Say the word, and You'll be Free:
Methods and Techniques for Natural Language Database Interfaces

Altigran da Silva
BDRI@UFAM
Manaus/Brazil

DOING@MaDICS
5 July 2021

Universidade Federal do Amazonas
Programa de Pós-Graduação em Informática - PPGI

ICOMP
Instituto de Computação

Topics

- NLIDB: Motivations, Challenges, Limitations, Demands and Opportunities
- Visions: Data versus Language
- Data-Centric Systems (DCS):
  - NALIR, TEMPLAR, ATHENA and ATHENA++
- Language-Centric Systems (LCS)
  - Background: Seq2Seq Models and Embeddings
  - Seq2SQL and DBPal
- Queries over Multiple Tables
- Conclusions, Remarks, Developments and References
- Hands-on: implementation of an NLIDB in Python

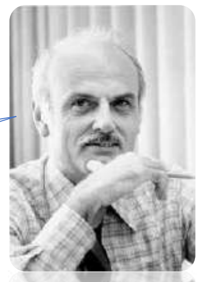
Formal Query Languages are still Hard

- Non-technical and casual users are overwhelmed by technical issues from formal query languages
- SQL was initially developed for executive people to use
- Reality: even trained users face problems to write correct queries [Bowen@WITS'04]
- Users must be aware of:
  - The details on the schema of each DB
  - The semantics of each DB element mentioned in the query
  - The ways for joining information in the DB
  - The syntax of the query language

Natural Language Interfaces for DBs

- Allow casual users to access information stored in DBs using queries expressed in natural language
- The "philosopher's stone" of DB interfaces [Codd@IFIP'1974]

"The only way to encourage the casual user to interact with a database system is to allow free use of their native language."



Edgar Frank "Ted" Codd
Creator of the Relational Model

Example [Affolter@VLDBJ'19]

movie Brad Pitt

Show me all movies with the actor Brad Pitt.

What are the movies with the actor Brad Pitt?

```
SELECT m.title
FROM   Movie m JOIN Starring s ON s.movieId =m.id
               JOIN Actor a ON a.actorId=s.actorId
               JOIN Person p ON p.id = a.actorId
WHERE p.FirstName="Brad" AND p.LastName="Pitt"
```

NL Interfaces for DBs – Challenges

- Requirements
  - Understand the user's intention or information needs when formulating the query.
  - Correctly represent this intention in a structured query language
- Ultimately, it would imply in solving the general problem of Natural Language Understanding (NLU)
  - NLU is hypothetically one of the AI-Hard problems
- Thus, all current systems are necessarily approximated and limited, considering the Codd's statement

NL Interfaces for DBs – Limitations

- All existing methods use general pre-processing techniques
- Each one is based on assumptions on the NL queries they support
- Many times, these hypothesis are not explicit
- Examples of techniques/resources
  - Rules
  - Lexicons, ontologies, dictionaries, etc.
  - Training
  - User interaction
  - Logs

NL Interfaces for DBs – Evaluation

- There is no full understanding of how good techniques really are
- It is unknown how applicable they would be to real world situations
- Different studies, based on different datasets
- Often have limitations and assumptions, implicitly hidden in the context or datasets.
- Some evaluation metrics are commonly used, but they are quite simplistic and do not adequately represent the quality of results.

NL Interfaces for DBs – Evaluation

- Benchmark queries for NLDB qualitative evaluation [Affolter@VLDBJ'19]

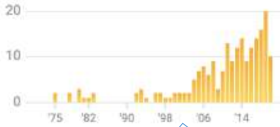
	NL Queries	Likely Operations
Q1	Who is the director of 'Inglourious Basterds'?	Join + String-based Selection
Q2	All movies with a rating higher than 9.	Join + Range-based Selection
Q3	All movies starring Brad Pitt from 2000 until 2010	Join + Date-based Selection
Q4	Which movie has grossed most?	Join + Agreagation/Orderin
Q5	Show me all drama and comedy movies.	Join + União
Q6	List all great movies.	Concept/Subjectivity
Q7	What was the best movie of each genre	Join + Aggregation
Q8	List all non-Japanese horror movies.	Join + Negation-based Selecion
Q9	All movies with rating higher than the rating of 'Sin City'.	Join + Subquery
Q10	All movies with the same genres as 'Sin City'.	Join + Subquery

Demand and Opportunities

- Demands
  - Popularization of IR Systems - Search Engines: Users become used to explore by themselves
  - Data Scientist, Data Journalists
  - Democratization of access to online DBs for casual users
  - Massive use of conversational interfaces
- Opportunities:
  - Technical maturity in NLP, ML & IR allow extracting text semantics with precision and efficiency

Increasing Interest – Citations per Year[1]

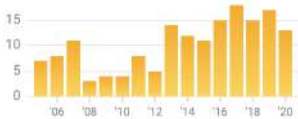
LUNAR - 1973



Progress in natural language understanding: an application to lunar geology

[Woods@AFIPS'1973]

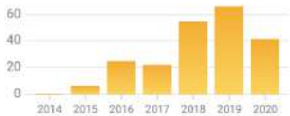
Precise - 2004



Modern Natural Language Interfaces to Databases

[Popescu@COLING94]

NALIR – 2014
Best Paper VLDB



Constructing an Interactive Natural Language Interface for Relational Databases

[Li@PVLDB'14]

[1] Semantic Scholar, September 2020

Increasing Interest – DB Community

- ICDE 2020: 4 papers
- SIGMOD 2020: 5 papers, one tutorial
- VLDB 2020: 3 papers, one tutorial
- ICDE 2021: 4 papers
- SIGMOD 2021: 3 papers
- VLDB 2021: 1 paper so far

Visions: Data *versus* Language

- Data-Centric Systems (DCS):
  - Focus: Map references to DB elements occurring in the query
  - Use rule-based techniques to map NL query words to SQL clauses
  - Less dependent on the DB; More dependent on variations in NL queries
- Language-Centric Systems (LCS):
  - Focus: Instance of the automatic language translation problem
  - Use Deep Learning models and algorithms
  - More dependent on the DB; Less dependent on variations in NL queries

Visions: Data *versus* Language

Data-Centric	
NaLIR	[Li@PVLDB'14]
SQLizer	[Yaghmazadeh,OOPSLA'17]
Templar	[Baik@ICDE'19]
ATHENA	[Saha@PVLDB'16]
ATHENA++	[Sen@PVLDB'20]

Language-Centric	
NSP	[Iyer@ACL'17]
Seq2SQL	[Zhong@CoRR'17
SQLNet	[Xu@CoRR'17]
Coarse2Fine	[Lapata@ACL'18]
STAMP	[Zhou@ACL'18]
PT-MAML	[Huang@NAACL-HLT'18]
TypeSQL	[Yu@NAACL-HLT'18]
SyntaxSQLNet	[Yu@EMNLP'18]
GNN	[Guo@ACL'19]
IRNet	[Bogin@ACL'19]
Photon	[Zeng@CoRR'20]
DBPal	[Weir@SIGMOD'20]

Visions: Data *versus* Language

Data-Centric	
NaLIR	[Li@PVLDB'14]
SQLizer	[Yaghmazadeh,OOPSLA'17]
Templar	[Baik@ICDE'19]
ATHENA	[Saha@PVLDB'16]
ATHENA++	[Sen@PVLDB'20]

