강화학습 기반 QA

2017-12-02

Re:Bayes
김영삼

Chatbots and Conversational Agents · Bots (software) · +1

Why are chatbots hot now?

Answer · Request · Follow 10 · Comment · Downvote

39 Answers

Slobodan Spasenovski, Tech follower, Photographer, Blogger
Answered Oct 9

The reason number one is that Facebook opened the door for chatbot. In this way every company can take advantage of the huge use base that Facebook have. Other companies started to follow Facebook example, Like Shopify, Slack and others.

97 Views · 1 Upvote

Upvote 1 · Downvote

Add a comment... Recommended All

Abhimanyu Godara, Building a personal chatbot platform Bottr.me
Answered May 2

I think personal chatbots have the potential to revolutionize the way we live today!

Related Questions

How can I build an intellige
What are the best (AI) cha
How can I create a chatbot
How do chatbots work?
How do you monetize a ch
What is the future of chatb
How are chatbots created?
What is the smartest chatb
Are chatbots the new apps
What are the top 3 Instagr
automatically like hashtags

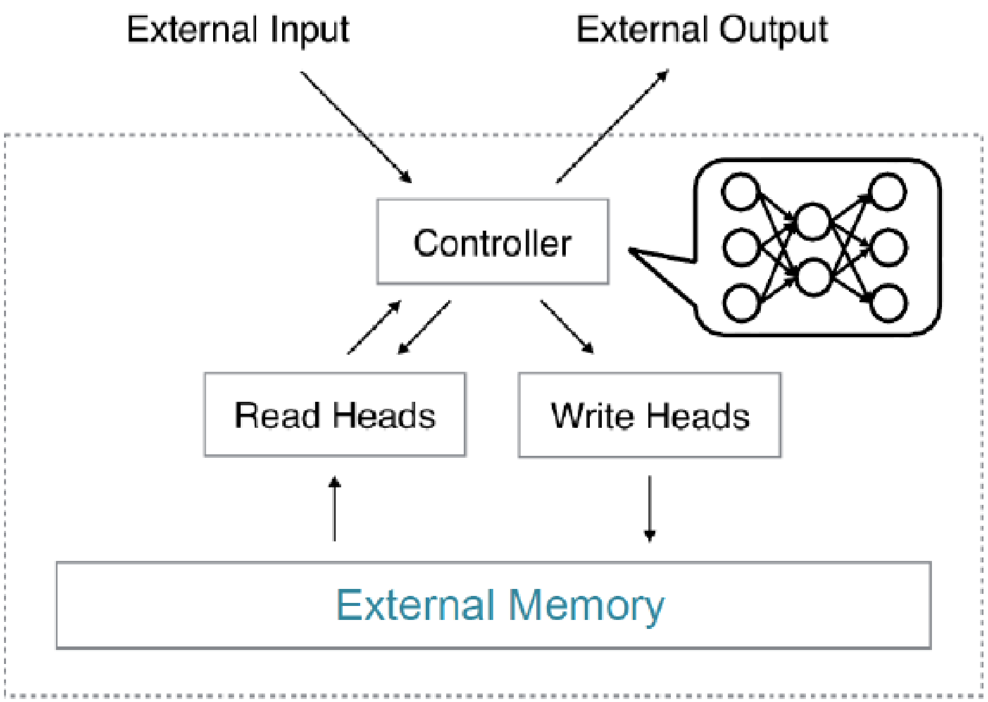
More Related Questions

Chatbots are getting heavy

- Chatbots are a part of dialogue system.
- Today, chatbots are part of virtual assistants such as Google Assistant, and are accessed via many organizations' apps, websites, and on instant messaging platforms (Wikipedia).
- Also, the number of functions of chatbots are increasing.
- Question and Answering is one of them.

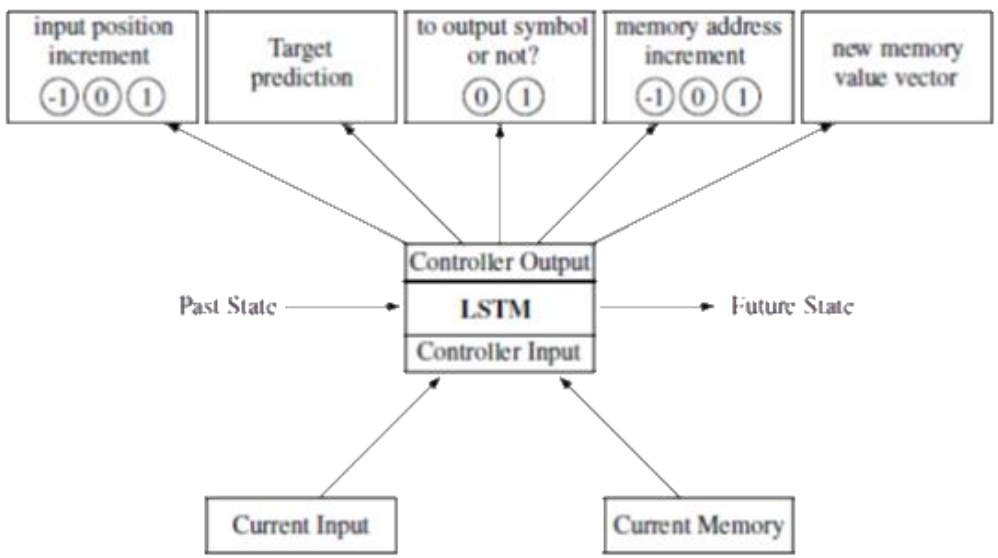
Neural Turing Machines

- NT makes memories differentiable with backprop training.



