also focus on providing data sets for evaluating multi-turn and conversational NLID [63, 65]. SParC [65] is a context-dependent/multi-turn version of the Spider data set. It consists of over 4,000 coherent question sequences, obtained from user interactions with 200 complex databases over 138 domains. CoSQL [63] is a dialogue version of the Spider and SParC data sets. It consists of 30k+ turns plus 10k+ annotated SQL queries, obtained from the same databases used in the Spider and SParC.

We believe that the next step to follow up on these benchmarking efforts is the inclusion of more complex analytical queries that will include more of the entity-based approaches and further push the state of the art.

Enterprise Adaption Natural language interfaces can democratize access to data within an enterprise by enabling non-technical users to explore the data easily. However, many enterprise applications require high accuracy, and the current state-of-the-art approaches still cannot achieved desirable levels. For natural interfaces to become widely adapted in the enterprise more research is needed to increase the precision while maintaining high recall for both simple and complex queries.

7 PRESENTERS

Fatma Özcan is a Principal Research Staff Member and a senior manager at IBM Research - Almaden. Her current research focuses on platforms and infra-structure for large-scale data analysis, knowledge graphs, NLQ and conversational interfaces to data, and query processing and optimization of semi-structured data. Dr Özcan got her PhD degree in computer science from University of Maryland, College Park. She has over 18 years of experience in industrial research, and has delivered core technologies into IBM products. She is the co-author of the book "Heterogeneous Agent Systems", and co-author of several conference papers and patents. She is the treasurer and secretary of ACM SIGMOD, and is on the board of trustees for the VLDB Endowment. She is an ACM Distinguished Member.

Abdul Quamar is a Research Staff Member at IBM Research - Almaden. He received his Ph.D. in Computer Science from the University of Maryland at College Park. His primary research interests are in large-scale distributed systems for management of structured, semi-structured and graph structured data. His current research focuses on building AI and cognitive systems for supporting natural language and conversational interfaces for data exploration.

Jaydeep Sen is a Research Software Engineer in IBM Research AI, India Lab. He received his Masters of Technology in Computer Science and Engineering from IIT Kanpur in 2013. His research interests include applications for natural language understanding, semantic reasoning, designing intelligent algorithms for "learning from small data" applications. He is currently contributing to the next line of AI inventions for Watson and IBM Cloud offerings.

Chuan Lei is a Research Staff Member at IBM Research - Almaden. He received his Ph.D. in Computer Science from Worcester Polytechnic Institute. Before joining IBM, he was a Researcher at NEC Laboratories America. His research focuses on building systems to support natural language query and conversational interfaces to knowledge bases. In addition, Chuan works on building systems for scalable data stream processing and complex event processing.

Vasilis Efthymiou is a Postdoctoral Researcher at IBM Research - Almaden. He received his Ph.D. from University of Crete, Greece. His research focuses on data integration, knowledge management, and query answering. He has given tutorials in top-tier data management conferences (WWW, ICDE, CIKM, ESWC), and he has co-authored the book "Entity Resolution in the Web of Data".

REFERENCES

[1] B Aditya, Gaurav Bhalotia, Soumen Chakrabarti, Arvind Hulgeri, Charuta Nakhe, Parag Parag, and S Sudarshan. 2002. Banks: Browsing and keyword searching in relational databases. In *VLDB*. VLDB Endowment.

[2] Katrin Affolter, Kurt Stockinger, and Abraham Bernstein. 2019. A comparative survey of recent natural language interfaces for databases. *The VLDB Journal* 28, 5 (01 Oct 2019), 793–819. https://doi.org/10.1007/s00778-019-00567-8

[3] Amazon[US]. 2018. Amazon Alexa. https://developer.amazon.com/alexa.

[4] Apple[US]. 2018. Siri. https://www.apple.com/ios/siri/.

[5] Takuto Asakura, Jin-Dong Kim, Yasunori Yamamoto, Yuka Tateisi, and Toshihisa Takagi. 2018. A Quantitative Evaluation of Natural Language Question Interpretation for Question Answering Systems. In *JIST*. 215–231.

[6] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. 2007. DBpedia: A Nucleus for a Web of Open Data. In *ISWC*. 722–735.

[7] Christopher Baik, H. V. Jagadish, and Yunyao Li. 2019. Bridging the Semantic Gap with SQL Query Logs in Natural Language Interfaces to Databases. In *ICDE*. 374–385.

