State of the Art and Open Challenges in NL Interfaces to Data

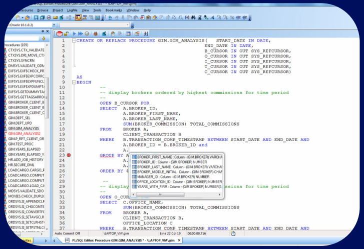
Fatma Özcan, IBM Research – Almaden
Abdul Quamar, IBM Research – Almaden
Jaydeep Sen, IBM Research – India,
Chuan Lei, IBM Research – Almaden
Vasilis Efthymiou, IBM Research - Almaden

Problem

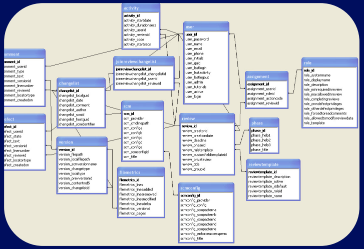
- Natural Language Querying of Complex Datasets

Democratize access to data !!

Easy Access for Business Users



User does need to know SQL or any other complex lang !!



Exact knowledge of underlying data is not required

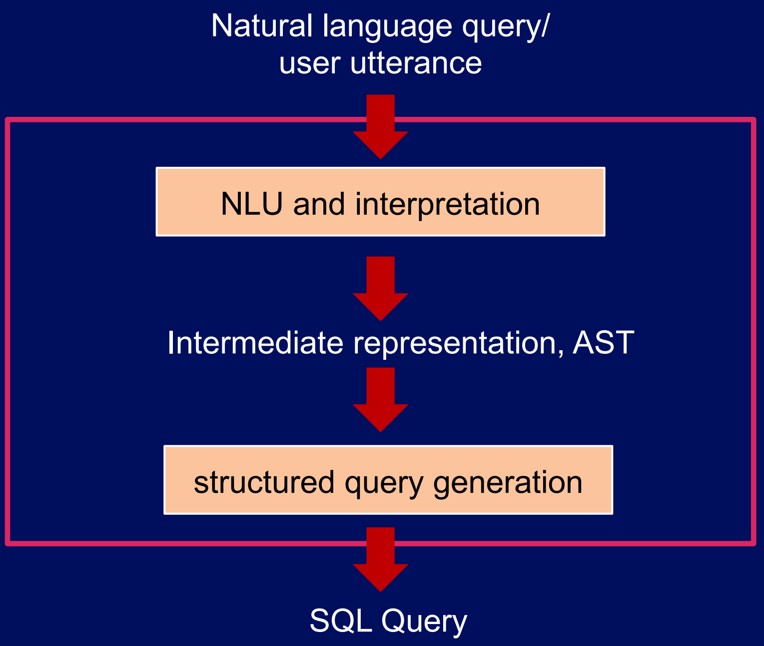
Conversational interfaces

Challenges and Opportunities

- Understanding user intent (disambiguation)
- Converting the intent to target language

- Recent advances in natural language understanding enable more applications
    - Glove, fastText, BERT, …
- Conversational agents also gaining popularity
    - Watson Assistant, SIRI, Cortana, Google Assistant,

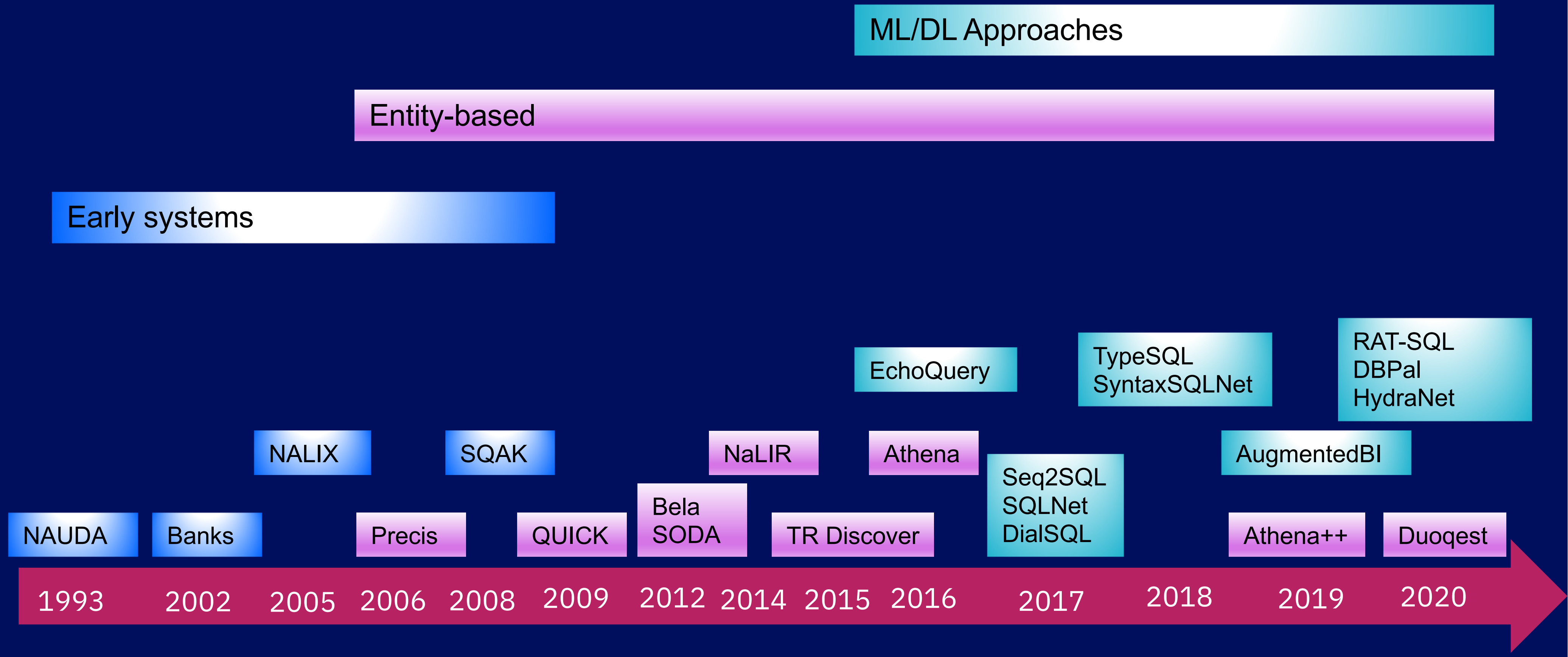
How does it work?



Entity-based:
Interpretation in two steps

ML/DL based:
Holistic, single step

Historical Perspective



What we will cover in this tutorial?

Complexity of the generated queries

- Simple to complex
  - Single table queries
  - Queries with joins between multiple tables
  - Complex queries with subqueries

Interpretation Approaches

- Entity based
- ML/DL based
- Hybrid

Extension to dialogue

- Opportunity for disambiguation via interaction with the user

Complexity

Why complexity?

❖ **Definition**: (target) Query complexity ∝ # of different SQL clauses needed to construct the complete query

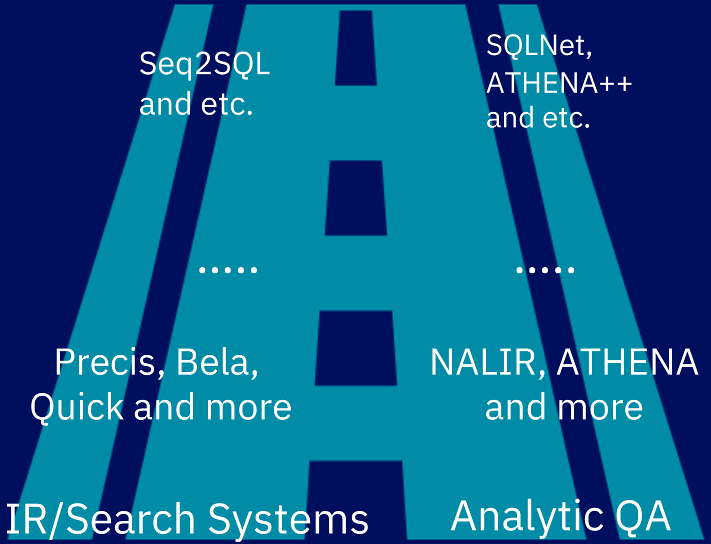
- Defining in terms of query clauses extends to other query languages as well
- Why not define complexity for NL queries .
  - NL query often is highly ambiguous and NLU is still an AI-Hard problem [Yampolskiy, R.V. 2013]

❖ **Complexity is often Application Specific**

- IR or Search on Single Table
  - What is the Capital of France
- Analytic Question Answering Systems over Database Schema
  - What is the average income per state in France

❖ The set of challenges differ depending on what type of queries are being needed and supported.

❖ Complexity of queries influences solution paradigms used in NLIDB systems



Target roadmap in terms of complexity

complexity

Select-Project → Select-Project-Aggregation → Select-Project-Aggregation-GroupBy-OrderBy

Single Table

Select-Project-Join → Select-Project-Join-Aggregation → Select-Project-Join-Aggregation-GroupBy-OrderBy

Multiple Tables

Comparison, Date, Simple, Aggregation → Top-k Queries, L2L Compare → Nested Queries → Window Aggregation, OLAP, etc.

BI and Analytic Queries

Complexity

An example schema

Loan

id	amount	date	decision	for
1	14500	01-04-2018	approved	123
2	25000	11-02-2019	disapproved	345
3	10000	20-12-2019	approved	789
..	..	..	..	..

Borrower

ssncode	name	Credit score	birth date	Lives in
123	Paul Smith	510	02-20-1970	987
345	John Doe	760	04-12-1982	654
..	..	..	..	..

Address

id	street	zipcode	state
987	streetA	100101	California
345	streetB	102102	Texas
..	..	..	..

Simple Select-project queries on single table

Borrower

ssncode	name	credit score	birthdate	lives in
12345	Paul Smith	510	02-20-1970	9876

- Example questions
  - *Show me the* ***ssncode*** *for* ***Paul Smith***
- Challenges
  - Domain understanding
    - What column names or/and data instances have been mentioned in the query

      => "ssncode", Paul Smith (=borrower.name)

    ⚠ [What if ?] Show me ssn for Mr. Smith.
  - NLU
    - What column is supposed to be used for Projection, Filter

    => SELECT(borrower.ssncode)

    ⚠ [What if ?] For Paul Smith, tell me the ssncode?

Select-project-aggregation queries on a single table

- Example NLQs:
  - *What is the average amount of loans approved by year*

- Challenges
  - NLU
    - Is there an aggregation ? What is the argument of aggregation?

      ⇒ AVG(amount)

      ⚠ [What if ?] on average what is the amount of loans approved for each year
    - Is there a group by/order by? What are the arguments for each?
      - Group by(year(date))

  - Domain understanding
    - Are the arguments of Aggregation/Group by/Order By etc. semantically valid?

      ⚠ [What if ?] What is the average approved amount of loans by address

Loan

id	amount	date	status	for
1	14500	01-04-2018	approved	123
2	25000	11-02-2019	disapproved	345
3	10000	20-12-2019	approved	789

Business intelligence queries

Comparison-based filters

SIMPLE comparison: *show me people with credit score more than 500*

TIME dimension: *show me average loans in Q1 2019*

AGGREGATE comparison: *show me people with total loans more than 50,000*

Top-k queries: *show me top 5 zip codes in terms of maximum loan approved*

Like-to-Like comparison: *how does the total amount of approved loans in Q1 this year compare to last year*

Window Aggregation: what is the moving average of approved loan amount *in every consecutive 3 months of last year*

OLAP: *which of the zipcodes had an increase in average amount of loan approved by more than 20%*

Select-project-aggregation-join on multiple tables

- Example NLQs
  - *What is the average amount of loans approved by zipcode*

- Challenges
  - NLU
    - What are all the tables mentioned i.e., candidates for FROM clause?

    => LOANS ADDRESS (any implicit mention of intermediate tables?)

  - Domain understanding
    - How to join the tables needed to answer a user query?

      => LOANS INNER JOIN BORROWER INNER JOIN ADDRESS

    - Needs to know the complete domain schema with relations
    - For large schema, finding the right join path is the main challenge

Loans

id	amount	date	status	for
..	..	..	..	..

Borrower

ssncode	name	credit score	birth date	lives in
..	..	..	..	..

Address

id	street	zipcode	state
..	..	..	..

IBM

Business intelligence queries - challenges

- Complex computations
  - Which of the zipcodes had an increase in average amount of loan approved by more than 20%
- Implicit intents/arguments
  - Time dimension
    Show me average loans in Q1 2019 => (loan date) in Q1 2019
    Domain reasoning
    Who are the top 10 borrowers in zipcode 12345 => Top 10 borrowers (in terms of total amount of loans)

- Solving these challenges require a combination of rich NL understanding as well as domain reasoning
  - NLU: how to detect the primary intent or/and mention of computations in the query
  - Domain reasoning: How to to infer the implicit arguments and/or resolve ambiguity in the NL utterance

Nested queries

Examples:

1. Applying NOT operation to obtain complement set
2. Numeric Comparison between subqueries.
3. Enforcing Equality/inequality between subqueries

(.. and more categories....)