RL-NTM

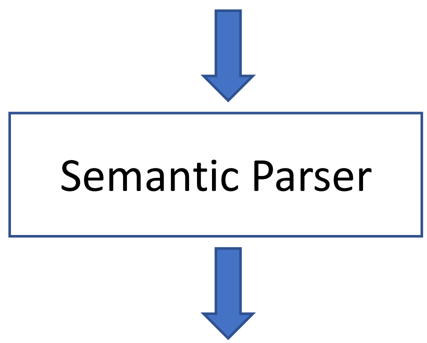
- RL-NTM은 강화학습 알고리즘을 이용해서 입력 테이프나 메모리 테이프의 어디에(where) 접속할 것인지, 그리고 역전파 알고리즘으로 무엇을 쓸 것인지(write)를 결정한다.



시맨틱 파싱 (Semantic Parsing)

- 자연어 표현을 논리형식으로 변환한다.

“한국에서 가장 높은 산은?”



(argmax $0 (and (산:t $0) (loc:t $0 한국:s)) (높이:i $0))

일반적 시맨틱 파싱의 문제점

- Symbolic executor is a non-differentiable operator.
- Typically constrained to a single schema, predefined template
- Requires hand-curated grammars
- Domain specific

Sequence-to-Tree Model

- Dong and Lapata (2016)
- Problem formulation
  - Build a model maps a natural language input $q = x_1, x_2, \ldots, x_n$ to a logical form $a = y_1, y_2, \ldots, y_n$.
  - Encoder that encodes natural language input q into a vector representation
  - Decoder that learns to generate $y_1, y_2, \ldots, y_n$ given the encoding vector.

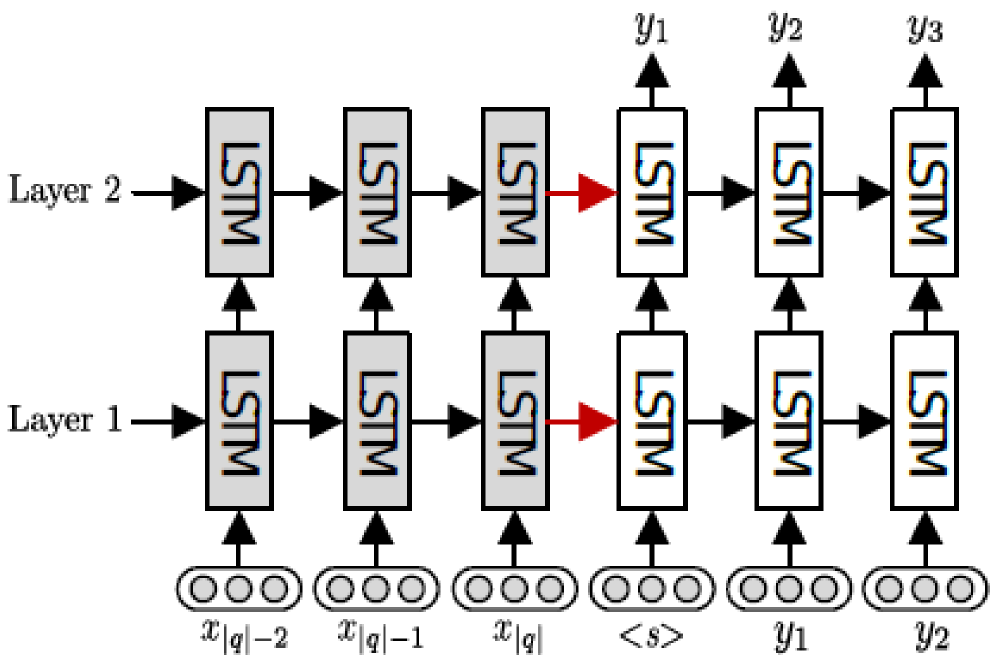


Figure 2: Sequence-to-sequence (SEQ2SEQ) model with two-layer recurrent neural networks.

Seq2Seq 모형의 문제점과 해결책

- 논리 형식(logical form)의 위계적 구조를 제대로 반영하지 못한다.
- 논리 형식을 제대로 표현하기 위해 위계적 트리 디코더(hierarchical tree decoder)가 필요하다.
- 이 모형은 입력에는 동일한 인코더를 사용하나, 전혀 다른 구조의 디코더를 사용한다.
- 위계적 구조를 표상하기 위해 non-terminal token인 <n>이 도입되고, parent feeding 메커니즘이 이용된다.