[8] Christopher Baik, Zhongjun Jin, Michael J. Cafarella, and H. V. Jagadish. 2020. Constructing Expressive Relational Queries with Dual-Specification Synthesis. In *CIDR*.

[9] Fuat Basik, Benjamin Hättasch, Amir Ilkhechi, Arif Usta, Shekar Ramaswamy, Prasetya Utama, Nathaniel Weir, Carsten Binnig, and Ugur Çetintemel. 2018. DBPal: A Learned NL-Interface for Databases. In *SIGMOD*. 1765–1768.

[10] Hannah Bast and Elmar Haussmann. 2015. More Accurate Question Answering on Freebase. In *CIKM*. 1431–1440.

[11] Asma Ben Abacha and Pierre Zweigenbaum. 2015. MEANS: A medical question-answering system combining NLP techniques and semantic Web technologies. *Inf. Process. Manage.* 51, 5 (2015), 570–594.

[12] Sonia Bergamaschi, Francesco Guerra, Matteo Interlandi, Raquel Trillo Lado, and Yannis Velegrakis. 2016. Combining user and database perspective for solving keyword queries over relational databases. *Inf. Syst.* 55 (2016), 1–19.

[13] Martin Beveridge and John Fox. 2006. Automatic Generation of Spoken Dialogue from Medical Plans and Ontologies. *J. of Biomedical Informatics* 39, 5 (2006), 482–499.

[14] Bing-Hwang Juang and S. Furui. 2000. Automatic recognition and understanding of spoken language - a first step toward natural human-machine communication. *Proc. IEEE* 88, 8 (2000), 1142–1165.

[15] Lukas Blunschi, Claudio Jossen, Donald Kossmann, Magdalini Mori, and Kurt Stockinger. 2012. SODA: Generating SQL for Business Users. *PVLDB* 5, 10 (2012), 932–943.

[16] Danica Damljanovic, Valentin Tablan, and Kalina Bontcheva. 2008. A Text-based Query Interface to OWL Ontologies. In *LREC*.

[17] Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. In *LREC*. 449–454.

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018).

[19] Kathleen Kara Fitzpatrick, Alison Darcy, and Molly Vierhile. 2017. Delivering Cognitive Behavior Therapy to Young Adults With Symptoms of Depression and Anxiety Using a Fully Automated Conversational Agent (Woebot): A Randomized Controlled Trial. *JMIR Ment Health* 4, 2 (2017), e19.

[20] Jianfeng Gao, Michel Galley, and Lihong Li. 2018. Neural Approaches to Conversational AI. *CoRR* abs/1809.08267 (2018).

[21] Toni Giorgino, Ivano Azzini, Carla Rognoni, Silvana Quaglini, Mario Stefanelli, Roberto Gretter, and Daniele Falavigna. 2005. Automated spoken dialogue system for hypertensive patient home management. *International Journal of Medical Informatics* 74, 2 (2005), 159 – 167.

[22] Izzeddin Gur, Semih Yavuz, Yu Su, and Xifeng Yan. 2018. DialSQL: Dialogue Based Structured Query Generation. In *ACL*. 1339–1349.

[23] IBM. 2020. Watson Analytics. https://www.ibm.com/watson-analytics.

[24] Manasa Jammi, Jaydeep Sen, Ashish R. Mittal, Sagar Verma, Vardaan Pahuja, Rema Ananthanarayanan, Pranay Lohia, Hima Karanam, Diptikalyan Saha, and Karthik Sankaranarayanan. 2018. Tooling Framework for Instantiating Natural Language Querying System. *PVLDB* 11, 12 (2018), 2014–2017.

[25] Esther Kaufmann and Abraham Bernstein. 2010. Evaluating the usability of natural language query languages and interfaces to Semantic Web knowledge bases. *J. Web Semant.* 8, 4 (2010), 377–393.

[26] Georgia Koutrika, Alkis Simitsis, and Yannis E. Ioannidis. 2006. Précis: The Essence of a Query Answer. In *ICDE*. 69–78.

[27] D. Küpper, M. Storbel, and D. Rösner. 1993. NAUDA: A Cooperative Natural Language Interface to Relational Databases. In *ACM SIGMOD*.

[28] Chuan Lei, Vasilis Efthymiou, Rebecca Geis, and Fatma Özcan. 2020. Expanding Query Answers on Medical Knowledge Bases. In *EDBT*.

[29] Chuan Lei, Fatma Özcan, Abdul Quamar, Ashish R. Mittal, Jaydeep Sen, Diptikalyan Saha, and Karthik Sankaranarayanan. 2018. Ontology-Based Natural Language Query Interfaces for Data Exploration. *IEEE Data Eng. Bull.* 41, 3 (2018), 52–63.