- *Show me zipcodes that has* ***no borrowers*** *with credit score more than 600.*
- *Find all borrowers with* ***more*** *loans in* ***this year than last year*** *?*
- *Who had an approved and rejected loan in the* ***same year*** *?*
- (and more examples...)

Subqueries:

- subquery formation: how to segregate the NL query into subquery parts ?
- *Find all borrowers with* ***more*** *loans in* ***this year than last year*** *?*
  => {borrowers, loans, this year} > {borrowers, loans, last year}

Challenges:

NLU
- hard to detect : Many reasons why *a NL query may require a nested query.*

- Domain Understanding
  - subquery formation: Needs to reason over domain semantics and query context.

Key takeaways

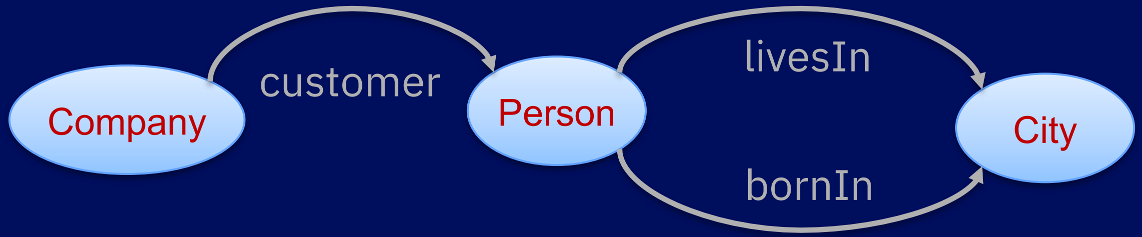
- ❖ Complexity is correlated with the application need
  - QA systems aimed for analytic queries over DB schema needs complex queries
- ❖ Multiple table queries needs the knowledge of full domain schema
- ❖ More complex queries in general need
  - Deeper domain understanding
  - Ability to reason over query intent and domain semantics.
- ❖ Implications
  - ML models → adequate domain specific training examples
  - Entity based models → domain abstraction (ontology graph, schema graph, etc.)

NLQ Interpretation: Entity-based Approaches

Entity-based Approaches

Recognize entities and relationships between the entities in a query

Example: *"Show me Amazon customers who are also from Seattle"*



Internal/external representation of the underlying data using:

- an index structure (e.g., inverted index over tables and columns)
- a taxonomy of terms and their synonyms (e.g., WordNet)
- an ontology (i.e., a rich semantic data model allowing complex query interpretation)

Using Index Structures or Taxonomies

Common approach:

- Parsing of the NLQ to machine-readable format
- Identify slots in NLQ that correspond to entities
- Look up entity slots in NLQ in an inverted index of labels

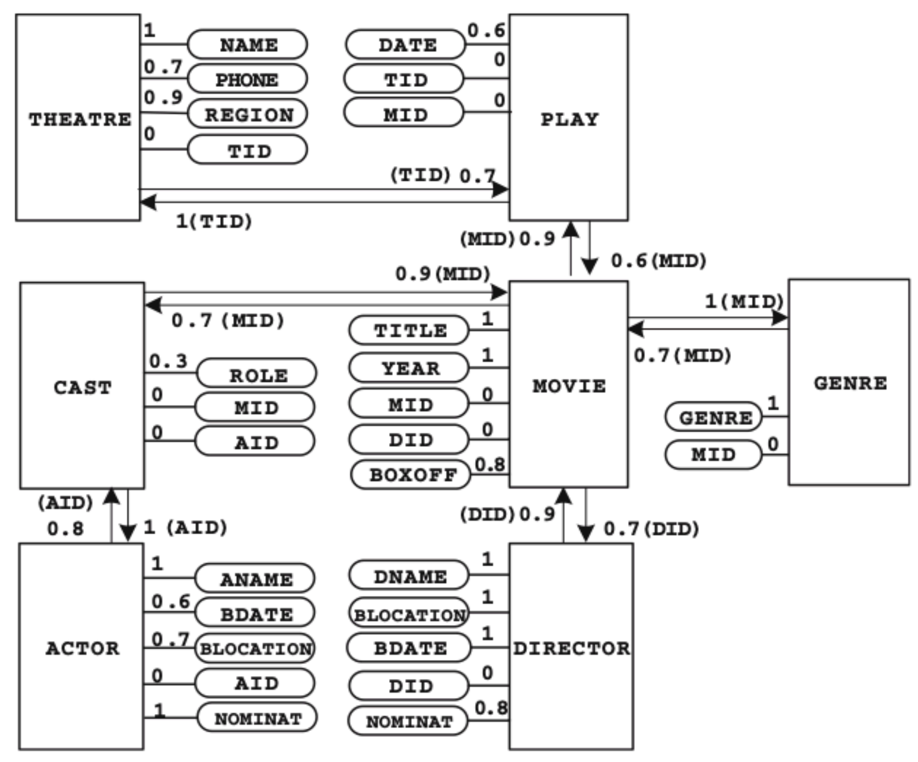
Example: "*Show me Amazon customers who are also from Seattle*"

Précis [Koutrika et al., ICDE 06] [Simitsis et al., VLDBJ 08]
QUICK [Zenz et al, J. Web Semantics 09]
DUOQUEST [Baik et al., CIDR 20] [Baik et al., SIGMOD 20]
NaLIR [Li et al., SIGMOD 14][Li et al., VLDB 14][Li et al., SIGMOD Rec. 16]

Research 26, Thursday

Précis [Koutrika et al., ICDE 06] [Simitsis et al., VLDBJ 08]

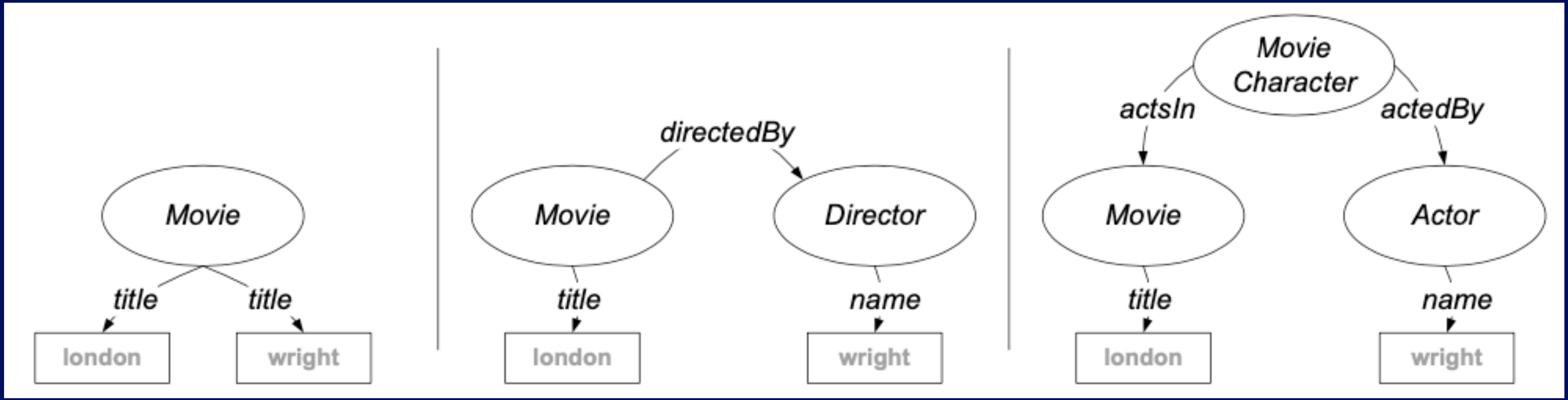
Example (keyword search in DNF):
"Clint Eastwood" AND "thriller"



- Interpretations:
  - thrillers directed by Clint Eastwood
  - thrillers in which Clint Eastwood is acting
  - thrillers directed by Clint Eastwood, in which Clint Eastwood is also acting
- Interpretations ranked based on join importance

QUICK [Zenz et al, J. Web Semantics 09]

Example (keyword search): “Wright London”

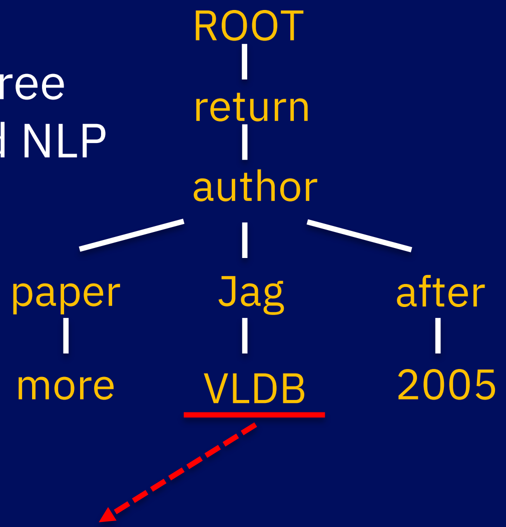


- ➢ User interaction to determine which interpretation is correct

NaLIR [Li et al., SIGMOD 14][Li et al., VLDB 14][Li et al., SIGMOD Rec. 16]

Example: “show all authors who have more papers than H. V. Jagadish in VLDB after 2005”

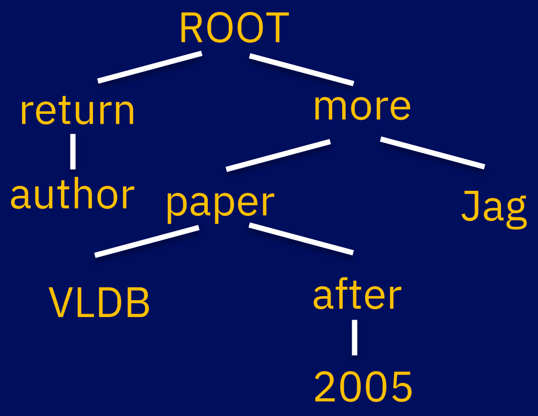
Initial parse tree from Stanford NLP



may refer to WordNet terms:
VLDB conference, PVLDB, and VLDB Journal

User interaction to disambiguate

Refined parse tree



User may further edit the refined parse tree (e.g., add new nodes)

Using an Ontology

Common approach:

- Look up entity slots in NLQ in an ontology
- Identify possible join paths based on the underlying ontology relationships

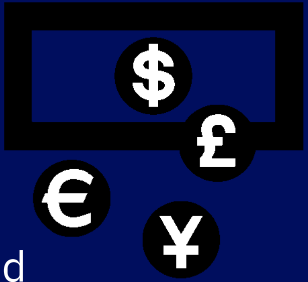
BELA [Walter et al., ISWC 12]
SODA [Blunschi et al., VLDB 12]
USI Answers [Waltinger et al., IAAI 13]
TR Discover [Song et al., ISWC 15]
ATHENA [Saha et al., VLDB 16][Lei et al., IEEE Data Eng. Bull. 18]
ATHENA++ [Sen et al., SIGMOD 19]

BELA [Walter et al., ISWC 12]

Example: “What is the currency of the Czech Republic?”

- Query templates:
  SELECT ?y WHERE {?x ?p ?y}
  slots: (?x: Czech Republic), (?p: currency)

- Inverted index lookup, built from DBpedia labels:
  - lookup result for ?x: *http://dbpedia.org/resource/Czech_Republic*
  - lookup result for ?p: *http://dbpedia.org/ontology/currency*
    - if no exact match, the closest property of Czech Republic to “currency” is returned



- Interpretation:
  - fill slots with lookup results:
    SELECT ?y WHERE {dbr:Czech_Republic ?dbonto:currency ?y}

SODA [Blunschi et al., VLDB 12]

- Looks up each query keyword in two indices:
  - one for the data in the database
  - one for the meta-data in ontologies (so-called *metadata warehouse*)
    - including synonyms and homonyms extracted from DBpedia

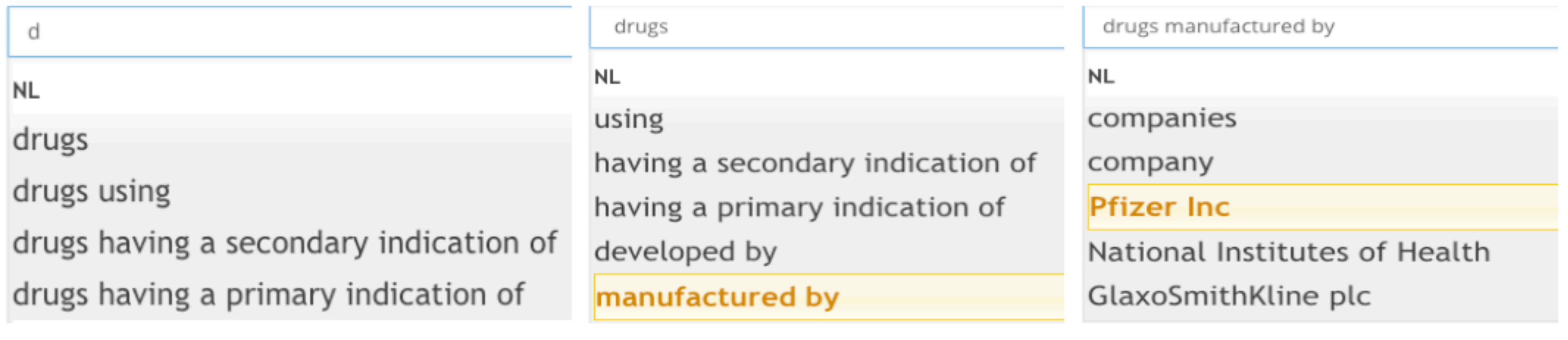
- Multiple interpretations generated
  - ontology hierarchies and relationships help in disambiguation
- Ranking of interpretations based on lookup scores aggregations
- Top-10 interpretations executed, and snippets are shown to user to select

USI Answers [Waltinger et al., IAAI 13]

- Parse query using Stanford Core NLP and ClearTK
- Dictionary- and regex-based look-ups to generate candidates
- Distinguish between concepts, instances, relationships, and identify time mentions
  - a dedicated annotator is used for each of the above components
- Data stored in a relational DB, meta-data represented in an ontology
  - allows relationship extraction between ontology concepts

TR Discover [Song et al., ISWC 15]

- ➢ Provides query auto-completion
  - ➢ suggestions based on nodes centrality in RDF graph



”d” is typed

”drugs” is selected and suggestions are provided

- properties having “Drugs” as subject in RDF graph

“manufactured by” is selected and “Pfizer Inc” can be chosen to complete the query

ATHENA [Saha et al., VLDB 16][Lei et al., IEEE Data Eng. Bull. 18]

- Two-phase approach (physical-logical independence):
  - Phase 1: Query interpretation against a domain ontology
  - Phase 2: Structured query generation

Example: “How many people bought IBM stocks in the last 5 years?”