Language-Centric	
NSP	[Iyer@ACL'17]
Seq2SQL	[Zhong@CoRR'17
SQLNet	[Xu@CoRR'17]
Coarse2Fine	[Lapata@ACL'18]
STAMP	[Zhou@ACL'18]
PT-MAML	[Huang@NAACL-HLT'18]
TypeSQL	[Yu@NAACL-HLT'18]
SyntaxSQLNet	[Yu@EMNLP'18]
GNN	[Guo@ACL'19]
IRNet	[Bogin@ACL'19]
Photon	[Zeng@CoRR'20]
DBPal	[Weir@SIGMOD'20]

Visions: Data *versus* Language

Data-Centric	
NaLIR	[Li@PVLDB'14]
SQLizer	[Yaghmazadeh,OOPSLA'17]
Templar	[Baik@ICDE'19]
ATHENA	[Saha@PVLDB'16]
ATHENA++	[Sen@PVLDB'20]

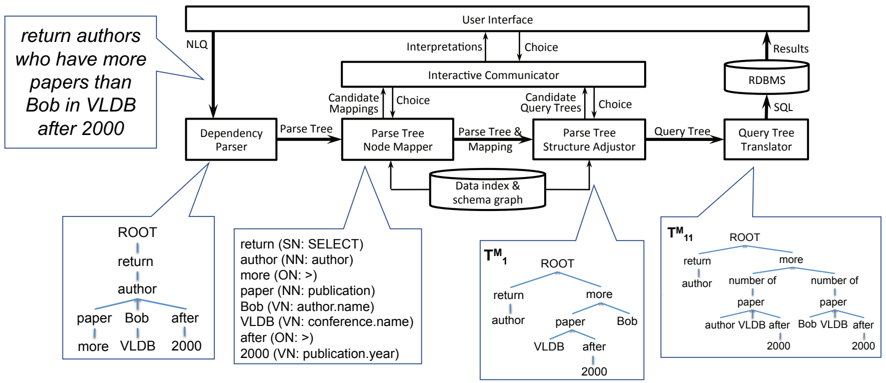
Language-Centric	
NSP	[Iyer@ACL'17]
Seq2SQL	[Zhong@CoRR'17
SQLNet	[Xu@CoRR'17]
Coarse2Fine	[Lapata@ACL'18]
STAMP	[Zhou@ACL'18]
PT-MAML	[Huang@NAACL-HLT'18]
TypeSQL	[Yu@NAACL-HLT'18]
SyntaxSQLNet	[Yu@EMNLP'18]
GNN	[Guo@ACL'19]
IRNet	[Bogin@ACL'19]
Photon	[Zeng@CoRR'20]
DBPal	[Weir@SIGMOD'20]

Data-Centric Systems (DCS)

NaLIR [Li@PVLDB'14]

- Natural Language Interface for Relational databases
- F. Li and H. V. Jagadish – DBGroup University of Michigan
- Best Paper VLDB, 2014
- Often used as a baseline in evaluation experiments with other DCS
- Original code and datasets: https://github.com/umich-dbgroup/NaLIR
- Python implementation by our group: http://t.ly/CM9y
  - Jupyter notebook prepared by Genoveva and Javier Espinosa, thanks!

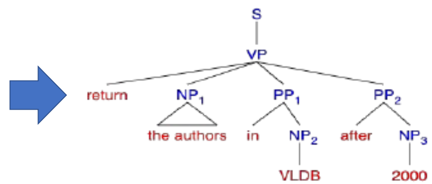
NaLIR [Li@PVLDB'14]



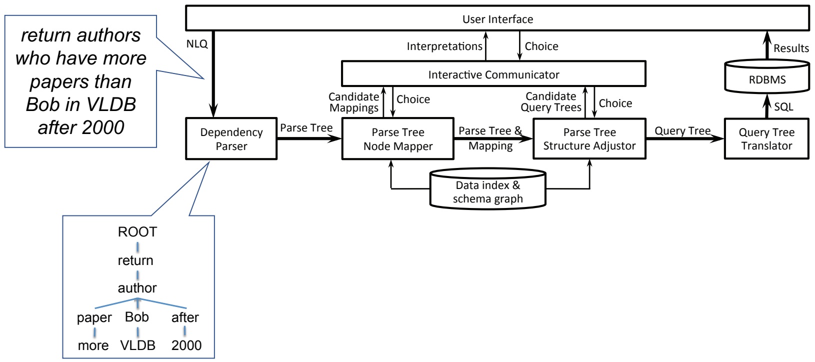
NaLIR [Li@PVLDB'14] – Parsing

- Dependency parsing: task of finding syntactic dependencies in a sentence.
- Syntactic dependencies: asymmetric binary relationship between words
  - Includes grammatical roles (subject, object, determinant, modifier)
- Results in a syntactic dependency tree
- Uses the well-know Stanford Parser [Marneffe@LREC'06]

return the authors in VLDB after 2000



NaLIR [Li@PVLDB'14] – Parsing

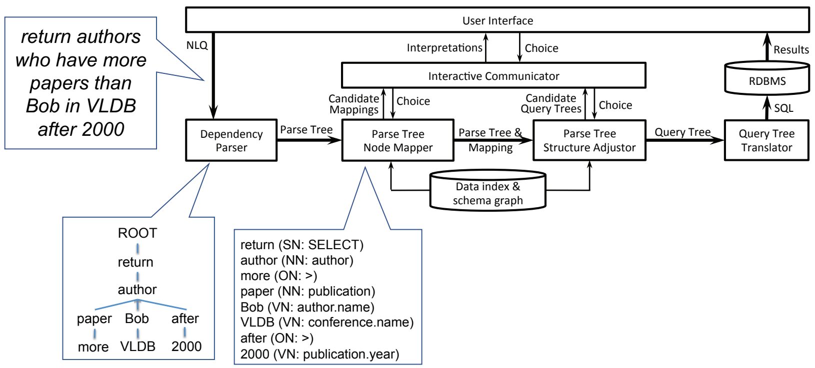


NaLIR [Li@PVLDB'14] – Node Mapper

- Identifies nodes that can be mapped to SQL components
- Uses a table manually constructed that maps NL phrases to SQL clauses
- Problems:
  - Some nodes are not mapped
  - Some nodes have multiple mappings

Node Type	Corresponding SQL Component
Select Node (SN)	SQL keyword: SELECT
Operator Node (ON)	an operator, e.g. =, <=, !=, contains
Function Node (FN)	an aggregation function, e.g., AVG
Name Node (NN)	a relation name or attribute name
Value Node (VN)	a value under an attribute
Quantifier Node (QN)	ALL, ANY, EACH
Logic Node (LN)	AND, OR, NOT

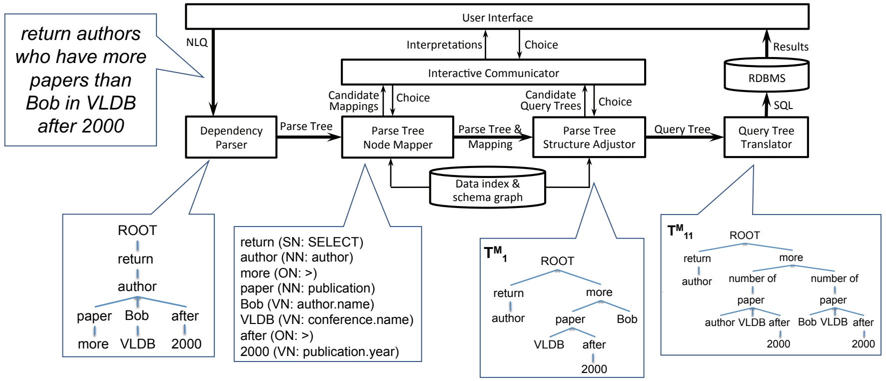
NaLIR [Li@PVLDB'14] – Node Mapper



NaLIR [Li@PVLDB'14] – Tree Adjustor

- Analyzes the dependency tree with the mapped nodes
- Generates Query Trees – candidate interpretations of the NL query
- It can also adjust candidate query trees to make them syntactically valid considering the SQL language
- Ranks the candidate query trees
- The "best" of them leads to the SQL query
- Adjustment and ranking based on a series of fixed heuristics.