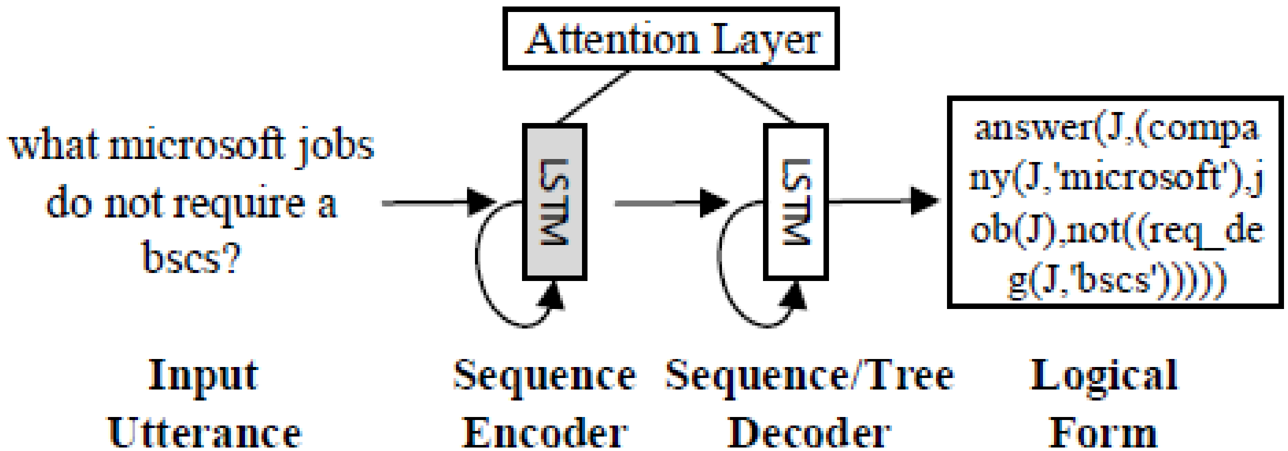
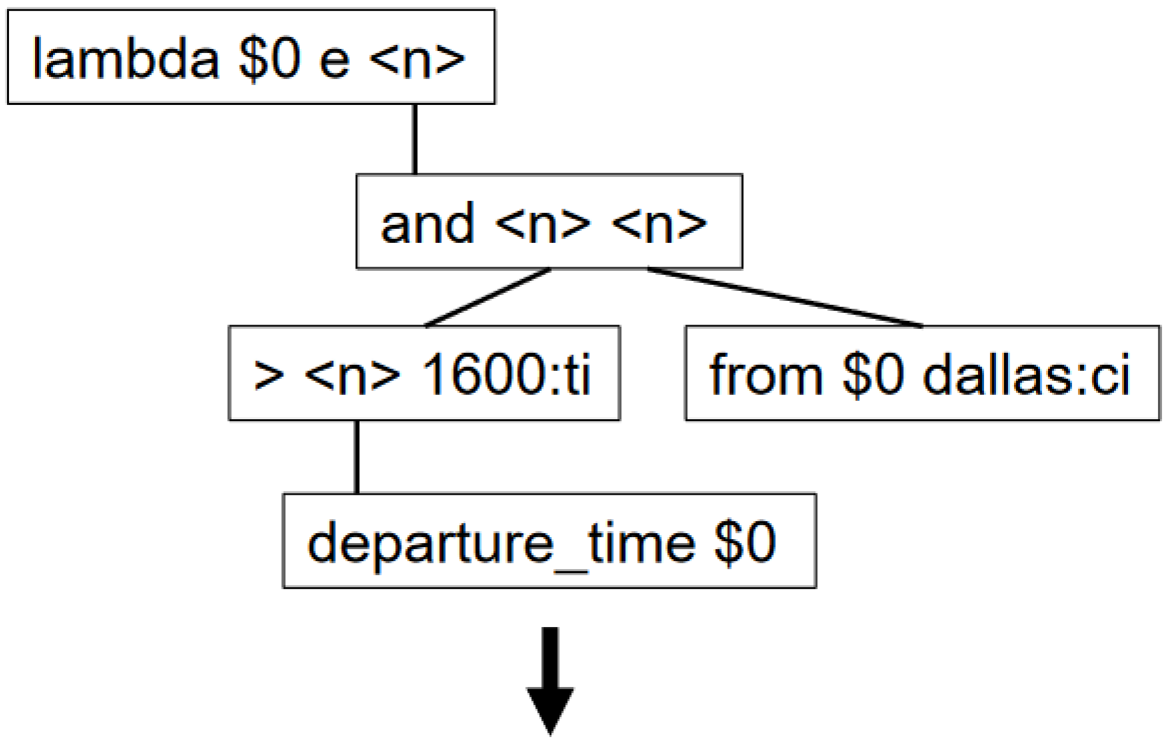