- Annotate each token with possible ontology elements (e.g., Company.name or ListedSecurity.legalName for “IBM” token)
- Selecting all combinations of candidate elements per token gives different interpretations
- Model every possible interpretation as an Interpretation Tree (ITree)
- Pick a single element for each token in a holistic way (Steiner Tree-based)

ATHENA++ [Sen et al., SIGMOD 19]

- Extends ATHENA to cover complex nested queries (commonly found in BI queries)

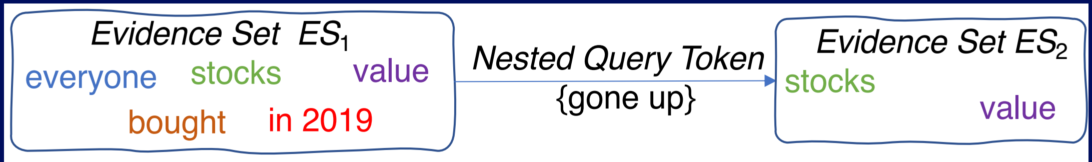
Transaction.type Transaction.time MonetaryAmount.value

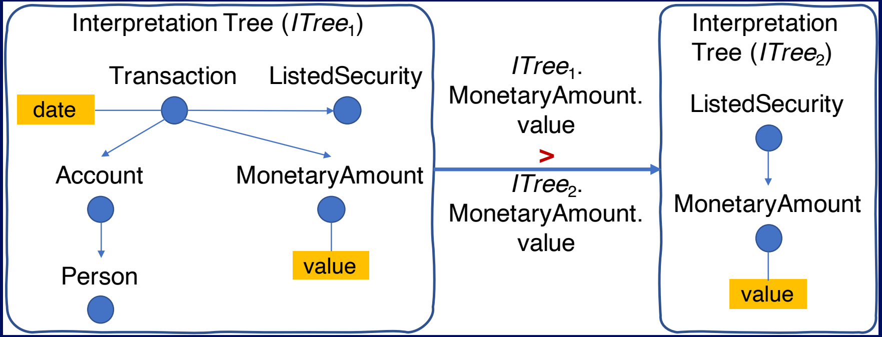
Example: "Show me everyone who bought stocks in 2019 that have gone up in value"

Person, Customer, Account Manager

ListedSecurity

Operator: '>'





Pros and Cons of Entity-based Approaches

Handling complex input queries and generating complex structured queries

Easier to incorporate domain knowledge

Usually don’t require labelled training data

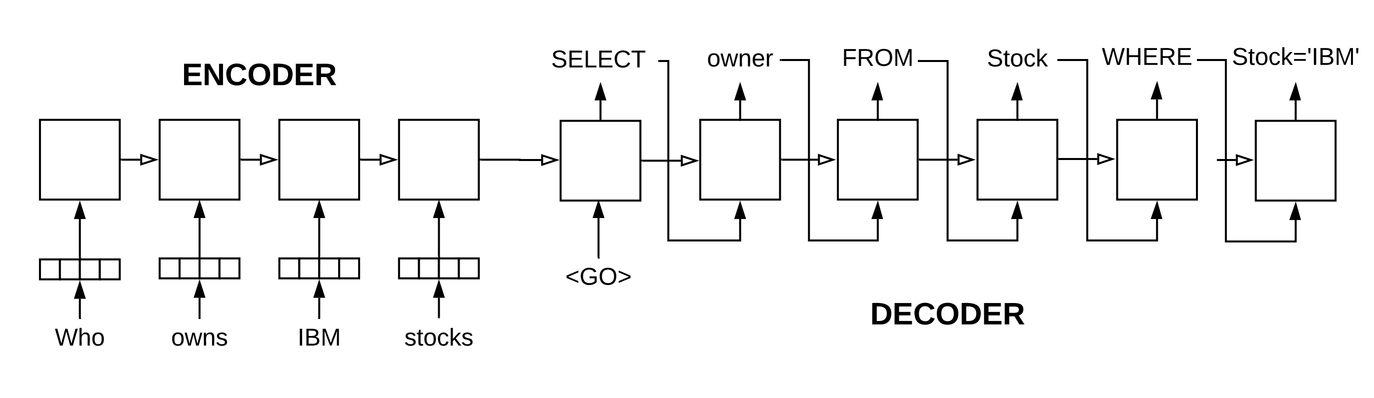
Highly sensitive to variations in the user query

NLQ Interpretation:
Machine Learning-based Approaches

Machine learning-based approaches

– General idea

- Apply supervised machine learning techniques (RNNs) on a set of question/answer pairs
  – Questions: natural language queries
  – Answers: respective SQL statements



Machine learning-based approaches: progression

Seq2Seq (RNN)

- ***Seq2SQL***
- ***SQLNet***
- ***TypeSQL***
- *IncSQL*
- *SQLova*
- *X-SQL*
- *HydraNet*

Domain Adaptation

- ***SyntaxSQLNet***
- *IRNet*
- *RYANSQL*
- ***Adversarial*** *method*
- ***RAT-SQL***

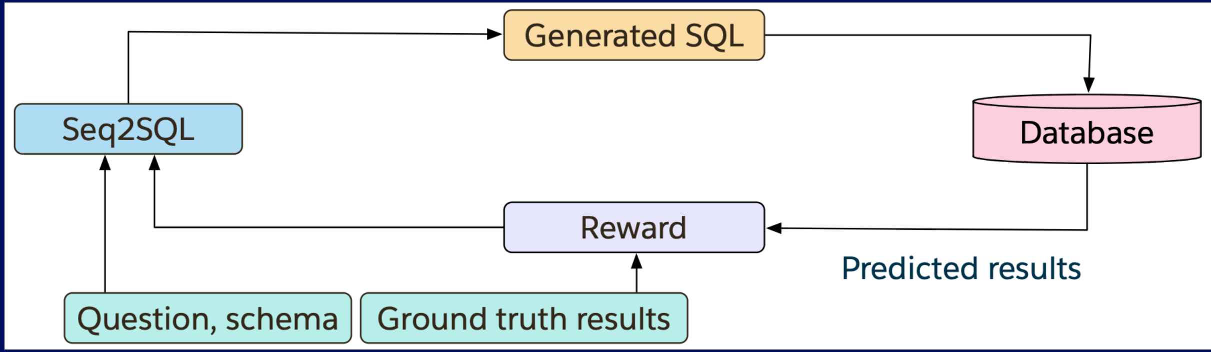
Training Data Generation

- ***DBPal***

Seq2SQL [Zhong et al, arXiv 2017]

– Key ideas

- A deep neural network leverages SQL structure to prune generated query space
- Policy-based reinforcement learning (RL) to generate query conditions
- A mixed object (cross entropy losses + RL rewards from in-the-loop query execution)

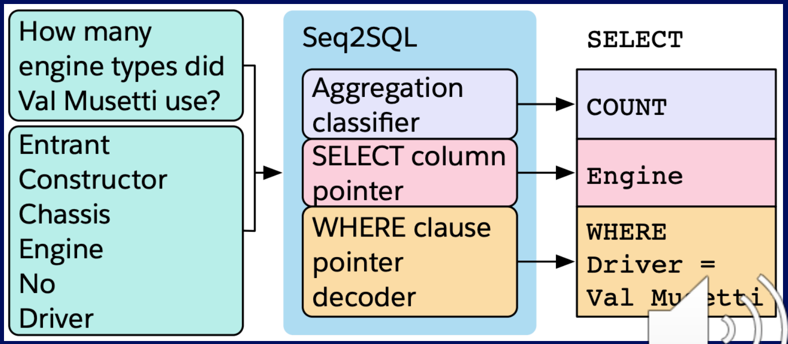


Seq2SQL cont.

– Aggregation operation
  - An MLP over aggregated hidden representations of the inputs
  - 4 possible outputs: COUNT, MIN, MAX, or NONE

– Select column
  - A list of column representations using LSTM + a question representation (similar to aggregation operation)
  - Combine two representations as input for an MLP

– Where clause
  - Augmented pointer network and RL



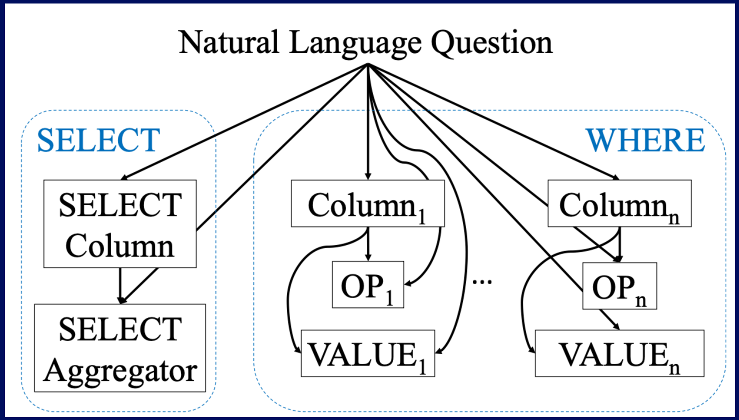
SQLNet [Xu et al. arXiv 2017]

– Key ideas

- Sketch-based approach to avoid RL and “order-matters” issue
- Sequence-to-set prediction using column attention (WHERE clause)
  – An MLP with one layer over the embeddings computed by 2 LSTMs (one for the question, one for the column names)

SQL sketch

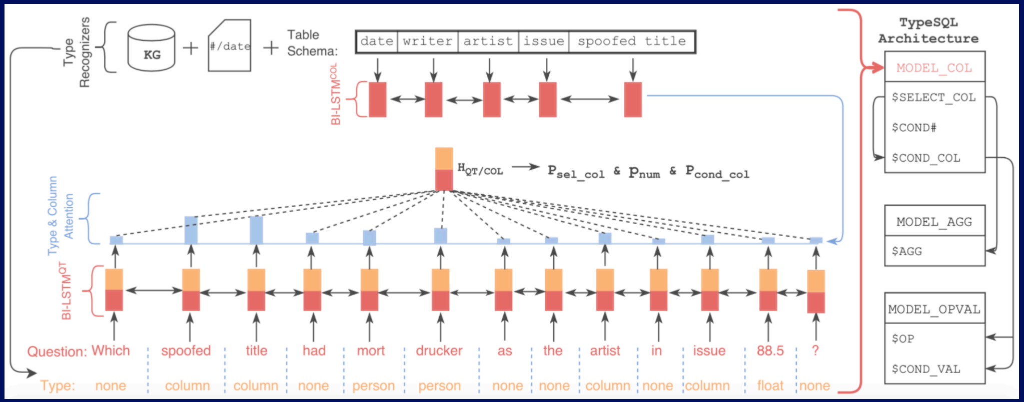
```
SELECT $AGG $COLUMN
WHERE $COLUMN $OP $VALUE
(AND $COLUMN $OP $VALUE)*
```



TypeSQL [Yu et al. NAACL 2018]

– Key ideas

- Sketch-based approach to fill query slots
- Utilize types extracted from either knowledge graph or table content to help model better understand entities and numbers in the question
  - Two bi-directional LSTMs to encode words in the question with their types and the column names separately
  - The output hidden states of LSTMs are then used to predict the values for the slots in the SQL sketch