NaLIR [Li@PVLDB'14] – Tree Adjustor



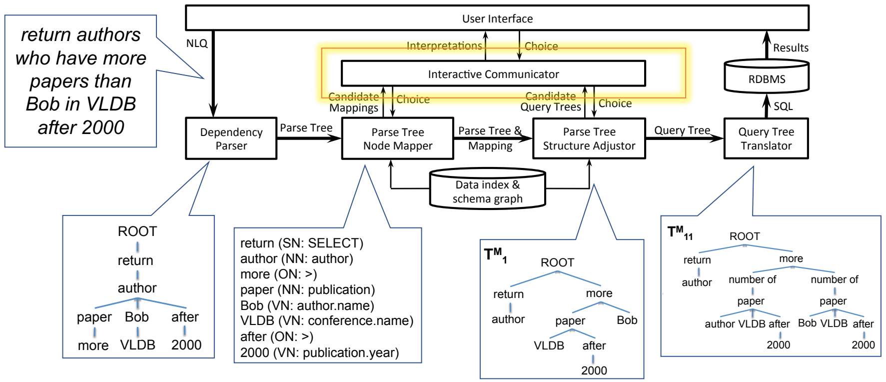
NaLIR [Li@PVLDB'14] – User Interaction

- Several problems can arise in the process
  - Parsing can generate spurious nodes from the query point of view
  - Mapping can fail or be ambiguous
  - Tree adjusting and ranking may fail
- In all these cases, the user is called to intervene
  - Perform adjustments and changes manually.

Experiments with the *Microsoft Academic Search DB*

Queries	With User Interaction	No User Interaction
Easy	34/34	24/32
Medium	34/34	23/34
Hard	20/30	15/32

NaLIR [Li@PVLDB'14] – User Interaction

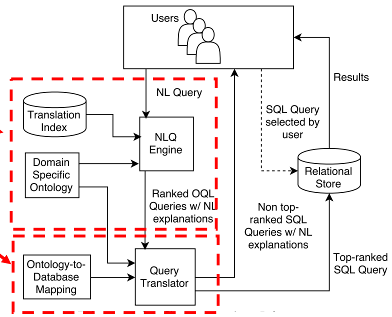


Templar [Baik@ICDE'19]

- From the same group that created NaLIR
- Attempt to decrease user dependency
- Proposes relying on information mined from a query log
- Uses optimization techniques to improve:
  - Mapping of words in the NL query to DB elements
  - Join path generation

ATHENA [Saha@PVLDB'16]

- Ontology-driven system to
- Enable NL queries over relational DBs
- Developed at IBM
- Two Phases:
  - Phase 1: LN Query Interpretation
  - Phase 2: Structured query generation

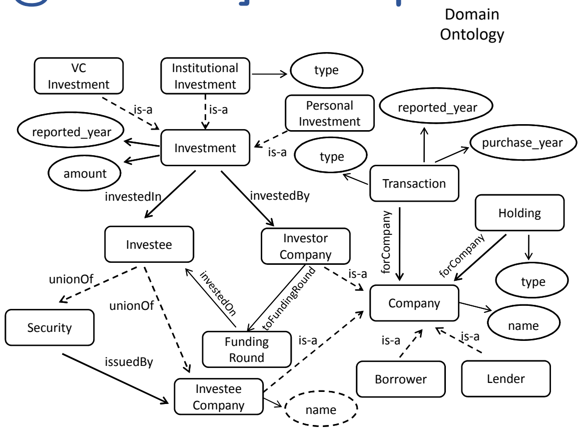


ATHENA [Saha@PVLDB'16] – Fase 1

- Phase 1: NL query Interpretation
- Maps each query word to ontology elements to which it may refer
  - Ex: "Mirian" mapped to `Article.Author` and `Event.Coordinator`
- Mapping combinations yield various interpretations of the LN query
- Each combination corresponds to a tree in the ontology graph
  - Interpretation Trees or ITree
- Finding these trees is a variation of the Steiner Tree Problem
  - An NP-Complete problem

ATHENA [Saha@PVLDB'16] – Example

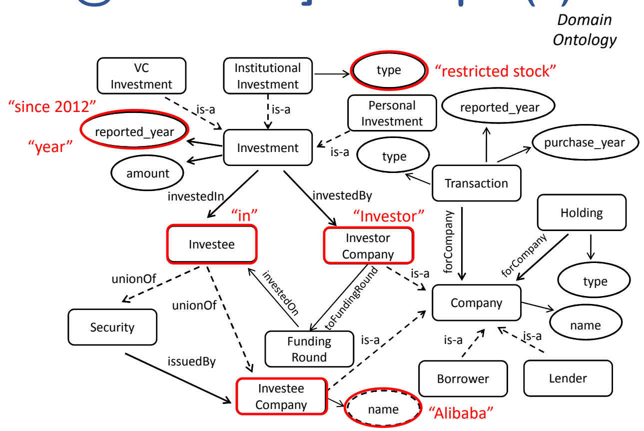
Show me restricted stock investments in Alibaba since 2012 by investor and year



ATHENA [Saha@PVLDB'16] – Example (2)

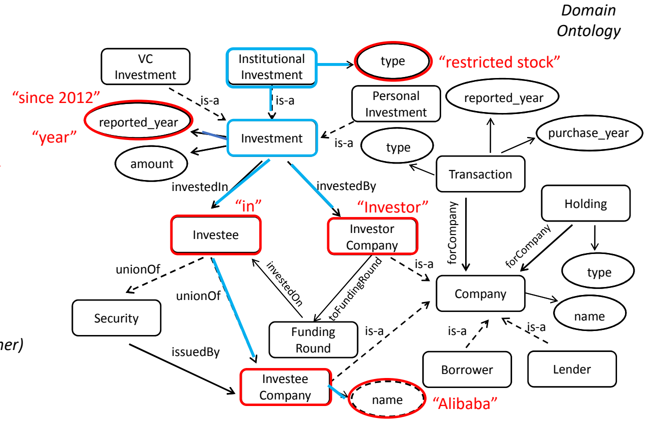
Show me restricted stock investments in Alibaba since 2012 by investor and year

Terminal Nodes



ATHENA [Saha@PVLDB'16] – Example (3)

Show me *restricted stock*
investments *in Alibaba*
since 2012 *by* *investor*
and *year*



Terminal Nodes

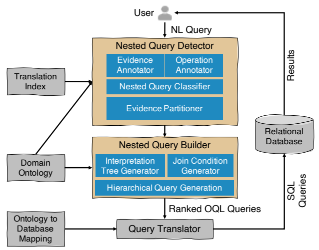
Connection Nodes (Steiner)

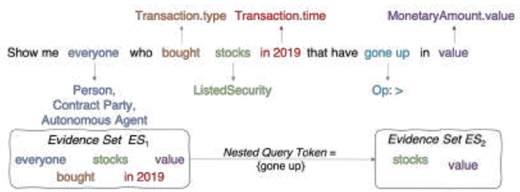
ATHENA [Saha@PVLDB'16] – Fase 2

- Phase 2: Structured Query Generation
  - Relies on an *Ontology-to-Database Mapping (MDG)*
  - Describes how elements of the ontology are mapped to the DB elements
    - (concepts, property, relationships) => (tables, views, columns, FKs)
  - The "best" ITrees transformed into queries according to the MDS
  - Ontology Query Language (OQL):
    - Intermediate language used to allow logical independence
  - Query Translator : OQL to SQL
    - Other QLs can be used
  - ATHENA provides a ranked list of the queries
    - The user can choose the most appropriate

ATHENA++ [Sen@PVLDB'20]

- Extends ATHENA to cover complex nested queries
- The original query is partitioned into possible nested queries, according to a predefined taxonomy.