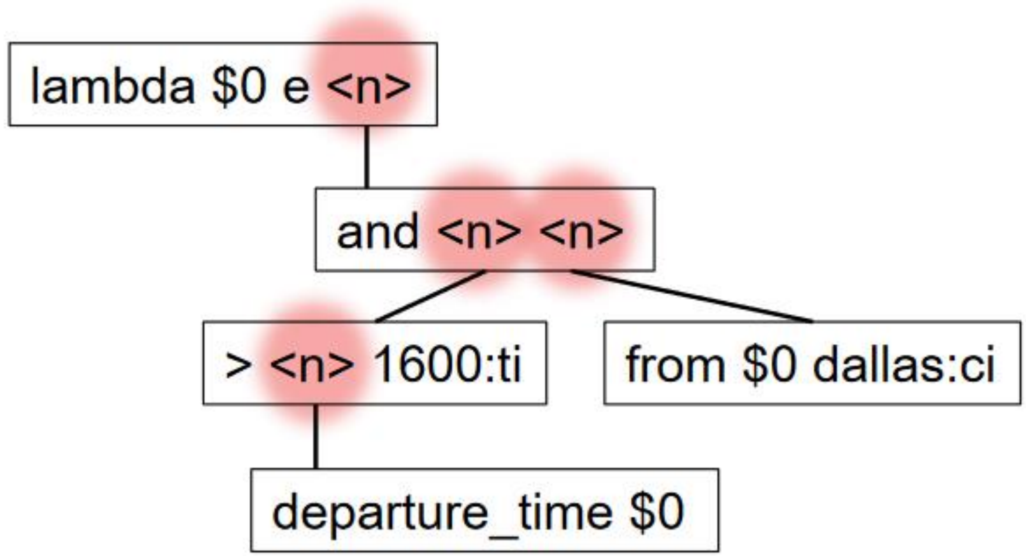
Figure 1: Input utterances and their logical forms are encoded and decoded with neural networks. An attention layer is used to learn soft alignments.

논리형식의 위계적 구조



lambda $0 e (and (> (departure_time $0) 1600:ti) (from $0 dallas:ci))

Define a “nonterminal” <n> token to indicate subtrees in decoder



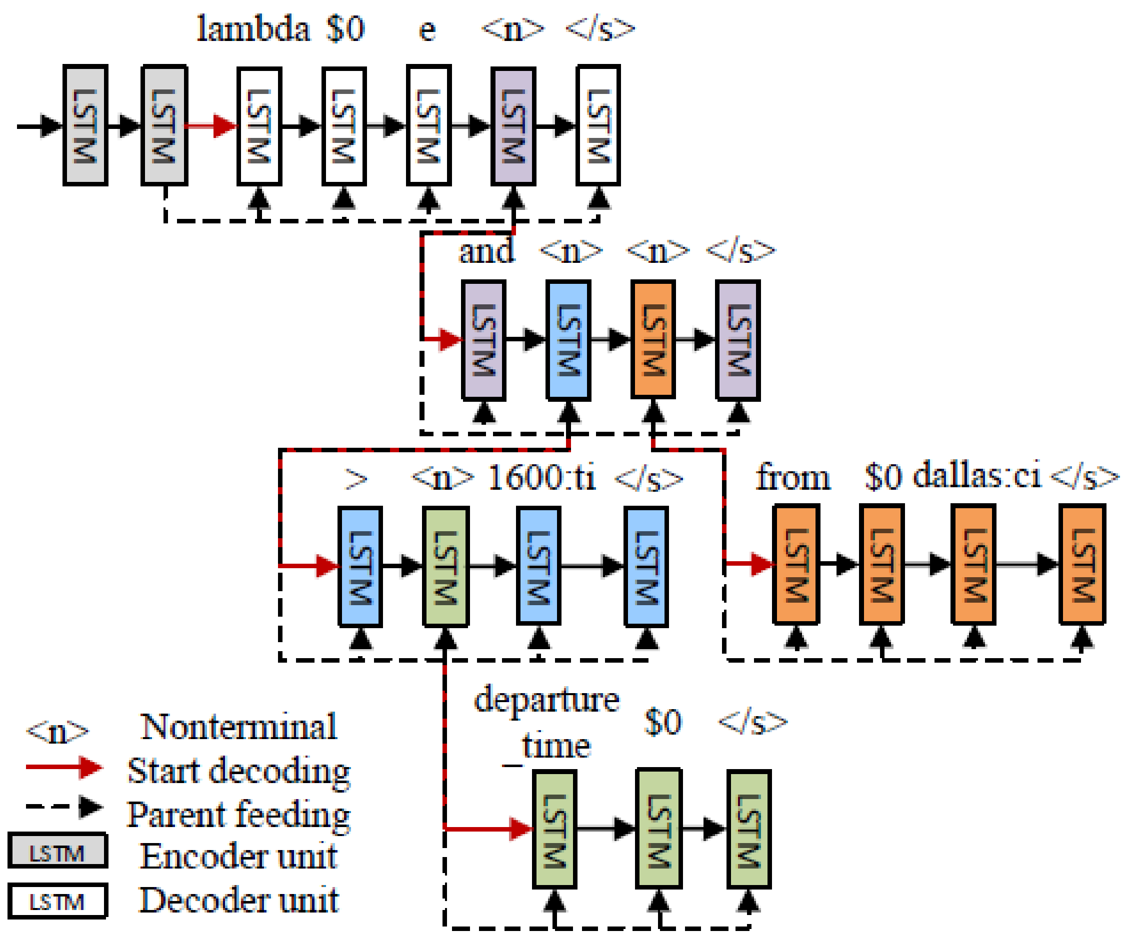


Figure 3: Sequence-to-tree (SEQ2TREE) model with a hierarchical tree decoder.