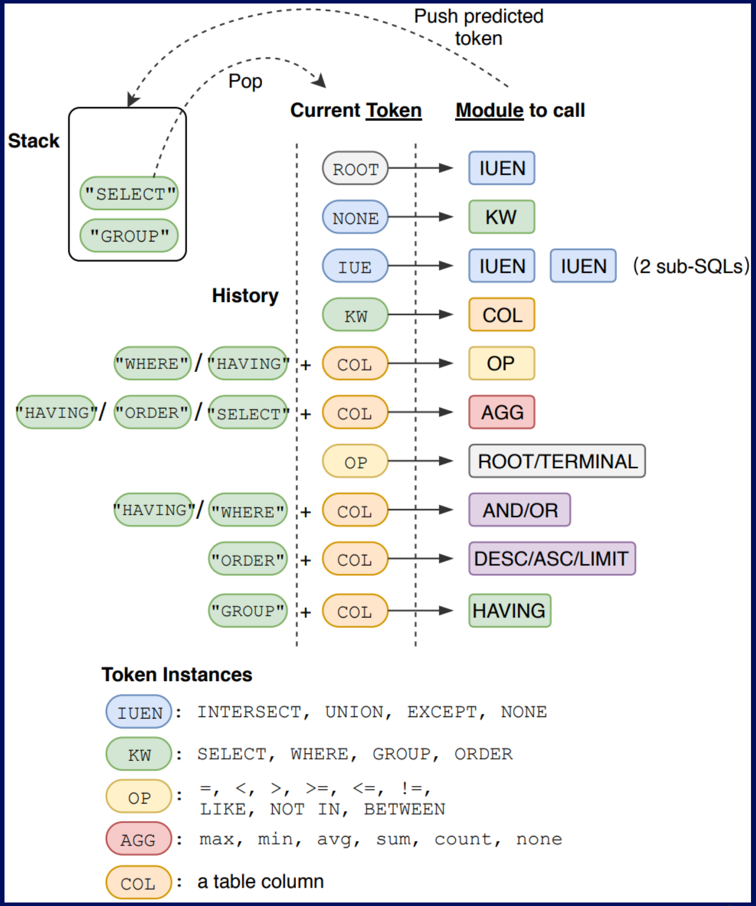


Machine Learning-based Approach

SyntaxSQLNet [Yu et al. EMNLP 2018]

– Key ideas

- SQL path history and table-aware column attention encoders
  - Attention mechanism to encode question representation as well as SQL path history
- SQL specific syntax tree-based decoder with SQL path history
  - Determine a specific module to invoke and predict the next SQL token to generate based on the current SQL token and the tokens gone over to reach the current token



Adversarial method for domain adaptation [Wang et al. ICDE 2020]

– Key ideas

- Separate out data-specific components and focus on the latent semantic structure
- Domain-specific knowledge will NOT be a strong signal for prediction

– Example

What is the ~~height~~ c0 of ~~LeBron James~~ v1?

```
SELECT c0 WHERE c1 = v1
```

```
SELECT height WHERE name =
'LeBron James'
```

answer

What is the ~~population~~ c0 of ~~NYC~~ v1?

```
SELECT c0 WHERE c1 = v1
```

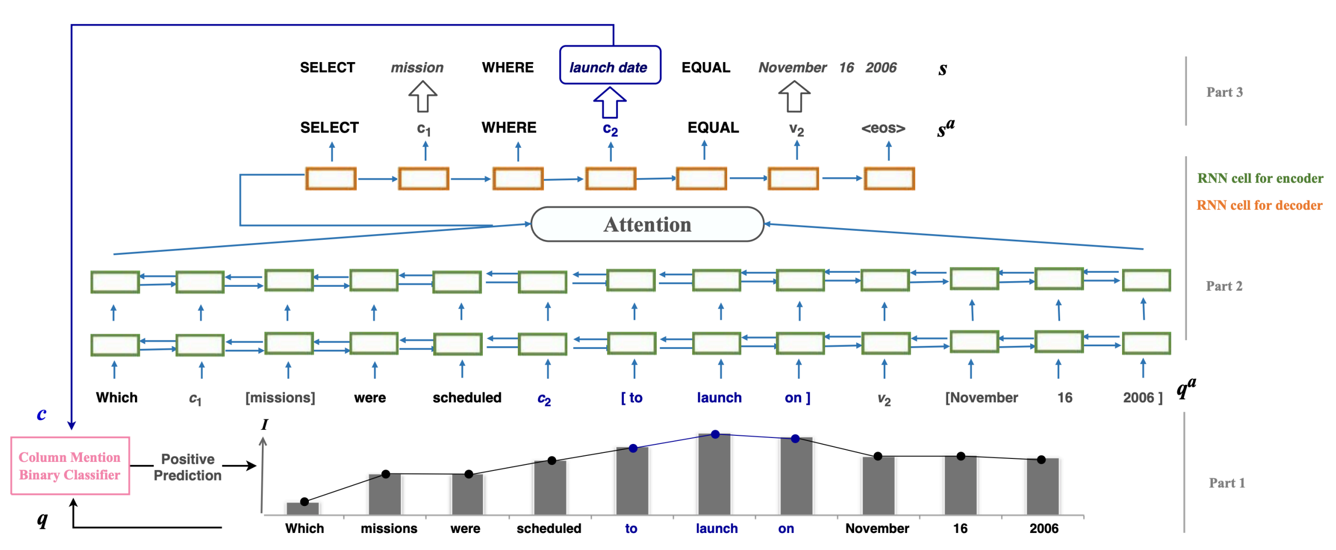
```
SELECT population WHERE city =
'NYC'
```

Adversarial method for domain adaptation cont.

– Phrase that mentions "to launch on" should be the most influential to the prediction using Fast Gradient Method

1. Classifier predicts if domain-specific keyword is mentioned

2. Identify terms by searching for a continuous span that changes the prediction the most



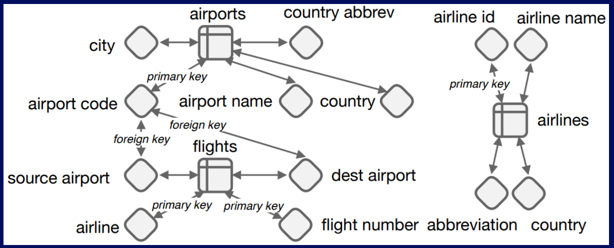
Machine Learning-based Approach

RAT-SQL [Wang et al. ACL 2020]

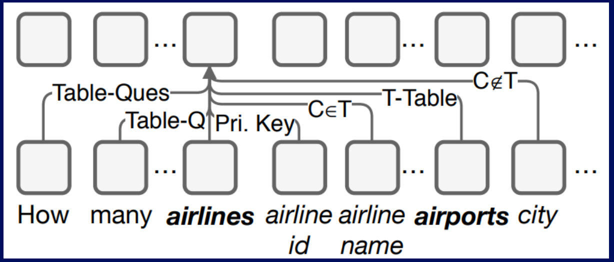
– Key ideas

- Relation-aware self-attention
  - Schema entities and question words
  - Predefined schema relations
- Represent database schema and the question-contextualized schema as graph
  - Schema linking
    - » Name-based and value-based linking
    - » Memory-schema alignment matrix

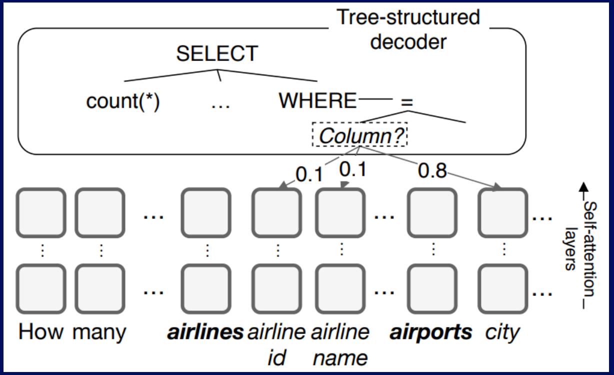
Schema Graph



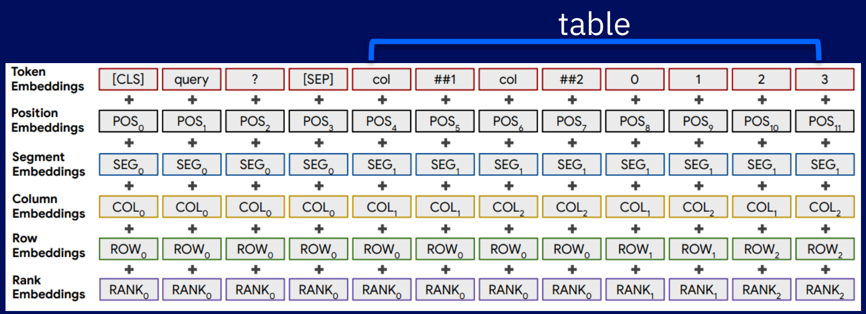
RAT Layer



Tree Decoder

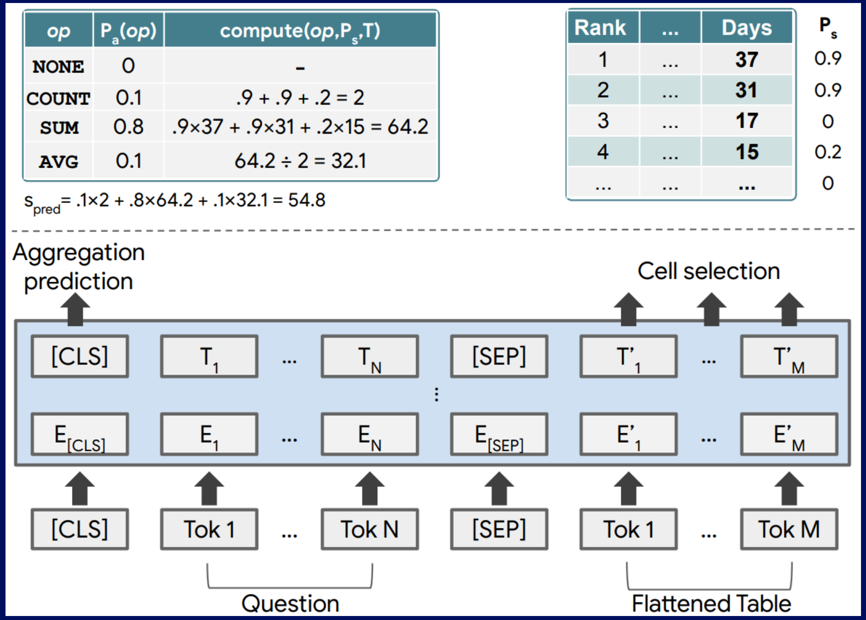


TAPAS [Herzig et al. ACL 2020]



– Key ideas

- Extend BERT's architecture to pre-train the model over tables and related text segments
  - Additional positional embeddings used to encode tabular structure
- Weak supervision reasons over tables without generating logical forms
  - Predict the denotation by selecting table cells
  - Optionally apply aggregation operator to such selection



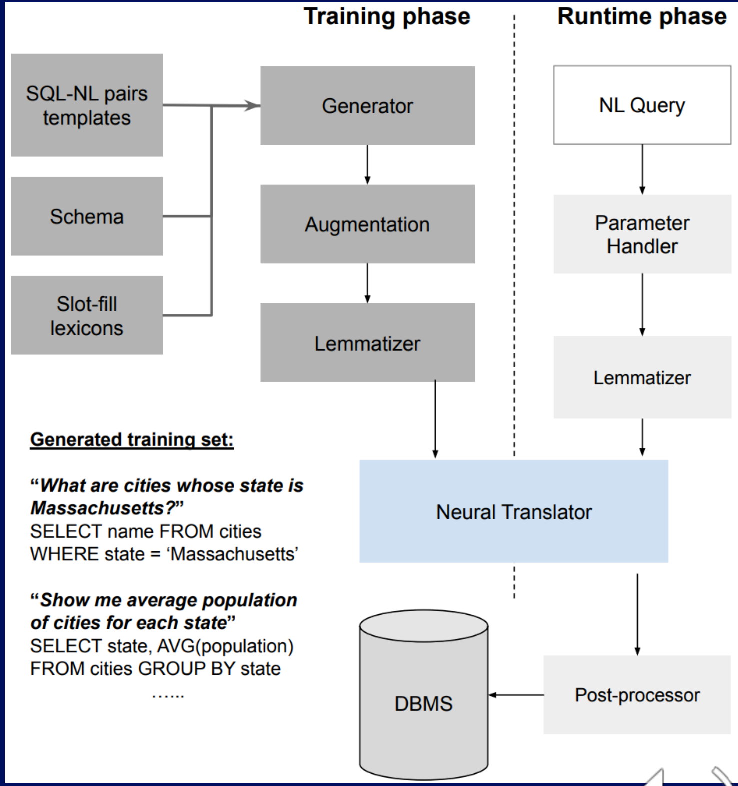
Based on material from: Herzig et al. 2020. TAPAS: Weakly Supervised Table Parsing via Pre-training. *ACL*.

DBPal [Weir et al. SIGMOD 2020]

– Key idea – generates synthetic training data

Improve overall translation accuracy
Increase robustness to linguistic variations
Specialize the model for the target database

- Training phase
  – Provide large corpora of synthesized training data
- Runtime phase
  – Replace the constants in the input NL query with placeholders to make the translation model independent from the actual database



DBPal cont.