Language-Centric Systems (LCS)

Language-Centric Systems (LCS)

- Emerged mainly from the NLP community
- Main advantage: rely on machine learning instead of fixed rules
  - E.g., trained to recognize: "major cities" => "city.population > 150,000"
- Explore state-of-the-art Deep Learning techniques
- Specifically: sequence conversion method - Seq2Seq
- Challenge: Training
  - Needed for each target database
    - May involve queries and instances
  - Sometimes costly and error-prone

Sequence-to-Sequence Models (Seq2Seq)

- A Neural Network Model
- Transforms a sequence of elements into another sequence of elements
- Includes two networks: an Encoder (COD) and a Decoder (DEC)
- COD: takes an input sequence and maps to an n-dimensional vector
- DEC: takes the vector and transforms it into an output sequence.
- Most typical application example is machine translation

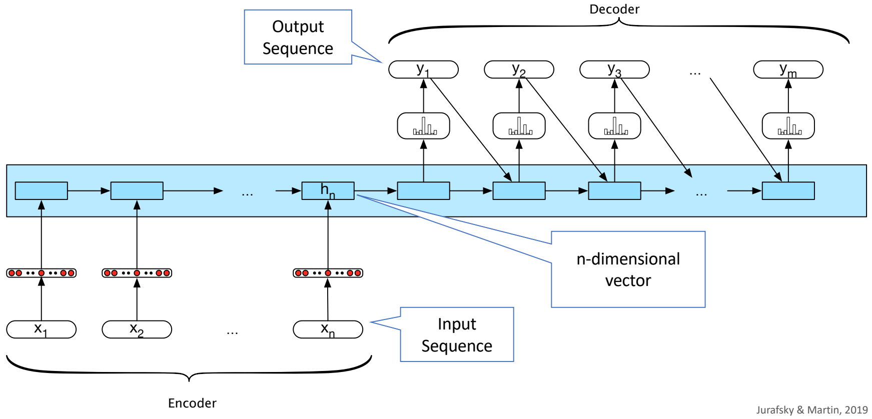
Coder (COD) and Decoder (DEC)

- Imagine COD and DEC as translators, each one speaking two languages.
- The first language is the mother tongue, which differs between the two
- For example, Portuguese and French
- The second is an imaginary language that the two speak
  - This correspond to the n-dimensional vector
- To translate Portuguese into French
  - The COD converts a Portuguese phrase into the imaginary language
  - As the DEC is able to read the imaginary language, it can translate the phrase into French.

Coder (COD) and Decoder (DEC) (2)

- Suppose that, initially, neither COD nor DEC are very fluent in the imaginary language.
- So that they can learn, we train them with several examples
  - This corresponds to the model training
- Usually implemented with Recurrent Neural Networks (RNN)
- Alternatives: LSTMs, Bi-LSTMs, GRU, transformers, ...
- Stacked nets can be used.
- Top-layer output states are the final representation

Encoder-Decoder Networks



Jurafsky & Martin, 2019

Word Embeddings

- Neural models manipulate vectors
- In the case of text, embeddings are vectors that represent words
  - They can also represent sentences, documents, and even attributes of a table!
- In the case of words: word embeddings – semantics inferred from context
- Example:
  - Predict the following word given a prefix:
    - *"When I got home, I forgot to feed the _____* "
  - Suppose we see the following training sentence:
    - " *When I get home, I have to remember to feed the cat*"
  - A traditional model can predict "cat" but not "dog"
  - A neural model can assign high probability also to "dog"
  - Considering that "cat" and "dog" have similar embeddings

Pre-training and Word Embeddings

- Models assume the existence of previously generated embeddings for a large set of words.
- In many cases, the embeddings obtained with methods such as word2vec are sufficient to get good results.
- There many more recent and powerful methods
- The process of generating word embeddings and its properties are itself a subject worth discussing.

Distributional Hypothesis & Vector Semantics

- Distributional Hypothesis (HD)
  - Words with similar meanings tend to occur in similar contexts.
  - Formulated in the 1950's by several linguists
- Vector Semantics
  - Instantiates the HD, creating representations of the meaning of words, called embeddings, from their distributions in a corpus.
  - Used in NLP applications to exploit word semantics
  - Base for more powerful word representation (e.g., ELMo and BERT)
- Representation Learning: embeddings can be learned automatically from input texts

Distributional Hypothesis - Example

- What is Jambú ?
- The word was seen in the following contexts:
  - ”Jambú is delicious sautéed with garlic"
  - ”Jambú is excellent on rice"
  - "... Jambú leaves with salty sauces..."
- Some of the words in the above texts were seen in contexts such as:
  - "... spinach sautéed with garlic over rice ..."
  - "... chard stems and leaves are delicious..."
  - "... collard greens and other salted vegetables..."

Adapted from Jurafsky & Martin, 2019

Distributional Hypothesis - Example

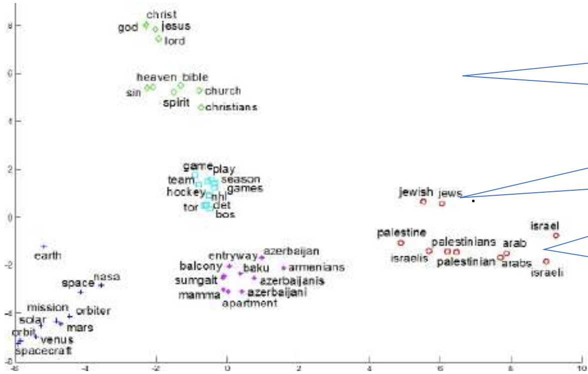
- What is Jambú ?
- The word was seen in the following contexts:
  - ”Jambú is delicious sautéed with garlic"
  - ”Jambú is excellent on rice"
  - "... Jambú leaves with salty sauces..."
- Some of the words in the above texts were seen in contexts such as:
  - "... spinach sautéed with garlic over rice ..."
  - "... chard stems and leaves are delicious..."
  - "... collard greens and other salted vegetables..."

Adapted from Jurafsky & Martin, 2019

Vector Semantics

- Words represented as vectors or embedding in a multidimensional semantic space
- Allows to estimate the similarity between words (or sentences).
- Combines two intuitions: Distributional Hypothesis and representation of words as numerical vectors.
- There are several versions of vector semantics, each one defining the elements of the vectors in slightly different ways.
- In general, all of them are based on some form of weighted count of neighbor words

Word Embeddings Examples



2D projection of embeddings for a few words.