Algorithm 1 Decoding for SEQ2TREE

Input: q: Natural language utterance

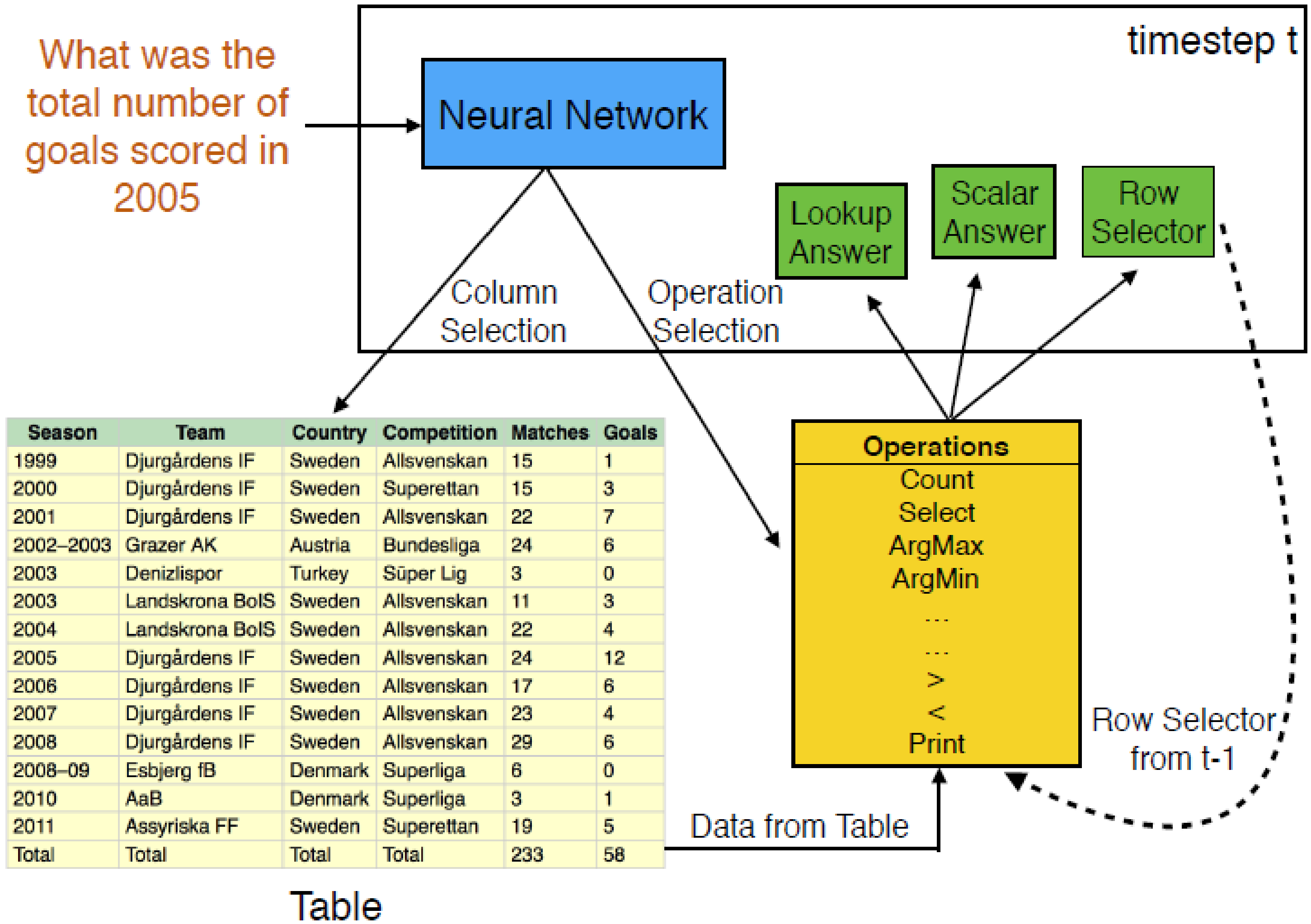
Output: $\hat{a}$: Decoding result

▷ *Push the encoding result to a queue*
$Q.init(\{hid : \mathsf{SeqEnc}(q)\})$
▷ *Decode until no more nonterminals*
while $(c \leftarrow Q.pop()) \neq \varnothing$ **do**
  ▷ *Call sequence decoder*
  $c.child \leftarrow \mathsf{SeqDec}(c.hid)$
  ▷ *Push new nonterminals to queue*
  for $n \leftarrow$ nonterminal in $c.child$ **do**
    $Q.push(\{hid : \mathsf{HidVec}(n)\})$
$\hat{a} \leftarrow$ convert decoding tree to output sequence

- 이 알고리즘은 주어진 자연어 쿼리에서 non-terminal node를 찾을 수 없을 때까지 재귀적으로 수행되며, 위계적 트리구조를 구축한다.
- 부모 지도 연결(parent-feeding connection)은 비단말 부모 노드의 히든 벡터를 자식 노드들과 concatenate 하여 좀 더 위계적 구조를 반영하며 디코딩을 하도록 만든다.