– Training phase

- Data instantiation
  – Each SQL template <-> one or more NL templates (slot filling)
  SQL template – `Select {Attribute}(s) From {Table} Where {Filter}`
  NL template – `{SelectPhrase} {Attribute}(s) {FromPhrase} {Table}(s) {WherePhrase} {Filter}`
- Data augmentation
  – Automatic paraphrasing (using PPDB)
  – Missing information (drop ping words and subphrases)
- Optimization procedure

Data Instantiation	
$size_{slotfills}$	Maximum number of instances created for a NL-SQL template pair using slot-filling dictionaries.
$size_{tables}$	Maximum number of tables supported in join queries.
$groupBy_p$	Probabilities of generating a *GROUP BY* version of a generated query pair.
$join_{boost}$, agg_{boost}, $nest_{boost}$	Control the balance of various types of SQL statements relative to each other and the number of templates used.

Data Augmentation	
$size_{para}$	Maximum size of subclauses that are automatically replaced by a paraphrase.
num_{para}	Maximum number of paraphrases that are used to vary a subclause.
$num_{missing}$	Maximum number of words that are removed for a given input NL query.
$randDrop_p$	Probability of randomly dropping words from a generated query.

Machine learning-based approach takeaways

– Pros

- Robust to natural language variations
- Easy instantiation

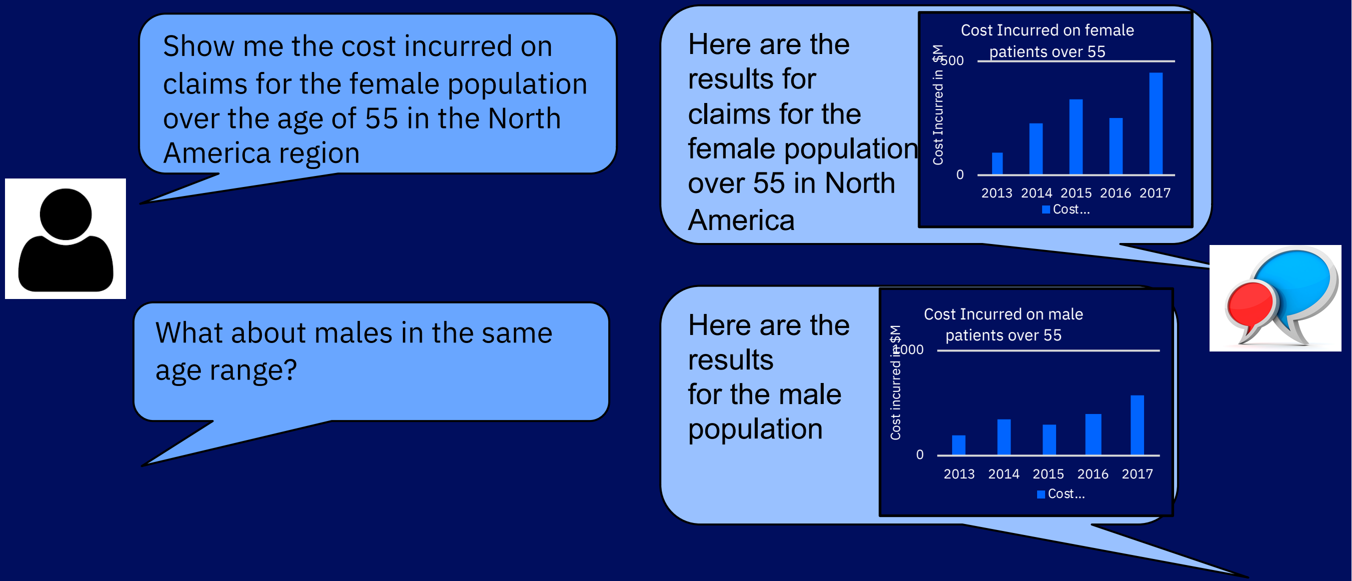
– Cons

- Limited capability of handling complex queries
- Require large amounts of training data

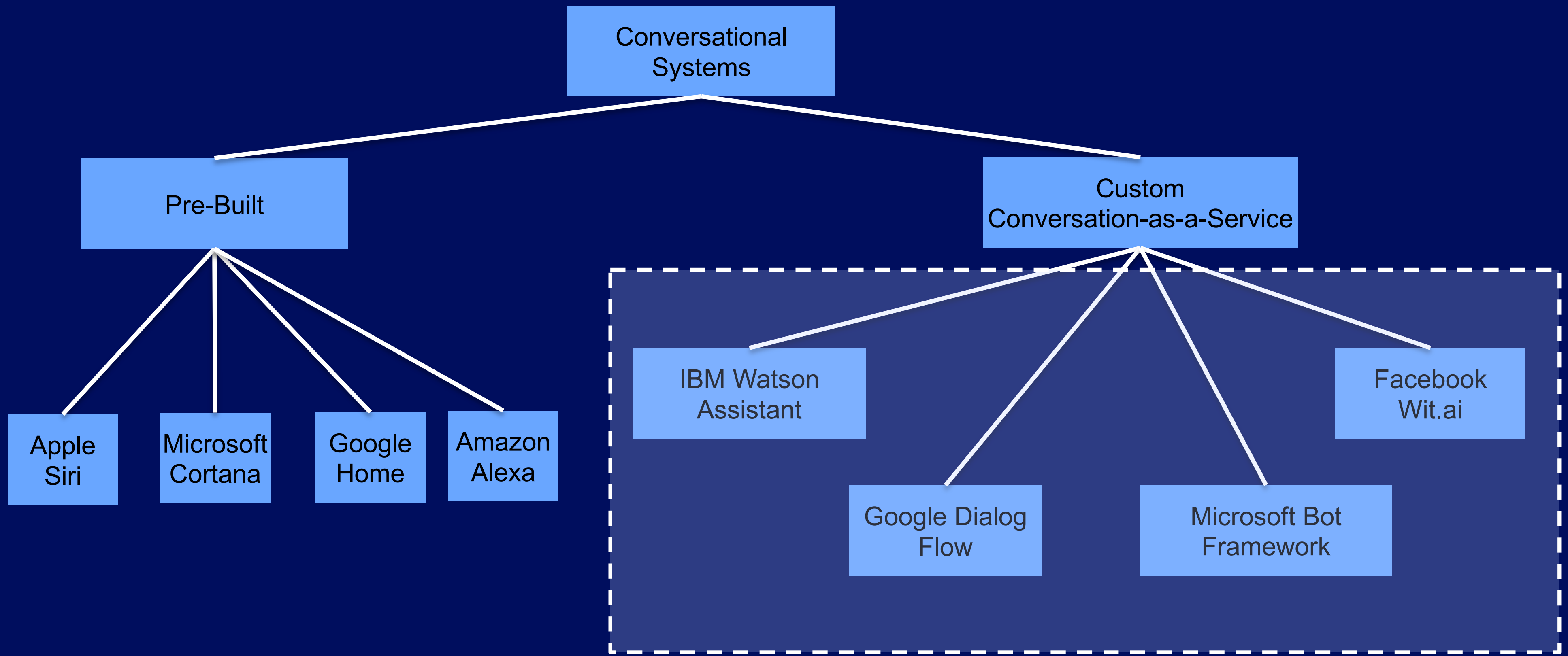
Extension to Dialog

Dialog as an extension to one-shot Q&A

- Next natural step in NLID is a dialog
  - Ability to understand, respond and clarify ambiguity using a two-way conversation
  - Persistent context across turns of conversation
  - Interactive experience for data exploration



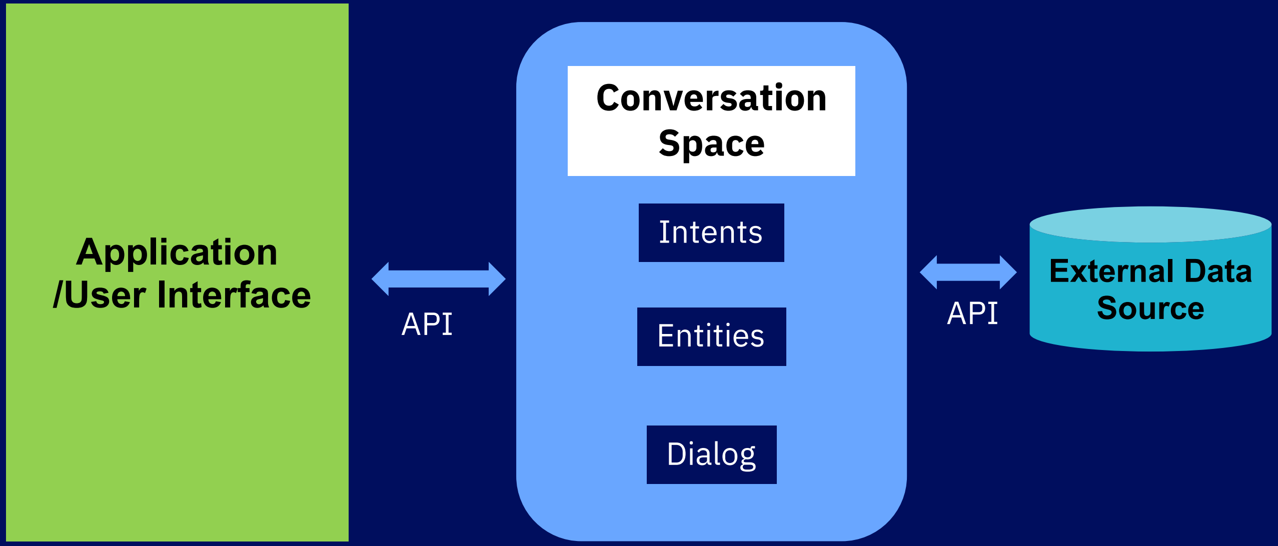
Taxonomy of Conversation Systems



Based on material from J. Gao, M. Galley, and L. Li. Neural approaches to conversational AI. CoRR, abs/1809.08267, 2018.

Extension to Dialog

Components of a Conversation System



Intents:

- Intents express the purpose or goal expressed in the user query/input

Entities:

- Represent real world objects relevant in the context of a user query

Dialog:

- Uses discovered intents, entities and context from the application to provide an interactive conversational experience to the user

External Data Sources:

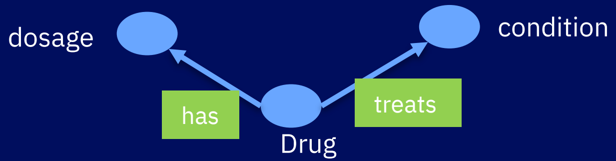
- Interaction with external data sources needs to be orchestrated to respond to user/application queries

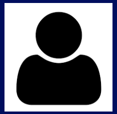
Extension to Dialog

Intent Identification

- Intent Specification
  - Need for up-front specification of a fixed set of intents based on
    - What the users might want to ask(Expected Workload)
    - What the chatbot is designed to handle/support

- Approaches for Intent Classification
  - ML Classifiers
  - Deep Learning Techniques
    - Seq2Seq Networks
      - Translate a natural language query into SQL
      - [SEQ2SQL, SQLNet, …]
    - Utilize user feedback
      - Dial SQL [Gur et al. ACL 2018]
      - Echo Query [Gabriel Lyons et al. SIGMOD 2016]
    - Utilize conversational context
      - Editing based SQL Query generation [Zhang et al. EMNLP 2019]
  - Utilize user feedback and context
      - A-BI, [Francia et al., EDBT 2019]





- Does Anthralin treat Psoriasis?
- What Drugs treat Psoriasis?

Intent : Treatment

- What is the dosage of Anthralin for children?
- What is the pediatric dosage for Anthralin?

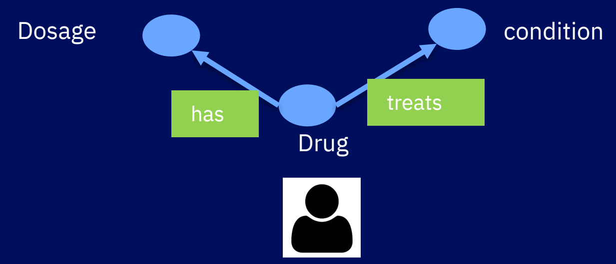
Intent : Dosage

Extension to Dialog

Intent Identification: ML Classifiers

Follow a two-step approach

1. Classify user utterances into a set of predefined intents
2. Structured Query Generation
   - A structured query generated corresponding to each identified intent
   - Template based query generation a common approach
   - One template corresponding to each intent
   - Templates populated using the entities identified in the user utterance to generate structured query



What Drugs treat Psoriasis?