Words with similar semantics are close in space

The close words are not necessarily syntactically "similar"

Reproduced from Xun et al. KDD '17

Word2Vec

- Algorithm Skip-gram with negative Sampling [Mikolov@NIPS'13]
- Method for generating short and dense embeddings
- Including in the word2vec package and therefore is commonly called word2vec.
- Fast, efficient for training.
- Available online with code and pre-trained embeddings.
- Other popular methods:
  - GloVe [Pennington@EMNLP'14] e fastText [Bojanowski@TACL'17]

Word2Vec – Intuition

- Instead of counting how often each word occurs next to a word W, train a binary classifier to calculate the probability of words occurring near W.
- The embedding is formed from the weights of the learned classifiers.
- Revolutionary intuition: we can use the current text as an implicitly unsupervised training corpus for this classifier;
- A word V occurring near W acts as a positive example.
- Avoids the need for any type of manual labeling
- Proposed in the context of neural language models [Collobert@JMLR'11]

Language-Centered System (LCS)

- Use pre-trained embeddings to encode and represent:
  - Queries in Natural Language
  - Database schemas
  - Database Instances - all tuples with attribute values
- SQL query generated using Seq2Seq models
- All current systems are supervised
- Many consider that the DB contains a single table

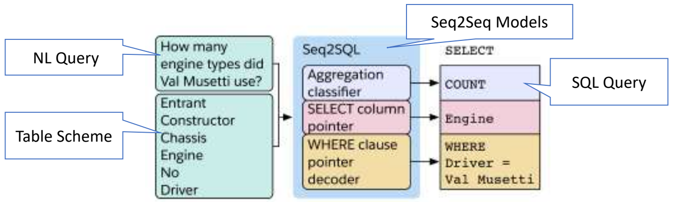
LCS – Benchmarks

- Many LCS are focused on a specific benchmark

Benchmark	DBs and Tables	Queries	Systems
WikiSQL [Zhong@CoRR'17] https://github.com/salesforce/WikiSQL	26,531 tables extracted from Wikipedia HTML tables	80,654 <NL,SQL> pairs No joins Labeled with Mec.Turk	Seq2SQL [Zhong@CoRR'17] SQLNet [Xu@CoRR'17] Coarse2Fine [Lapata@ACL'18] STAMP [Zhou@ACL'18] PT-MAML [Huang@NAACL-HLT'18] TypeSQL[Yu@NAACL-HLT'18]
Spider [Yu@EMNLP'18] https://yale-lily.github.io//spider	200 DBs 128 domains ~5 tables/DB	10.181 NL and 5.693 SQL Include Joins Labeled by 11 Grads	SyntaxSQLNet [Yu@EMNLP'18] GNN [Guo@ACL'19] IRNet [Bogin@ACL'19]

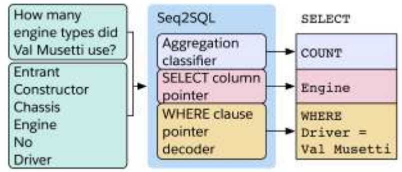
Seq2SQL [Zhong@CoRR'17]

- Developed at Salesforce Research
- Introduces WikiSQL Benchmark - used for training and testing
- Takes advantage of the inherent structure of SQL queries
  - Encodes the NL query and a target table
  - Predicts each part of the SQL query separately



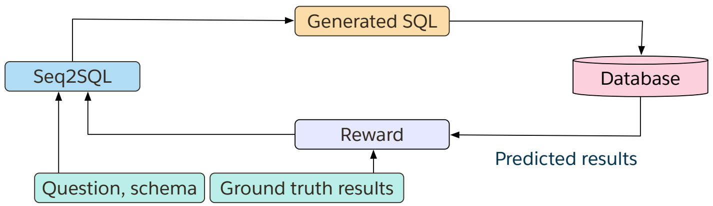
Seq2SQL [Zhong@CoRR'17] – Inference

- Aggregation operations
  - An RNN encodes NL query
  - 4 possible outputs:
    - COUNT, MIN, MAX or NONE
- Columns in SELECT (Projections)
  - An RNN encodes combinations of the query and each column
- WHERE Clause (Selection Predicates)
  - An RNN encodes the query, each column and subset of the SQL vocabulary
  - Pointer Network: Output Vocabulary is made up of input words
- Does not support joins



Seq2SQL [Zhong@CoRR'17] – Training

- Given the columns of the table, for each NL query, generates a candidate SQL query that runs on the DB
- The result of the execution is used as a reward to train a reinforcement learning algorithm



Seq2SQL [Zhong@CoRR'17] – Training (2)

How to Talk to Your Database, by Victor Zhong (https://blog.einstein.ai/how-to-talk-to-your-database/

Seq2SQL [Zhong@CoRR'17] – Similar Systems

- Many other similar systems also encode the DB at the input and decode the output using pointer networks
- Some also assume a SQL (Slot-Filling) template:
  - SQLNet [Xu@CoRR'17], Coarse2Fine [Lapata@ACL'18]
  - TypeSQL[Yu@NAACL-HLT'18]
- Others decode the SQL query as a sequence of words
  - STAMP [Zhou@ACL'18], PT-MAML [Huang@NAACL-HLT'18]
- Others decode the SQL query into a syntax tree
  - IRL [Bogin@ACL'19], GNN [Guo@ACL'19], SyntaxeSQLNet [Yu@EMNLP'18]

DBPal [Weir@SIGMOD'20]

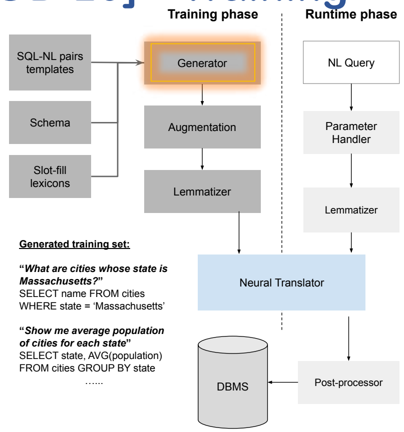
- Johns Hopkins Univ. ,TU Darmstadt e Brown Univ.
- Seq2Seq + attention mechanisms
- Focus on using a limited volume of training data
- Generates synthetic training examples
  - Technique known in ML as data augmentation
  - Uses templates and paraphrase
- Improves overall translation precision
- Increases robustness to language variations

DBPal [Weir@SIGMOD'20]

- Output Vocabulary
  - Schema elements in output vocabulary, not input
  - This vocabulary also includes SQL keywords and constant values
- Narrower vocabulary than usual in Seq2Seq: reduces complexity
  - DEC: Chooses words from this vocabulary to generate the SQL query as the resulting sequence
- Consequences
  - Model specializes in target BD
  - Can only process a query if it contains vocabulary words
  - Model needs to be trained for each new database

DBPal [Weir@SIGMOD'20] – Training

- Relies on multiple SQL query templates
- For each template, there is 1 or more NL templates
- Training generator:
  - Instantiate NL templates with schema elements.
  - NL slots filled with words/phrases from a manually constructed dictionary.

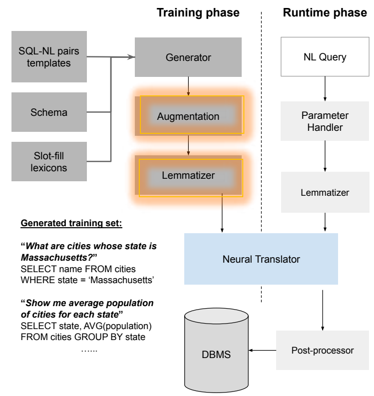


DBPal [Weir@SIGMOD'20] – Training (2)

- Example Template
  - SQL: SELECT {Att}(s) FROM {Table} WHERE {Filter}
  - LN: {SelectPhrase} the {Att}(s) {FromPhrase} {Table}(s) {WherePhrase} {Filter}
- Exemple of an instantiated template
  - SQL: SELECT name FROM patient WHERE age=20
  - LN: Show me the name of all patients with age 20
- Currently, there are about 100 templates
- Typical training sets
  - DBs with a single table: ~1 MM <NL,SQL> template pairs
  - DBs with more tables: ~2 to 3 MM <NL,SQL> template pairs