Neural Programmer

- Neelakantan et al. (2017)
- A Sequence-to-sequence model that maps language utterances to programs
- Utilizes a key-variable memory to handle compositionality, a symbolic machine
- Produces both a program, a result of the program
- Operations: count, select, argmax, …, etc.
- Variables: row_selector, scalar_answer, lookup_answer



Season	Team	Country	Competition	Matches	Goals
1999	Djurgårdens IF	Sweden	Allsvenskan	15	1
2000	Djurgårdens IF	Sweden	Superettan	15	3
2001	Djurgårdens IF	Sweden	Allsvenskan	22	7
2002–2003	Grazer AK	Austria	Bundesliga	24	6
2003	Denizlispor	Turkey	Süper Lig	3	0
2003	Landskrona BoIS	Sweden	Allsvenskan	11	3
2004	Landskrona BoIS	Sweden	Allsvenskan	22	4
2005	Djurgårdens IF	Sweden	Allsvenskan	24	12
2006	Djurgårdens IF	Sweden	Allsvenskan	17	6
2007	Djurgårdens IF	Sweden	Allsvenskan	23	4
2008	Djurgårdens IF	Sweden	Allsvenskan	29	6
2008–09	Esbjerg fB	Denmark	Superliga	6	0
2010	AaB	Denmark	Superliga	3	1
2011	Assyriska FF	Sweden	Superettan	19	5
Total	Total	Total	Total	233	58

Table

Neural Enquirer

- Yin et al. (2016)
- NN architecture for answering NL questions given KB table.
- End-to-End differentiable network
- Trained using query-answer pairs
- Main components
    - Query encoder
    - Table encoder
    - Executor: Reader, Annotator (two neural networks)