Intent identification

Intent : Treatment

Structured Query Template

```
SELECT oDrug.name
 FROM Drug oDrug INNER JOIN Condition oCondition
 WHERE oDrug.treats=oCondition.ConditionID
        AND oCondition.name = '<@Condition>'
```

Populate template with extracted entities

Structured Query (SQL)

```
SELECT oDrug.name
 FROM Drug oDrug INNER JOIN Condition oCondition
 WHERE oDrug.treats=oCondition.ConditionID
        AND oCondition.name = 'Psoriasis'
```

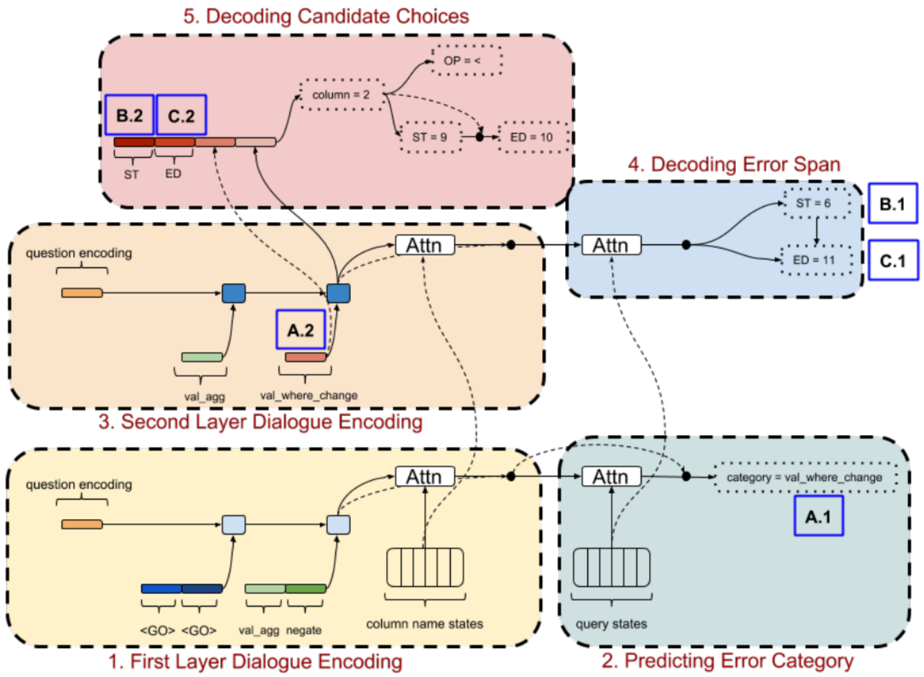
Intent Identification: Utilizing Feedback

- Iterative query generation-Dial SQL [Gur et al. ACL 2018]:
  - Iterative dialog-based query generation framework using user feedback
  - Ask users for validating via simple multi-choice questions interface
  - User feedback is then leveraged to revise the query

- Multi-Layer RNN network to encode dialogue history and provide candidate query choices to users
  - First layer encodes dialog history
  - Second layer decodes error span
  - Third layer decodes list of choices to offer to user

Natural Language Question :	What are the countries that joined the NATO before 2004 ?
Ground Truth SQL Query :	SELECT country WHERE date of join < 2004
Initial SQL Query :	SELECT count (country) WHERE date of join = 2004

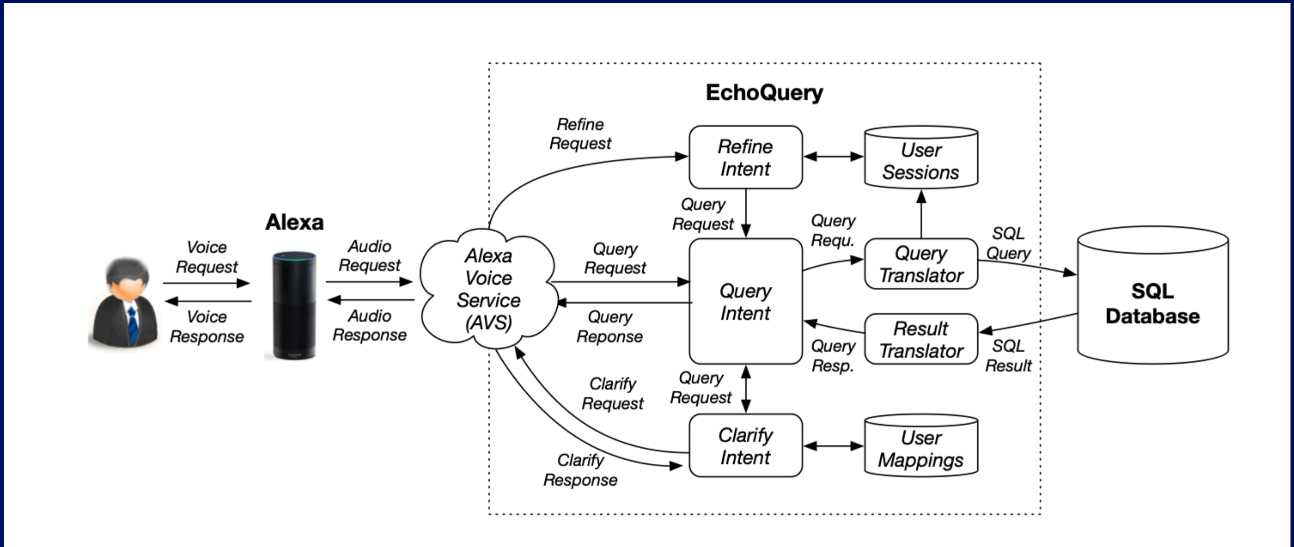
Turn Side	**NL Expression using Rule-Based Templates**	**Response**
System:	Could you please confirm if you are asking **total number of the results**?	validate_agg (1, 2)
User:	No, that's not true.	negate()
System:	Could you please select one of the following options : (i) **average number of results**, (ii) **all the results**?	offer_agg (average, no_agg)
User:	It should be **all the results**.	select(2)
System:	Could you please confirm if **date of join is 2004** ?	validate_where_change(6, 11)
User:	That's wrong.	negate()
System:	Could you please select one of the following options : (i) **date of join is before 2004**, (ii) **date of invitation is before 2004**?	offer_where (date of join < 2004, date of invitation < 2004)
User:	**Date of join is before 2004**.	select(1)



Intent Identification: Utilizing Feedback

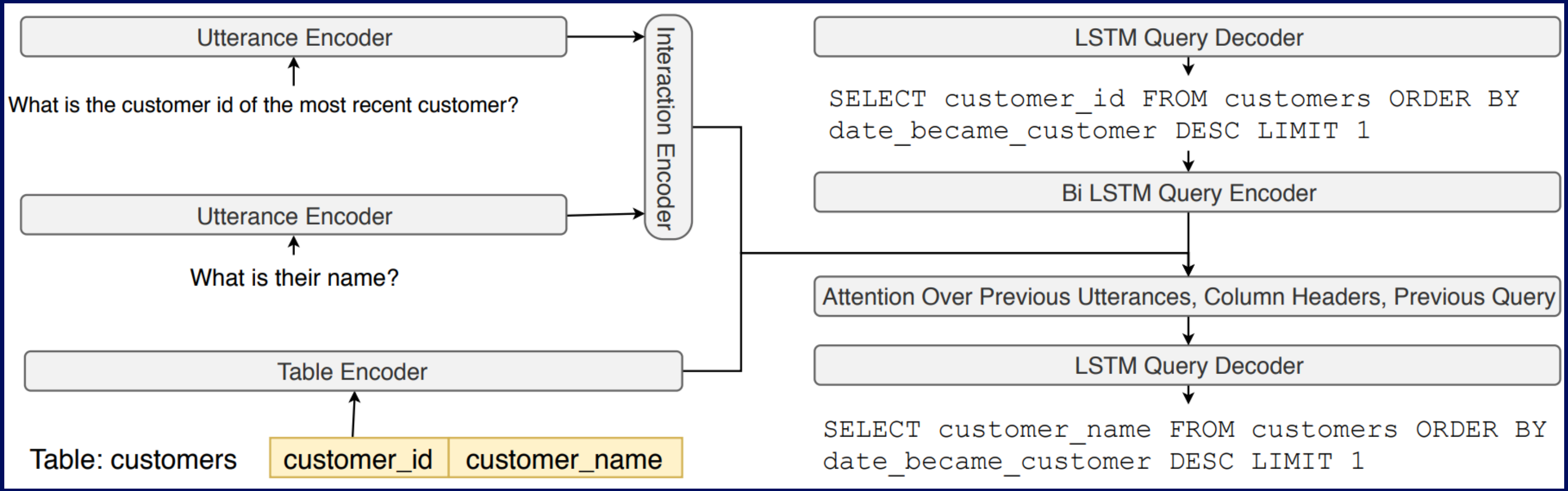
Echo Query [Gabriel Lyons et al. SIGMOD'16]:

- User feedback for query clarification
- Vocabulary personalization through user interactions
- Focuses on NL to SQL translation for simple SPJ Queries with filters and group bys
- Hands Free Voice Dialogue based interaction
- Built using the Amazon Alexa Voice Service



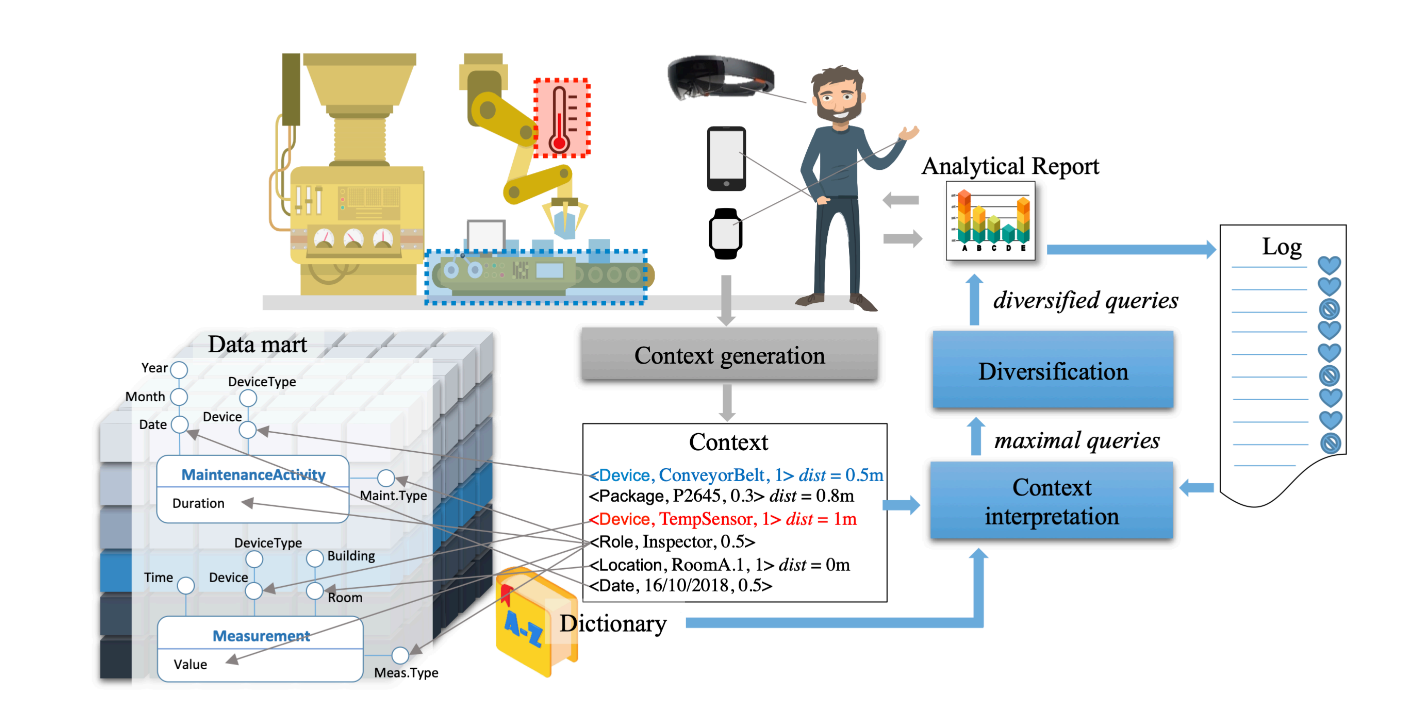
Intent Identification: Utilizing Conversational Context

- Editing based SQL Query generation [Zhang et al. EMNLP 2019]:
  - An encode-decoder architecture with attention mechanisms
  - Use neural networks (Bi-LSTMs) to capture semantic understanding of user utterances, table schema and the mapping between the two
    - Utterance encoder uses bi-LSTM to generate utterance token embedding with attention to the column header embeddings and **context** from previous utterances
    - Table encoder uses bi-LSTM with attentions to encode the internal structure of the table schema as well as the relationship between utterance and the table schema



Intent Identification: Utilizing User Feedback and Context

- Augmented Business Intelligence: (A-BI, Francia et al., EDBT 2019)
  - Takes the situational context of the user into account (Device, Role, location, date, etc.)
  - Incorporates user feedback on queries
  - Uses collaborative filtering for better user experience (Recommendations)