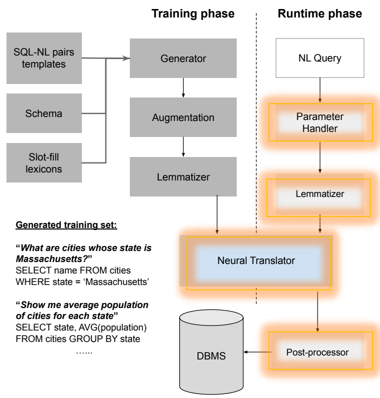
DBPal [Weir@SIGMOD'20] – Training (3)

- Augmentation: Generation of synthetic pairs
- Goal: covering a broad spectrum of linguistic variations for the same SQL query.
- Add pairs <NL,SQL> with linguistic variations in NL
- Words and subphrases randomly exchanged in NL queries, using paraphrases provided by the Paraphrase Database (PPDB)
  - Show patients names with age @AGE => Display patients names with age @AGE.
- Lemmatization: normalize words in <LN, SQL>
  - E.g.: "cars" and "car's" replaced by "car"
- Also applied in run time



DBPal [Weir@SIGMOD'20] – Execution

- Neural Translator: translates the query
- Outcome from training.
- Query constants replaced by markers (placeholders)
- Makes the query independent from the DB state used in the training.
- Then, the lemmatizer is applied
- Postprocessor: Replaces markers with constants
- The query can run in the DBMS



Queries over Multiple Tables

Queries over Multiple Tables

- Possible approach: Generation of Candidate Networks
- Problem:
  - Given a set of references to DB elements, generate a join expression that retrieves joint tuples that best satisfy all of these references
- Concepts:
  - Each reference may correspond to different sets of tuples in the DB
  - **Query Match**: Combination of tuples sets that satisfies user need
  - **Candidate Network**: join expression that retrieves a Query Match
    - Alternatively: a matching tree extracted from the DB schema seem as a graph

References to DB

will smith films

PERSON

	ID	Name
t_1	1	Will Smith
t_2	2	Will Theakston
t_3	3	Maggie Smith
t_4	4	Sean Bean
t_5	5	Elijah Wood

MOVIE

	ID	Title	Year
t_6	6	Men in Black	1997
t_7	7	I am Legend	2007
t_8	8	Harry Potter and the Sorcerer's Stone	2001
t_9	6	The Lord of the Rings: The Fellowship of the Ring	2001
t_{10}	10	The Lord of the Rings: The Return of the King	2003
t_{11}	11	Silent Hill	2006

CHARACTER

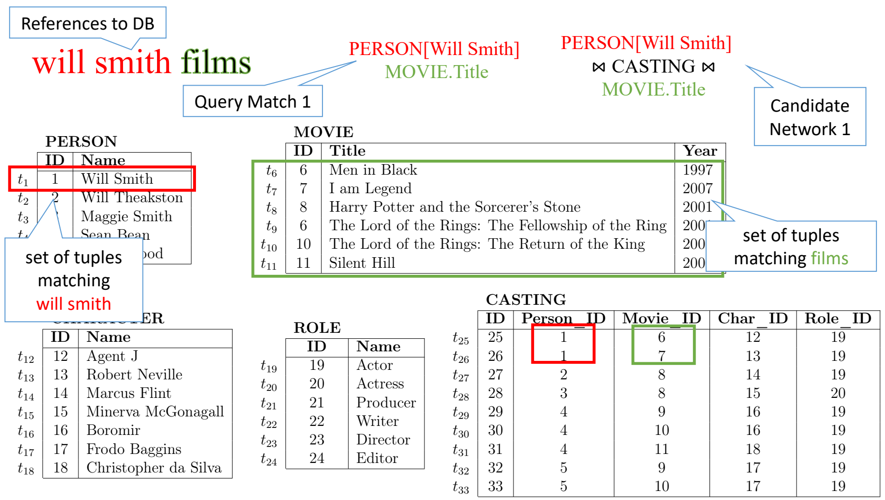
	ID	Name
t_{12}	12	Agent J
t_{13}	13	Robert Neville
t_{14}	14	Marcus Flint
t_{15}	15	Minerva McGonagall
t_{16}	16	Boromir
t_{17}	17	Frodo Baggins
t_{18}	18	Christopher da Silva

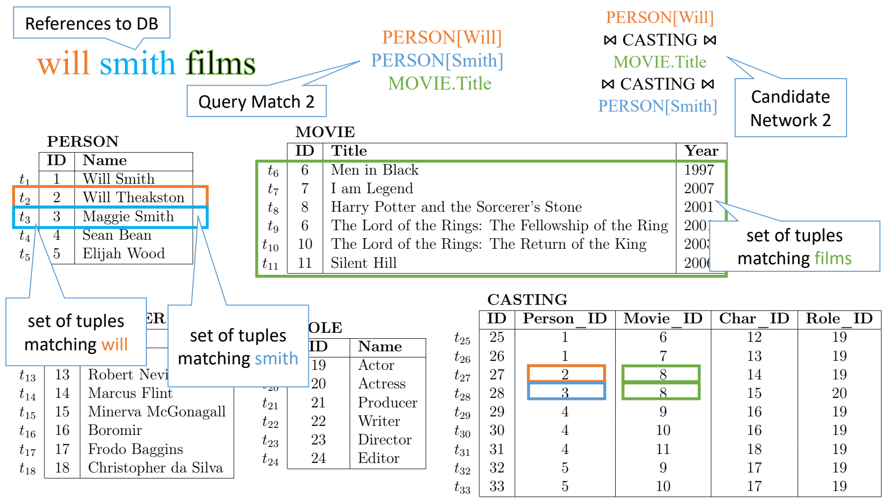
ROLE

	ID	Name
t_{19}	19	Actor
t_{20}	20	Actress
t_{21}	21	Producer
t_{22}	22	Writer
t_{23}	23	Director
t_{24}	24	Editor

CASTING

	ID	Person_ID	Movie_ID	Char_ID	Role_ID
t_{25}	25	1	6	12	19
t_{26}	26	1	7	13	19
t_{27}	27	2	8	14	19
t_{28}	28	3	8	15	20
t_{29}	29	4	9	16	19
t_{30}	30	4	10	16	19
t_{31}	31	4	11	18	19
t_{32}	32	5	9	17	19
t_{33}	33	5	10	17	19





Candidate Networks Generated

will smith films

ID	Name	ID	Person_ID	Movie_ID		ID	Title	Year
1	Will Smith	25	1	6	...	6	Men in Black	1997
1	Will Smith	26	1	7	..	7	I am Legend	2007

will smith films

ID	Name	ID	Person_ID	Movie_ID	...	ID	Title	Year	ID	Person_ID	Movie_ID	...	ID	Name
2	Will Theakston	27	2	8	...	8	Harry Potter and the Sorcerer's Stone	2001	28	3	8	...	3	Maggie Smith

Candidate Network Generation

- Combinatorial Problem: answers must include all references minimally, i.e., without redundancies
- Example: Mondial Database (CIA Factbook)
  - 28 tables, 17,115 tuples, 104 FKs
- For the query "*South East*" :
  - 208 possible Query Matches
  - 105 possible Candidate Networks
    - Up to 10 tables involved

Candidate Network Generation - Approaches

- Problem raised in the context of the DISCOVER system [Hristidis@VLDB'02], pioneer work in keyword queries over DBs
- A series of works produced in our group improved the efficiency in the generation process and the quality of the Candidate Networks
- Efficient Generation of Candidate Networks
  - [Oliveira@ICDE'18] and [Oliveira@TKDE'20]
- Ranking of Candidate Networks
  - [Oliveira@ICDE'15] and [Oliveira@TKDE'20]

Conclusion and Remarks

What was not covered here …

- Systems for keyword queries in relational BDs
  - [Yu@IDEB'10] : A little old survey
  - [Affolter@VLDBJ'19] : Much more recent. It also covers various DCS
- Experimental Results
  - [Kim@PVLDB'20]: Excellent recent survey with experimental results of several NLIDBs with various benchmarks.
- Applications in Conversational and Dialogue Systems
  - [Ozcan@SIGMOD'20]: Tutorial at SIGMOD 2020.
    - Authors from the ATHENA/ATHENA++ group at IBM. It also covers several NLIDBs.