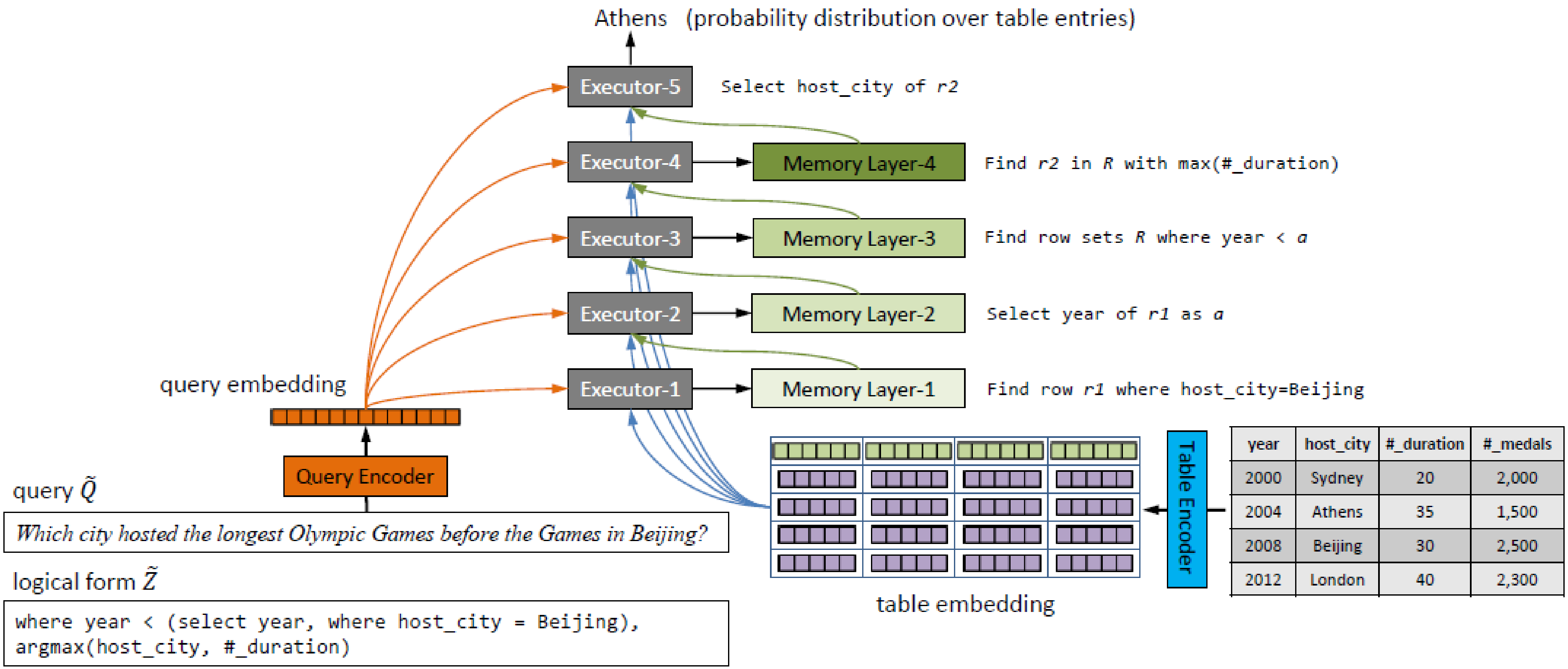


Figure 1: An overview of NEURAL ENQUIRER with five executors

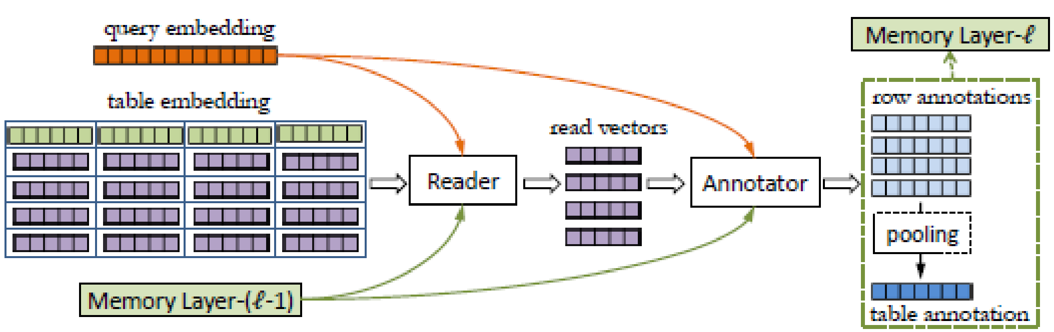
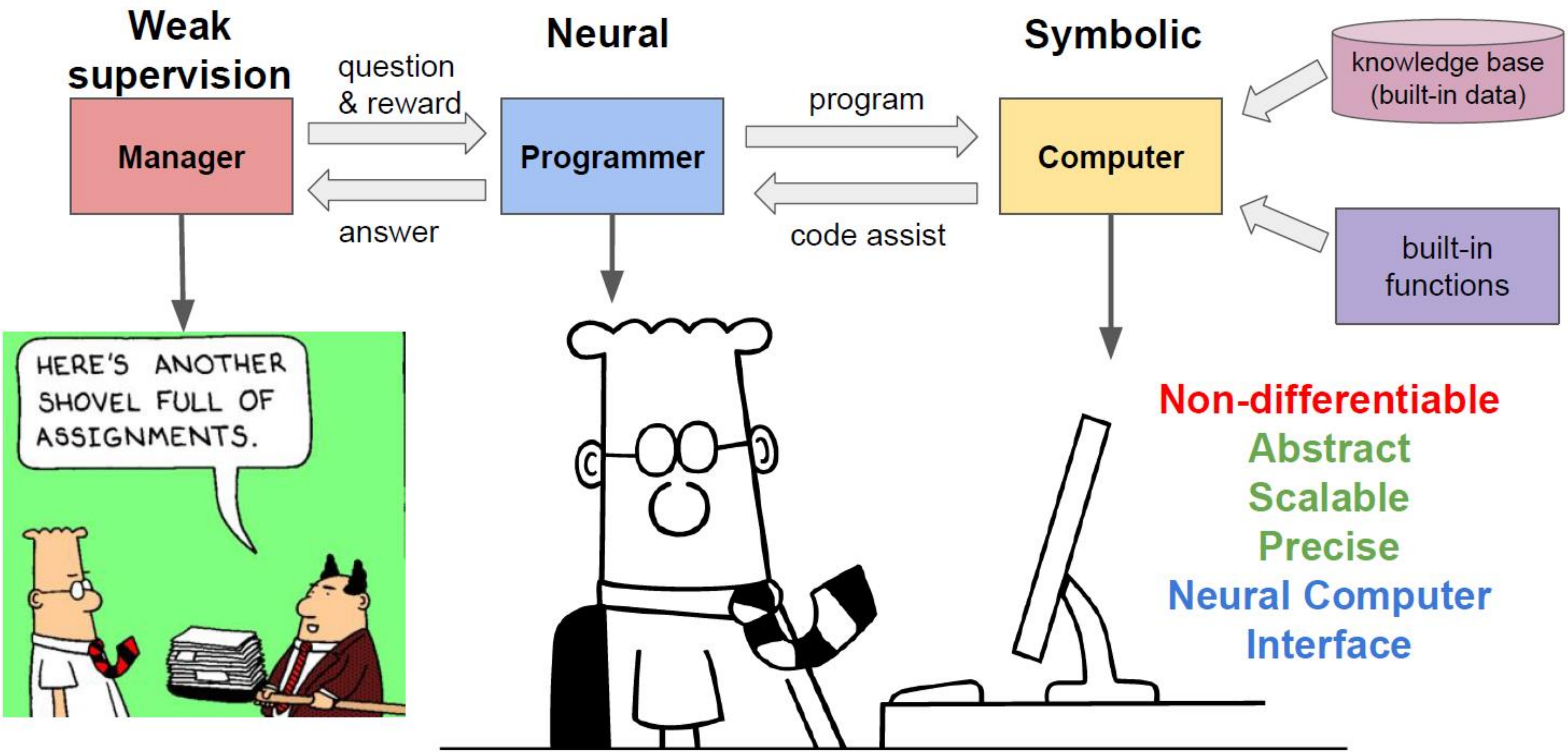