[30] Fei Li and H. V. Jagadish. 2014. Constructing an Interactive Natural Language Interface for Relational Databases. *PVLDB* 8, 1 (2014), 73–84.

[31] Fei Li and Hosagrahar Visvesvaraya Jagadish. 2014. NaLIR: an interactive natural language interface for querying relational databases. In *SIGMOD*. 709–712.

[32] Fei Li and H. V. Jagadish. 2016. Understanding Natural Language Queries over Relational Databases. *SIGMOD Record* 45, 1 (2016), 6–13.

[33] Yunyao Li and Davood Rafiei. 2017. Natural Language Data Management and Interfaces: Recent Development and Open Challenges. In *SIGMOD*. 1765–1770.

[34] Yunyao Li, Huahai Yang, and H. V. Jagadish. 2005. NaLIX: An Interactive Natural Language Interface for Querying XML. In *ACM SIGMOD*.

[35] Stavros Mallios and Nikolaos G. Bourbakis. 2016. A survey on human machine dialogue systems. In *IISA*. 1–7.

[36] Edward J. McCluskey. 1986. *Logic design principles - with emphasis on testable semicustom circuits.* Prentice Hall.

[37] Michael F. McTear. 2002. Spoken Dialogue Technology: Enabling the Conversational User Interface. *ACM Comput. Surv.* 34, 1 (2002), 90–169.

[38] Microsoft. 2020. Microsoft PowerBI. https://powerbi.microsoft.com/en-us/blog/tag/natural-language/.

[39] Microsoft[US]. 2018. Microsoft Cortana. https://www.microsoft.com/en-us/windows/cortana.

[40] Adam S. Miner, Arnold Milstein, Stephen Schueller, et al. 2016. Smartphone-Based Conversational Agents and Responses to Questions About Mental Health, Interpersonal Violence, and Physical Health. *JAMA Internal Medicine* 176, 5 (2016), 619–625.

[41] Panupong Pasupat and Percy Liang. 2015. Compositional Semantic Parsing on Semi-Structured Tables. In *ACL*. 1470–1480.

[42] Abdul Quamar, Chuan Lei, Dorian Miller, Fatma Özcan, Jeffrey Kreulen, Robert J Moore1, and Vasilis Efthymiou. 2020. An Ontology-Based Conversation System for Knowledge Bases. In *SIGMOD*.

[43] Nicole M. Radziwill and Morgan C. Benton. 2017. Evaluating Quality of Chatbots and Intelligent Conversational Agents. *CoRR* abs/1704.04579 (2017).

[44] Diptikalyan Saha, Avrilia Floratou, Karthik Sankaranarayanan, Umar Farooq Minhas, Ashish R. Mittal, and Fatma Özcan. 2016.