Some Further Developments

- Database Exploration - Tool for Data Scientists and Analyst
  - Doctors, biologists, financial analysts, lawyers, marketing staff, ...
  - Old proposal [Dar@VLDB'98], but only recently carried out.
  - Examples: SODA [Blunschi@VLDB'12] and ATHENA [Lei@IDEB'18]
- Natural language as inter-model *Lingua Franca*
  - Polystores [Duggan@SIGREC'15]: Federations of DBs with multiple data models
  - Data Lakes : centralized repository of raw or minimally cured data available to perform analytical activities [Terrizzano@CIDR'15]
  - Idea explored with keyword queries at INRIA [Hadda@CoRR'20]
- Somewhat surprising connection with the schema evolution problem
  - More "relaxed" queries are less vulnerable to changes in the DB schema
- Idea explored in LESSQL [Afonso@SANER'20] developed by our group.

Thanks to …

- The Divine Wisdom
- Mirian, Genoveva and Anne-Lyse for the kind invitation
- You all for attending … I am honored
- UFAM, Institute of Computing, Graduate Program in Informatics
- The Database and Information Retrieval Group
- JusBrasil and Méliuz
  - Research Support and access to real problems that matter
- CAPES, CNPq and FAPEAM
  - Research Support
- Paulo Martins, Lucas Citolin, Brandell Ferreira,
- SAMSUNG: Support to Paulo Martins

References – Surveys and Tutorial

- [Affolter@VLDBJ'19] Katrin Affolter, Kurt Stockinger, Abraham Bernstein: A comparative survey of recent natural language interfaces for databases. VLDB J. 28(5): 793-819 (2019)
- [Kim@PVLDB'20] Hyeonji Kim, Byeong-Hoon So, Wook-Shin Han, Hongrae Lee:Natural language to SQL: Where are we today? Proc. VLDB Endow. 13(10): 1737-1750 (2020)
- [Ozcan@SIGMOD'20] Fatma Ozcan, Abdul Quamar, Jaydeep Sen, Chuan Lei, Vasilis Efthymiou: State of the Art and Open Challenges in Natural Language Interfaces to Data. SIGMOD Conference 2020: 2629-2636
  - IBM **Tutorial - same authors of ATHENA**
  - https://leichuan.github.io/files/sigmod20-tutorial-slides.pdf

Further References

Further References (1)

- [Affolter@VLDBJ'19] Katrin Affolter, Kurt Stockinger, Abraham Bernstein: A comparative survey of recent natural language interfaces for databases. VLDB J. 28(5): 793-819 (2019) Survey VLDBJ
- [Afonso@SANER'20] Afonso, A., da Silva, A., Conte, T., Martins, P., Cavalcanti, J., & Garcia, A. (2020, February). LESSQL: Dealing with Database Schema Changes in Continuous Deployment. In 2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER) (pp. 138-148). IEEE.
- [Baik@ICDE'19] C. Baik, H. V. Jagadish, and Y. Li. Bridging the semantic gap with SQL query logs in natural language interfaces to databases. In ICDE, pages 374–385, 2019. Templar
- [Basik@SIGMOD'18-Demo] Basik, F., Hättasch, B., Ilkhechi, A., Usta, A., Ramaswamy, S., Utama, P., Weir, N., Binnig, C., Cetintemel, U.: DBPal: A learned NL-interface for databases. In: Proceedings of the 2018 International Conference on Management of Data, pp. 1765–1768. ACM (2018) DBPAL
- [Blunschi@VLDB'12] Lukas Blunschi, Claudio Jossen, Donald Kossmann, Magdalini Mori, Kurt Stockinger: SODA: Generating SQL for Business Users. Proc. VLDB Endow. 5(10): 932-943 (2012) SODA

Further References (2)

- [Bogin@ACL'19] B. Bogin, J. Berant, and M. Gardner. Representing schema structure with graph neural networks for text-to-sql parsing. In ACL, pages 4560–4565, 2019. IRNet
- [Bojanowski@TACL'17] Piotr Bojanowski, Edouard Grave, Armand Joulin, Tomas Mikolov: Enriching Word Vectors with Subword Information. Trans. Assoc. Comput. Linguistics 5: 135-146 (2017)
- [Bowen@WITS'04] Bowen, P., Chang, C., Rohde, F.: Non-length based query challenges: an initial taxonomy. In: 14th Annual Workshop on Information Technology and Systems, WITS, pp. 74–79 (2004)
- [Codd@IFIP'1974] E. F. Codd: Seven Steps to Rendezvous with the Casual User. IFIP Working Conference Data Base Management 1974: 179-200
- [Collobert@JMLR'11] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, Pavel P. Kuksa: Natural Language Processing (Almost) from Scratch. J. Mach. Learn. Res. 12: 2493-2537 (2011)

Further References (3)

- [Dar@VLDB'98] Dar, S., Entin, G., Geva, S., & Palmon, E. (1998, August). DTL's DataSpot: Database exploration using plain language. In VLDB (Vol. 98, pp. 24-27).
- [Duggan@SIGREC'15] Jennie Duggan, Aaron J. Elmore, Michael Stonebraker, Magda Balazinska, Bill Howe, Jeremy Kepner, Sam Madden, David Maier, Tim Mattson, and Stan Zdonik. 2015. The BigDAWG Polystore System. SIGMOD Rec. 44, 2 (June 2015), 11–16. Polystores
- [Guo@ACL'19] J. Guo, Z. Zhan, Y. Gao, Y. Xiao, J. Lou, T. Liu, and D. Zhang. Towards complex text-to-sql in cross-domain database with intermediate representation. In ACL, pages 4524–4535, 2019. GNN
- [Hadda@CoRR'20] Mhd Yamen Haddad, Angelos Anadiotis, Yamen Mhd, Ioana Manolescu: Graph-based keyword search in heterogeneous data sources. CoRR abs/2009.04283 (2020)
- [Hendrix@TODS'1978] G.G., Sacerdoti, E.D., Sagalowicz, D., Slocum, J.: Developing a natural language interface to complex data. ACM Trans. Database Syst. (TODS) 3(2), 105–147 (1978)
- [Huang@NAACL-HLT'18] P. Huang, C. Wang, R. Singh, W. Yih, and X. He. Natural language to structured query generation via meta-learning. In NAACL-HLT, pages 732–738, 2018. PT-MAML

Further References (4)

- [Iyer@ACL'17] S. Iyer, I. Konstas, A. Cheung, J. Krishnamurthy, and L. Zettlemoyer. Learning a neural semantic parser from user feedback. In ACL, pages 963–973, 2017. NSP
- [Kim@PVLDB'20] Hyeonji Kim, Byeong-Hoon So, Wook-Shin Han, Hongrae Lee:Natural language to SQL: Where are we today? Proc. VLDB Endow. 13(10): 1737-1750 (2020) Survey dos Coreanos
- [Lapata@ACL'18] M. Lapata and L. Dong. Coarse-to-fine decoding for neural semantic parsing. In ACL, pages 731–742, 2018. Coarse2Fine
- [Lei@IDEB'18] Lei, C., Özcan, F., Quamar, A., Mittal, A. R., Sen, J., Saha, D., & Sankaranarayanan, K. (2018). Ontology-Based Natural Language Query Interfaces for Data Exploration. IEEE Data Eng. Bull., 41(3), 52-63.
- [Li@PVLDB'14] F. Li and H. V. Jagadish. Constructing an interactive natural language interface for relational databases. PVLDB, 8(1):73–84, 2014. NALIR
- [Marneffe@LREC'06] M.-C. de Marneffe, B. MacCartney, and C. D. Manning. Generating typed dependency parses from phrase structure parses. In LREC, pages 449–454, 2006.