Intent Identification

- Open Challenges
  - Requirement of substantial amount of training data
  - Incorporation of domain specific understanding
    - Understanding workload patterns and their mapping to the domain schema
  - Ability to handle unseen user utterances
    - Need for Hybrid approaches
      - Intent classification for a fixed set of pre-defined intents
      - Dynamic learning:
        - Rule based interpretation[Athena] / Other NN approaches required to respond to new/unseen utterances
        - Allows learning of new intents dynamically

Entity Specification

- Entities are a critical part of deep domain understanding
  - Constitute the domain vocabulary for the conversation system
    - Can refer to both meta data and data instances (Company: IBM, Drug:Aspirin)
  - Synonyms
    - Provide flexibility in understanding user utterances
    - General purpose synonyms may be provided using external sources such as WordNet

Entities:	**Examples**
Concepts:	Drug, Precautions, Dosage, Indication
Risk:	Contra-Indication, Black Box Warning
Drug Interaction:	DrugFood Interaction, DrugLab Interaction
Drug:	Aspirin, Ibuprofen, Citicoline, Pancreatin
Indication:	Fever, Headache, Bronchitis, Diabetes
Contra-Indication:	Cardiovascular disease, Breast carcinoma

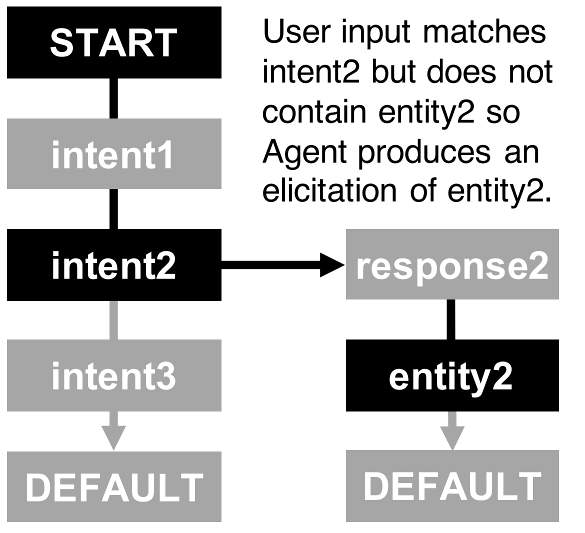
Entity	**Synonyms**
Adverse Effect:	Side effect, adverse reaction, adverse event, AE
Condition:	disease, finding, disorder
Drug:	medicine, meds, medication, substance
Precaution:	caution, safe to give
Dosage:	dosing
Dose adjustment:	dose modification, dosing modification, dose reduction

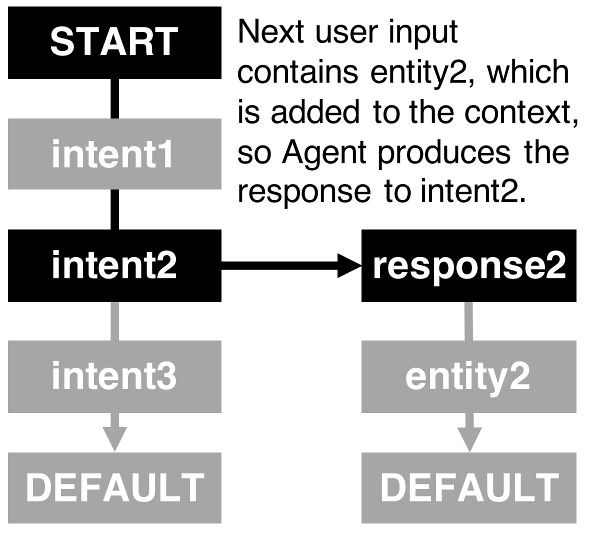
Entity Specification

- Open Challenges
  - Amount of state that needs to be built for entity recognition can become quite large
  - Deep domain understanding
    - Domain specific synonyms (Kidney Disease, Renal Failure)
    - Hierarchical relationships
      - Taxonomies
      - External ontologies
      - Query Relaxation: Incorporating information from external KBs [Chuan et.al EDBT 2020]

Building the dialog

- **The Dialog Tree**
  - Defines the space of user utterances the system can recognize and respond to
  - Responses conditioned on a combination of intents and entities identified in the user utterance
  - Context captured from previous utterances





Open Challenges:

- Designing dialog to support expected interaction patterns
  - Static specification common but laborious
  - Learning dialog from prior user experience (Agent based systems [Miner et al. JAMA Internal Medicine 2016])
- Need to handle both domain specific requests and general conversation management [IBM's Alma, Conversational UX Design, Moore et al., ACM 2019]

Training Examples

- Intent identification relies on training samples for identifying intents from user utterances
- The distribution and number of generated training examples for different intents, and the methodology for training the classifier model have a direct impact on its accuracy.
- Domain specific understanding required to generate appropriate training samples
- Most manual methods do not scale well

Training examples for dosage for a drug

Show me the **Dose Adjustment** for Aspirin?

Find **Dose Adjustment** for Aspirin?

Give me the **increased dosage** for Aspirin?

How do I perform a **Dose Adjustment** for Aspirin?

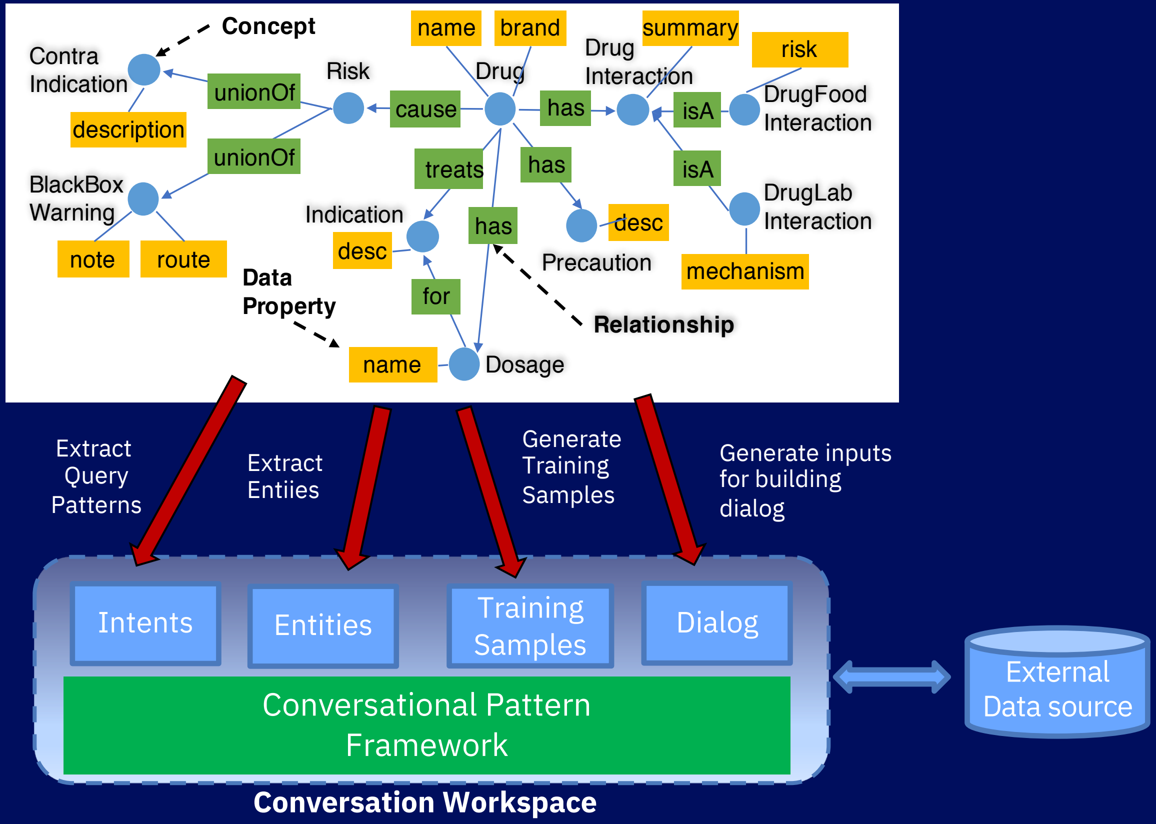
I want to see the **modifications to dosing** for Aspirin?

- Need for incorporating domain specific knowledge
- Substantial manual effort required to build a domain specific conversation system
  - Automatic generation of intents and training examples for domain specific applications
  - Ontologies provide a way to capture and utilize domain specific information

Extension to Dialog

Ontology-based approach for building conversational systems

Quamar et.al SIGMOD 2020, C. Lei et.al IEEE Engr Bulletin 2018



Domain Schema

- Ontologies capture the semantics of the domain schema in terms of entities, relationship providing deep domain specialization

Intents:

- Workload patterns mapped onto the domain schema and identified as intents

Entities:

- Build the domain vocabulary of the system
- Ontology concepts, instances, synonyms

Dialog:

- Supports the desired interaction for the application conditioned on identified intent and entities

Knowledge Base data:

- Interaction with external data source through structured queries to respond to user/application queries

Open Challenges

Complex workloads with complex queries

Hybrid approaches for interpretations:
Combine the strength of ML and entity -based solutions

Domain adaptation and use in the enterprise

Extensions to conversation

Benchmarks

Benchmarks

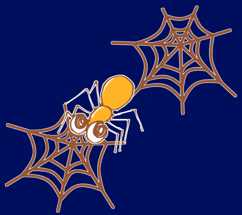
- Benchmarks allow tracking progress, and great tool
- Many emerging benchmarks for NL to SQL
  - Early ones -
    GEO http://www.cs.utexas.edu/users/ml/nldata/geoquery.html and
    MAS https://academic.microsoft.com/home
  - WikiSQL - https://github.com/salesforce/WikiSQL
  - Spider - https://yale-lily.github.io/spider
  - FIBEN – IBM
  - SParC (multi-turn) - https://yale-lily.github.io/sparc
  - CoSQL (multi-turn) - https://yale-lily.github.io/cosql

Benchmarks:

WikiSQL

- Crowd-source set of labeled dataset for NLQ over relational data
- About 80,000 hand-annotated example questions and corresponding SQL queries
- About 2400 tables from Wikipedia
- Single table queries with aggregation and selection
- Many systems that report results
  - HydraNet (2020) at 92.2% test execution accuracy
- PRO: Largest labeled data set that covers tables from many domains
- CON: Simple query focus; systems risking overfitting to the data set

Benchmarks:
Spider



- Large scale complex and cross-domain benchmark
- Emphasis on testing cross domain robustness
- Has multiple schemas, each having multiple tables
  - 200 databases with multiple tables, covering 138 different domains
- Complex workload: 5,693 unique complex SQL queries with 10181 NLQ
  - Joins, and nested queries, aggregations
- PRO: Cross domain focus, more complex queries
- CON: Individual database are still simple, more reflexive of database supporting web pages

Benchmark:

FIBEN

- Benchmark from IBM [Athena++, SIGMOD 2019]
- Simulates a complex financial data mart
  - Combines SEC data, and TPoX benchmark
- Final database conforms to a combined FIBO and FRO ontologies
- Complex query set: 300 complex BI queries with nesting, as well as joins, and aggregations
- PRO: Only workload that addresses a complex warehouse scenario
- CON: Single domain

Concluding Remarks

Very active area of research both from NLP and database communities

- Covered a subset of system that are representative, many more

Recent advancements in NLU major propellant

Yet, NLQ is not widely used in the enterprise

Conversational data exploration is the next wave


thank you
danke
謝謝
ngiyabonga
teşekkür ederim
рахмат
Баярлалаа
спасибо
mersi
barka
welalin
tack
spas
kia ora
faafetai lava
vinaka
blagodaram
dank je
misaotra
matondo
paldies
grazzi
mahalo
gracias
tapadh leat
хвала
asante
manana
obrigada
murakoze
tenki
kiitos
dankie
nanni
nandri
dhanyavad
bedankt
bayarlalaa
gracie
hvala
mauruuru
koszonom
enkosi
dziękuję
dankon
akun
aciu
djiere dieuf
tau
mochchakkeram
mamnun
chnorakaloutioun
gratias ago
gracies
sulpay
go raibh maith agat
sobodi
dekuji
sagolun
sukriya
kop khun krap
taiku
obrigado
mesi
didi madloba
najis tuke
ありがとう
tanemirt
grazie
arigatô
takk
dakujem
trugarez
terima kasih
rahmet
diolch
dhanyavadagalu
shukriya
merce
мерси
kam sah hamnida
rahmat
감사합니다
xiexie
ευχαριστώ
merci


QUESTIONS

References

- B. Aditya, G. Bhalotia, S. Chakrabarti, A. Hulgeri, C. Nakhe, P. Parag, and S. Sudarshan. Banks: Browsing and keyword searching in relational databases. VLDB 2002: 1083-1086
- K. Affolter, K. Stockinger, and A. Bernstein. A comparative survey of recent natural language interfaces for databases. The VLDB Journal, 28 (5), 793–819, 2019
- T. Asakura, J. D. Kim, Y. Yamamoto, Y. Tateisi, and T. Takagi. A Quantitative Evaluation of Natural Language Question Interpretation for Question Answering Systems. JIST 2018: 215–231
- C. Baik, Z. Jin, M. J. Cafarella, and H. V. Jagadish. Constructing Expressive Relational Queries with Dual-Specification Synthesis. CIDR 2020
- C. Baik, Z. Jin, M. J. Cafarella, H. V. Jagadish. Duoquest: A Dual-Specification System for Expressive SQL Queries. SIGMOD 2020: 2319-2329
- F. Basik, B. Hättasch, A. Ilkhechi, A. Usta, S. Ramaswamy, P. Utama, N. Weir, C. Binnig, and U. Çetintemel. DBPal: A Learned NL-Interface for Databases. SIGMOD 2018: 1765–1768
- M. Beveridge and J. Fox. Automatic generation of spoken dialogue from medical plans and ontologies. Journal of Biomedical Informatics, 39(5):482–499, Oct. 2006
- T. Bickmore, H. Trinh, R. Asadi, et al. Safety First: Conversational Agents for Health Care, Springer International Publishing, 33–57, 2018
- L. Blunschi, C. Jossen, D. Kossmann, M. Mori, and K. Stockinger. SODA: Generating SQL for Business Users. PVLDB 5(10): 932–943, 2012
- B. Bogin, M. Gardner, J. Berant. Representing Schema Structure with Graph Neural Networks for Text-to-SQL Parsing. ACL 2019: 4560–4565
- B. Bogin, M. Gardner, J. Berant. Global Reasoning over Database Structures for Text-to-SQL Parsing. EMNLP 2019: 3659–3664

References, cont.