ATHENA: An Ontology-Driven System for Natural Language Querying over Relational Data Stores. *PVLDB* 9, 12 (2016), 1209–1220.
[45] Salesforce. 2020. Einstein Analytics. https://developer.salesforce.com/einstein-analytics.
[46] Jaydeep Sen, Fatma Ozcan, Abdul Quamar, Greg Stager, Ashish R. Mittal, Manasa Jammi, Chuan Lei, Diptikalyan Saha, and Karthik Sankaranarayanan. 2019. Natural Language Querying of Complex Business Intelligence Queries. In *SIGMOD*. 1997–2000.
[47] Alkis Simitsis, Georgia Koutrika, and Yannis E. Ioannidis. 2008. Précis: from unstructured keywords as queries to structured databases as answers. *VLDB J.* 17, 1 (2008), 117–149.
[48] Sisense. 2020. SISENSE : Natural Language Insights. https://www.sisense.com/product/analyze/sisense-narratives/.
[49] Dezhao Song, Frank Schilder, Charese Smiley, Chris Brew, Tom Zielund, Hiroko Bretz, Robert Martin, Chris Dale, John Duprey, Tim Miller, and Johanna Harrison. 2015. TR Discover: A Natural Language Interface for Querying and Analyzing Interlinked Datasets. In *ISWC*. 21–37.
[50] Tableau. 2020. Tableau Cleargraph. https://www.tableau.com/about/blog/2017/8/welcome-tableau-cleargraph-nlp-team-74621.
[51] Sandeep Tata and Guy M. Lohman. 2008. SQAK: Doing More with Keywords. In *ACM SIGMOD*.
[52] Christina Unger, Lorenz Bühmann, Jens Lehmann, Axel-Cyrille Ngonga Ngomo, Daniel Gerber, and Philipp Cimiano. 2012. Template-based question answering over RDF data. In *WWW*. 639–648.
[53] Sebastian Walter, Christina Unger, Philipp Cimiano, and Daniel Bär. 2012. Evaluation of a Layered Approach to Question Answering over Linked Data. In *ISWC*. 362–374.
[54] Ulli Waltinger, Dan Tecuci, Mihaela Olteanu, Vlad Mocanu, and Sean Sullivan. 2013. USI Answers: Natural Language Question Answering Over (Semi-) Structured Industry Data. In *IAAI*.
[55] Wenlu Wang, Yingtao Tian, Hongyu Xiong, Haixun Wang, and Wei-Shinn Ku. 2018. A Transfer-Learnable Natural Language Interface for Databases. *CoRR* abs/1809.02649 (2018).
[56] Nathaniel Weir and Prasetya Utama. 2019. Bootstrapping an End-to-End Natural Language Interface for Databases. In *SIGMOD*. 1862–1864.
[57] Wizdee. 2020. Wizdee: Natural Language BI. http://wizdee.com/home/.
[58] Zhibiao Wu and Martha Stone Palmer. 1994. Verb Semantics and Lexical Selection. In *ACL*. 133–138.
[59] Xiaojun Xu, Chang Liu, and Dawn Song. 2017. SQLNet: Generating Structured Queries From Natural Language Without Reinforcement Learning. *CoRR* abs/1711.04436 (2017).
[60] Steve J. Young, Milica Gasic, Blaise Thomson, and Jason D. Williams. 2013. POMDP-Based Statistical Spoken Dialog Systems: A Review. *Proc. IEEE* 101, 5 (2013), 1160–1179.
[61] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. 2018. Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine* 13, 3 (2018), 55–75.
[62] Tao Yu, Zifan Li, Zilin Zhang, Rui Zhang, and Dragomir R. Radev. 2018. TypeSQL: Knowledge-Based Type-Aware Neural Text-to-SQL Generation. In *NAACL-HLT*. 588–594.
[63] Tao Yu, Rui Zhang, He Yang Er, Suyi Li, Eric Xue, Bo Pang, Xi Victoria Lin, Yi Chern Tan, Tianze Shi, Zihan Li, Youxuan Jiang, Michihiro Yasunaga, Sungrok Shim, Tao Chen, Alexander R. Fabbri, Zifan Li, Luyao Chen, Yuwen Zhang, Shreya Dixit, Vincent Zhang, Caiming Xiong, Richard Socher, Walter S. Lasecki, and Dragomir R. Radev. 2019. CoSQL: A Conversational Text-to-SQL Challenge Towards Cross-Domain Natural Language Interfaces to Databases. *ArXiv* abs/1909.05378 (2019).
[64] Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir R. Radev. 2018. Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task. In *EMNLP*. 3911–3921.
[65] Tao Yu, Rui Zhang, Michihiro Yasunaga, Yi Chern Tan, Xi Victoria Lin, Suyi Li, Heyang Er, Irene Li, Bo Pang, Tao Chen, Emily Ji, Shreya Dixit, David Proctor, Sungrok Shim, Jonathan Kraft, Vincent Zhang, Caiming Xiong, Richard Socher, and Dragomir R. Radev. 2019. SParC: Cross-Domain Semantic Parsing in Context. *CoRR* abs/1906.02285 (2019).
[66] Gideon Zenz, Xuan Zhou, Enrico Minack, Wolf Siberski, and Wolfgang Nejdl. 2009. From keywords to semantic queries - Incremental query construction on the semantic web. *J. Web Semant.* 7, 3 (2009), 166–176.
[67] Rui Zhang, Tao Yu, Heyang Er, Sungrok Shim, Eric Xue, Xi Victoria Lin, Tianze Shi, Caiming Xiong, Richard Socher, and Dragomir Radev. 2019. Editing-Based SQL Query Generation for Cross-Domain Context-Dependent Questions. In *EMNLP-IJCNLP*. 5337–5348.
[68] Weiguo Zheng, Hong Cheng, Lei Zou, Jeffrey Xu Yu, and Kangfei Zhao. 2017. Natural Language Question/Answering: Let Users Talk With The Knowledge Graph. In *CIKM*. 217–226.
[69] Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning. *CoRR* abs/1709.00103 (2017).