Further References (5)

- [Mikolov@NIPS'13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, Jeffrey Dean: Distributed Representations of Words and Phrases and their Compositionality. NIPS 2013: 3111-3119
- [Oliveira@ICDE'15] Pericles de Oliveira, Altigran Soares da Silva, Edleno Silva de Moura: Ranking Candidate Networks of relations to improve keyword search over relational databases. ICDE 2015: 399-410
- [Oliveira@ICDE'18'] Pericles de Oliveira, Altigran Soares da Silva, Edleno Silva de Moura, Rosiane Rodrigues: Match-Based Candidate Network Generation for Keyword Queries over Relational Databases. ICDE 2018: 1344-1347
- [Oliveira@TKDE'20] Pericles de Oliveira, Altigran Soares da Silva, Edleno Silva de Moura, Rosiane Rodrigues, "Efficient Match-Based Candidate Network Generation for Keyword Queries over Relational Databases," in IEEE Transactions on Knowledge and Data Engineering,
- [Ozcan@SIGMOD'20] Fatma Ozcan, Abdul Quamar, Jaydeep Sen, Chuan Lei, Vasilis Efthymiou: State of the Art and Open Challenges in Natural Language Interfaces to Data. SIGMOD Conference 2020: 2629-2636 Tutorial IBM
- [Pennington@EMNLP'14] Jeffrey Pennington, Richard Socher, Christopher D. Manning:Glove: Global Vectors for Word Representation. EMNLP 2014: 1532-1543

Further References (6)

- [Popescu@COLING'94] A. Popescu, A. Armanasu, O. Etzioni, D. Ko, and A. Yates. Modern natural language interfaces to databases: Composing statistical parsing with semantic tractability. In COLING, 2004 PRECISE
- [Saha@PVLDB'16] Saha, D., Floratou, A., Sankaranarayanan, K., Minhas, U. F., Mittal, A. R., & Özcan, F. (2016). ATHENA: an ontology-driven system for natural language querying over relational data stores. Proceedings of the VLDB Endowment, 9(12), 1209-1220. ATHENA
- [Santos@SBBD'18] Gilberto Santos, Pericles de Oliveira, Altigran Soares da Silva, Edleno Silva de Moura, Lathe: light-Weight Keyword Query Processing over Multiple Databases. SBBD Companion 2018: 1-4
- [Sen@PVLDB'20] Jaydeep Sen, Chuan Lei, Abdul Quamar, Fatma Özcan, Vasilis Efthymiou, Ayushi Dalmia, Greg Stager, Ashish R. Mittal, Diptikalyan Saha, Karthik Sankaranarayanan: ATHENA++: Natural Language Querying for Complex Nested SQL Queries. Proc. VLDB Endow. 13(11): 2747-2759 (2020) ATHENA++
- [Sen@SIGMOD'19-Demo] Jaydeep Sen, Fatma Ozcan, Abdul Quamar, Greg Stager, Ashish R. Mittal, Manasa Jammi, Chuan Lei, Diptikalyan Saha, Karthik Sankaranarayanan: Natural Language Querying of Complex Business Intelligence Queries. SIGMOD Conference 2019: 1997-2000 ATHENA++
- [Terrizzano@CIDR'15] Terrizzano, I. G., Schwarz, P. M., Roth, M., & Colino, J. E. (2015, January). Data Wrangling: The Challenging Yourney from the Wild to the Lake. In CIDR.

Further References (7)

- [Weir@SIGMOD'20] Nathaniel Weir, Prasetya Utama, Alex Galakatos, Andrew Crotty, Amir Ilkhechi, Shekar Ramaswamy, Rohin Bhushan, Nadja Geisler, Benjamin Hättasch, Steffen Eger, Ugur Çetintemel, Carsten Binnig: DBPal: A Fully Pluggable NL2SQL Training Pipeline. SIGMOD Conference 2020: 2347-2361 DBPAL
- [Woods@AFIPS'1973] Woods, W.A.: Progress in natural language understanding: an application to lunar geology. In: Proceedings of National Com- puter Conference and Exposition. AFIPS '73, pp. 441–450 (1973)
- [Xu@CoRR'17] X. Xu, C. Liu, and D. Song. Sqlnet: Generating structured queries from natural language without reinforcement learning. CoRR, abs/1711.04436, 2017. SQLNet
- [Yaghmazadeh,OOPSLA'17] N. Yaghmazadeh, Y. Wang, I. Dillig, and T. Dillig. Sqlizer: query synthesis from natural language. PACMPL, 1(OOPSLA):63:1–63:26, 2017 Sqlizer
- [Yu@EMNLP'18] T. Yu, M. Yasunaga, K. Yang, R. Zhang, D. Wang, Z. Li, and D. R. Radev. Syntaxsqlnet: Syntax tree networks for complex and cross-domain text-to-sql task. In EMNLP, pages 1653–1663, 2018. SyntaxSQLNet
- [Yu@EMNLP'18] 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 and text-to-sql tasks. In EMNLP, pages 3911–3921, 2018. SPIDER Benchmark

Further References (8)

- [Yu@IDEB'10] Yu, Jeffrey Xu, Lu Qin, and Lijun Chang. "Keyword search in relational databases: A survey." IEEE Data Eng. Bull. 33, no. 1 (2010): 67-78.
- [Yu@NAACL-HLT'18] T. Yu, Z. Li, Z. Zhang, R. Zhang, and D. R. Radev. Typesql: Knowledge-based type-aware neural text-to-sql generation. In NAACL-HLT, pages 588–594, 2018.TypeSQL
- [Zeng@ACL'20-Demo] Jichuan Zeng, Xi Victoria Lin, Steven C. H. Hoi, Richard Socher, Caiming Xiong, Michael R. Lyu, Irwin King: Photon: A Robust Cross-Domain Text-to-SQL System. ACL (demo) 2020: 204-214 Photon
- [Zeng@CoRR'20] Jichuan Zeng, Xi Victoria Lin, Caiming Xiong, Richard Socher, Michael R. Lyu, Irwin King, Steven C. H. Hoi: Photon: A Robust Cross-Domain Text-to-SQL System. CoRR abs/2007.15280 (2020) Photon
- [Zhong@arXiv’17] V Zhong, C Xiong, R Socher Seq2sql: Generating structured queries from natural language using reinforcement learning, arXiv, 2017
- [Zhong@CoRR'17] V. Zhong, C. Xiong, and R. Socher. Seq2sql: Generating structured queries from natural language using reinforcement learning. CoRR, abs/1709.00103, 2017. Seq2SQL e WikiSQL Benchmark.
- [Zhou@ACL'18] M. Zhou, G. Cao, T. Liu, N. Duan, D. Tang, B. Qin, X. Feng, J. Ji, and Y. Sun. Semantic parsing with syntax- and table-aware SQL generation. In ACL, pages 361–372, 2018. STAMP