- D. H. Choi, M. Cheol Shin, E. G. Kim, D. R. Shin. RYANSQL: Recursively Applying Sketch-based Slot Fillings for Complex Text-to-SQL in Cross-Domain Databases. CoRR, abs/2004.03125
- W. Cui, Y. Xiao, H. Wang, et al. Kbqa: Learning question answering over qa corpora and knowledge bases. PVLDB 10(5):565–576, 2017
- J. Devlin, M. W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. CoRRabs/1810.04805, 2018
- K. K. Fitzpatrick, A. Darcy, and M. Vierhile. 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):e19, Jun 2017
- J. Gao, M. Galley, and L. Li. Neural approaches to conversational AI. CoRR, abs/1809.08267, 2018
- T. Giorgino, I. Azzini, C. Rognoni, S. Quaglini, M. Stefanelli, R. Gretter, and D. Falavigna. Automated spoken dialogue system for hypertensive patient home management. International Journal of Medical Informatics, 74(2):159 – 167, 2005
- M. Golfarelli, S. Rizzi and M. Francia. Augmented Business Intelligence. DOLAP 2019
- J. Guo, Z. Zhan, Y. Gao, Y. Xiao, J. G. Lou, T. Liu, D. Zhang. Towards Complex Text-to-SQL in Cross-Domain Database with Intermediate Representation. ACL 2019: 4524–4535
- I. Gur, S. Yavuz, Y. Su, and X. Yan. Dialsql: Dialogue based structured query generation. ACL 2018: 1339– 1349
- P. He, Y. Mao, K. Chakrabarti, W. Chen. X-SQL: reinforce schema representation with context. MSR-TR-2019-6
- W. Hwang, J. Yim, S. Park, M. Seo. A Comprehensive Exploration on WikiSQL with Table-Aware Word Contextualization. CoRR, abs/1902.01069, 2019
- B. H. Juang and S. Furui. Automatic recognition and understanding of spoken language - a first step toward natural human-machine communication. Proceedings of the IEEE, 88(8):1142–1165, 2000

References, cont.

- M. Jammi, J. Sen, A. R. Mittal, S. Verma, V. Pahuja, R. Ananthanarayanan, P. Lohia, H. Karanam, D. Saha, and K. Sankaranarayanan. Tooling Frame-work for Instantiating Natural Language Querying System. PVLDB 11 (12): 2014–2017, 2017
- E. Kaufmann and A. Bernstein. Evaluating the usability of natural language query languages and interfaces to Semantic Web knowledge bases.J . Web Semant. 8(4): 377–393, 2010
- A. Kelkar, R. Relan, V. Bhardwaj, S. Vaichal, P. Relan. Bertrand-DR: Improving Text-to-SQL using a Discriminative Re-ranker. CoRR, abs/2002.00557
- G. Koutrika, A. Simitsis, and Y. E. Ioannidis. Précis: The Essence of a Query Answer. ICDE 2006: 69-78
- D. Küpper, M. Storbel, and D. Rösner. NAUDA: A Cooperative Natural Language Interface to Relational Databases. SIGMOD 1993: 529-533
- C. Lei, F. Özcan, A. Quamar, A. R. Mittal, J. Sen, D. Saha, and K. Sankaranarayanan. Ontology-Based Natural Language Query Interfaces for Data Exploration. IEEE Data Eng. Bull. 41: 52–63, 2018
- C. Lei, V. Efthymiou, R. Geis and F. Özcan: Expanding Query Answers on Medical Knowledge Bases. EDBT 2020: 567-578
- F. Li and H. V. Jagadish. Constructing an Interactive Natural Language Interface for Relational Databases. PVLDB 8(1): 73–84, 2014
- F. Li and H. V. Jagadish. NaLIR: an interactive natural language interface for querying relational databases. SIGMOD 2014: 709-712
- F. Li and H. V. Jagadish. Understanding Natural Language Queries over Relational Databases. SIGMOD Record 45(1): 6–13, 2016
- Y. Li and D. Rafiei. Natural Language Data Management and Interfaces: Recent Development and Open Challenges. SIGMOD 2017: 1765–1770
- Y. Li, H. Yang, and H. V. Jagadish. NaLIX: An Interactive Natural Language Interface for Querying XML. SIGMOD 2005: 900-902
- G. Lyons, V. Tran, C. Binnig, U. Cetintemel, and T.Kraska. 2016. Making the case for Query-by-Voice with EchoQuery. SIGMOD 2016: 2129–2132
- Q. Lyu, K. Chakrabarti, S. Hathi, S. Kundu, J. Zhang, Z. Chen. Hybrid Ranking Network for Text-to-SQL. MSR-TR-2020-7

References, cont.

- S. Mallios and N. Bourbakis. A survey on human machine dialogue systems. IISA 2016: 1–7
- M. F. McTear. Spoken dialogue technology: Enabling the conversational user interface. ACM Computing Surveys, 34(1):90–169, Mar. 2002
- A. S. Miner, A. Milstein, S. Schueller, et al. Smartphone-Based Conversational Agents and Responses to Questions About Mental Health, Interpersonal Violence, and Physical Health. JAMA Internal Medicine, 176(5):619–625, 2016
- R. J. Moore and R. Arar. Conversational UX Design: A Practitioner’s Guide to the Natural Conversation Framework. ACM, New York, NY, USA, 2019
- A. Quamar, C. Lei, D. Miller, F. Özcan, J. Kreulen, R. J. Moore, and V. Efthymiou. An Ontology-Based Conversation System for Knowledge Bases. SIGMOD 2020: 361-376
- D. Saha, A. Floratou, K. Sankaranarayanan, U. F. Minhas, A. R. Mittal, and F. Özcan. ATHENA: An Ontology-Driven System for Natural Language Querying over Relational Data Stores. PVLDB 9(12): 1209–1220, 2016
- J. Sen, F. Özcan, A. Quamar, G. Stager, A. R. Mittal, M. Jammi, C. Lei, D. Saha, and K. Sankaranarayanan. Natural Language Querying of Complex Business Intelligence Queries. SIGMOD 2019: 1997-2000
- T. Shi, K. Tatwawadi, K. Chakrabarti, Y. Mao, O. Polozov, W. Chen. IncSQL: Training Incremental Text-to-SQL Parsers with Non-Deterministic Oracles. CoRR, abs/1809.05054, 2018
- A. Simitsis, G. Koutrika, and Y. E. Ioannidis. Précis: from unstructured keywords as queries to structured databases as answers. The VLDB J. 17 (1), 117–149, 2008
- D. Song, F. Schilder, C. Smiley, C. Brew, T. Zielund, H. Bretz, R. Martin, C. Dale, J. Duprey, T. Miller, and J. Harrison. TR Discover: A Natural Language Interface for Querying and Analyzing Interlinked Datasets. ISWC 2015: 21-37
- S. Tata and G. M. Lohman. SQAK: Doing More with Keywords. SIGMOD 2008: 889-902
- S. Walter, C. Unger, P. Cimiano, and D.l Bär. Evaluation of a Layered Approach to Question Answering over Linked Data. ISWC 2012: 362-374

References, cont.

- C. Unger, L. Bühmann, J. Lehmann, A. C. N. Ngomo, D. Gerber, and P. Cimiano. Template-based question answering over RDF data. WWW 2012: 639–648
- U. Waltinger, D. Tecuci, M. Olteanu, V. Mocanu, and S. Sullivan. USI Answers: Natural Language Question Answering Over (Semi-) Structured Industry Data. IAAI 2013: 1471-1478
- W. Wang, Y. Tian, H. Wang, W. S. Ku: A Natural Language Interface for Database: Achieving Transfer-learnability Using Adversarial Method for Question Understanding. ICDE 2020: 97-108
- B. Wang, R. Shin, X. Liu, O. Polozov, M. Richardson. RAT-SQL: Relation-Aware Schema Encoding and Linking for Text-to-SQL Parsers. ACL 2020
- C. Wang, K. Tatwawadi, M. Brockschmidt, P. S. Huang, Y. Mao, O. Polozov, R. Singh. Robust Text-to-SQL Generation with Execution-Guided Decoding. CoRR, abs/1807.03100, 2018
- N. Weir, P. Utama, A. Galakatos, A. Crotty, A. Ilkhechi, S. Ramaswamy, R. Bhushan, N. Geisler, B. Hättasch, S. Eger, U. Çetintemel, C. Binnig: DBPal: A Fully Pluggable NL2SQL Training Pipeline. SIGMOD 2020: 2347-2361
- X. Xu, C. Liu, and D. Song. SQLNet: Generating Structured Queries From Natural Language Without Reinforcement Learning. CoRR, abs/1711.04436, 2017
- R. V. Yampolskiy.. Turing Test as a Defining Feature of AI-Completeness . In Artificial Intelligence, Evolutionary Computation and Metaheuristics (AIECM) --In the footsteps of Alan Turing. Xin-She Yang (Ed.). pp. 3-17. (Chapter 1). Springer. 2013
- S. J. Young, M. Gasic, B. Thomson, and J. D. Williams. Pomdp-based statistical spoken dialog systems: A review. Proceedings of the IEEE, 101(5):1160–1179, 2013
- T. Young, D. Hazarika, S. Poria, and E. Cambria.2018. Recent trends in deep learning based natural language processing.IEEE Computational Intelligence Magazine 13(3): 55–75, 2018
- T. Yu, Z. Li, Z. Zhang, R. Zhang, and D. R. Radev. TypeSQL: Knowledge-Based Type-Aware Neural Text-to-SQL Generation. NAACL-HLT 2018: 588–594

References, cont.

- T. Yu, R. Zhang, K. Yang, M. Yasunaga, D. Wang, Z. Li, J. Ma, I. Li, Q. Yao, S.Roman, Z. Zhang, and D. R. Radev. Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing andText-to-SQL Task. In EMNLP. 3911–3921, 2018
- T. Yu, R. Zhang, H. Y. Er, S. Li, E. Xue, B. Pang, X. VictoriaLin, Y. C. Tan, T. Shi, Z. Li, Y. Jiang, M. Ya-sunaga, S. Shim, T. Chen, A. R. Fabbri, Z. Li, L. Chen, Y. Zhang, S. Dixit, V. Zhang, C. Xiong, R. Socher, W. S. Lasecki, and D. R. Radev. CoSQL: A Conversational Text-to-SQL Challenge Towards Cross-Domain Nat-ural Language Interfaces to Databases. CoRR, abs/1909.05378, 2019
- T. Yu, R. Zhang, M. Yasunaga, Y. C. Tan, X. V. Lin, S. Li, H. Er, I. Li, B. Pang, T. Chen, E. Ji, S. Dixit, D. Proctor, S. Shim, J. Kraft, V. Zhang, C. Xiong, R. Socher, and D. R. Radev. SParC: Cross-Domain Semantic Parsing in Context. CoRRabs/1906.02285, 2019
- G. Zenz, X. Zhou, E. Minack, W.Siberski, and W. Nejdl. From keywords to semantic queries - Incremental query construction on the semantic web. J. Web Semant. 7, 3 (2009), 166–176
- R. Zhang, T. Yu, H. Er, S. Shim, E. Xue, X. V. Lin, T. Shi, C. Xiong, R. Socher, and D. Radev. Editing-Based SQL Query Generation for Cross-Domain Context-Dependent Questions. EMNLP-IJCNLP 2019: 5337–5348
- W. Zheng, H. Cheng, L. Zou, J. X. Yu, and K. Zhao. Natural language question/answering: Let users talk with the knowledge graph. CIKM 2017: 217–226
- V. Zhong, C. Xiong, and R. Socher. Seq2SQL:Generating Structured Queries from Natural Language using Reinforcement Learning. CoRR, abs/1709.00103, 2017