Figure 2: Overview of an Executor-ℓ

Neural Symbolic Machines

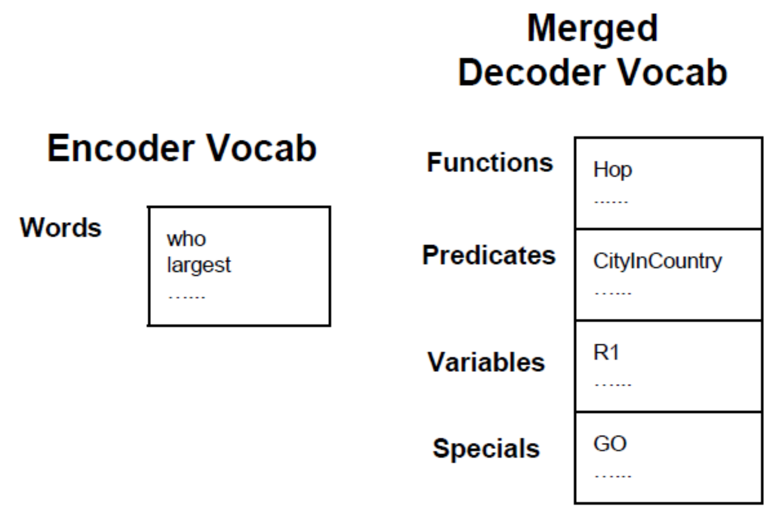
- Liang et al., (2017)
- Contains Manager, Neural Programmers and Symbolic Computer
- Manager: provides weak supervision using RL
- Neural Programmer: Seq2Seq with key-variable memory
- Symbolic Computer: Lisp interpreter that performs program execution

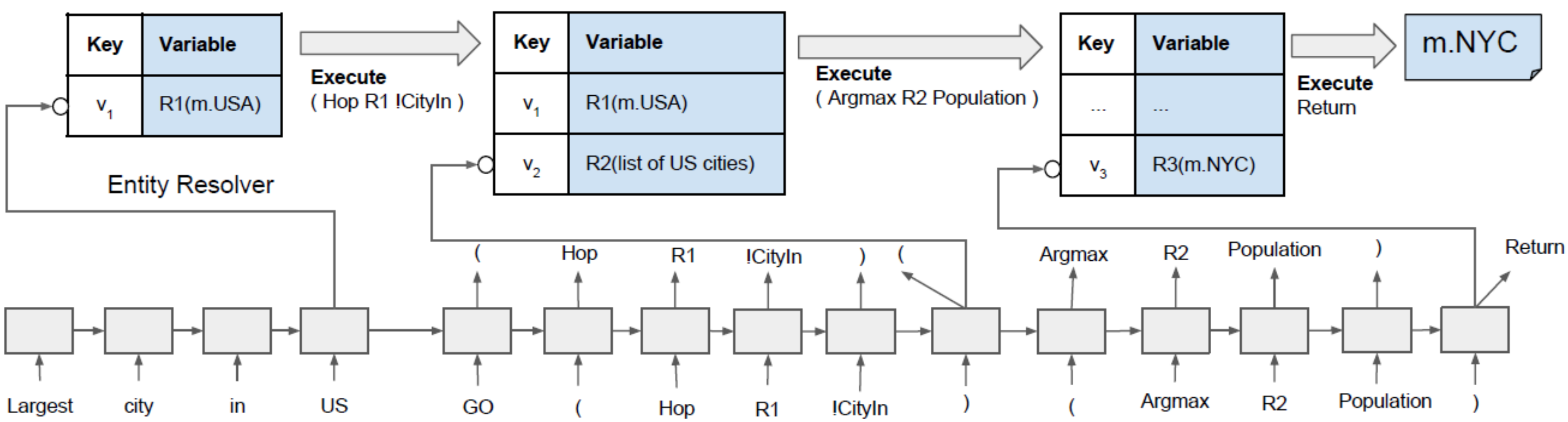
The MPC Framework



Semantic Parsing with NSM

- Add a **key-variable memory** to Seq2Seq model for compositionality
- The **'keys'** are the output of GRUs
- The **'variables'** are just symbols referencing results in computer: 'R1', 'R2'





Seq2SQL

- Zhong et al., (2017)
- DNN translating NL questions to SQL queries
- Uses rewards over database to learn policy to generate query
- Three component networks
  - Aggregation classifier
  - SELECT column pointer
  - WHERE clause pointer decoder
- Provides the dataset, WikiSQL

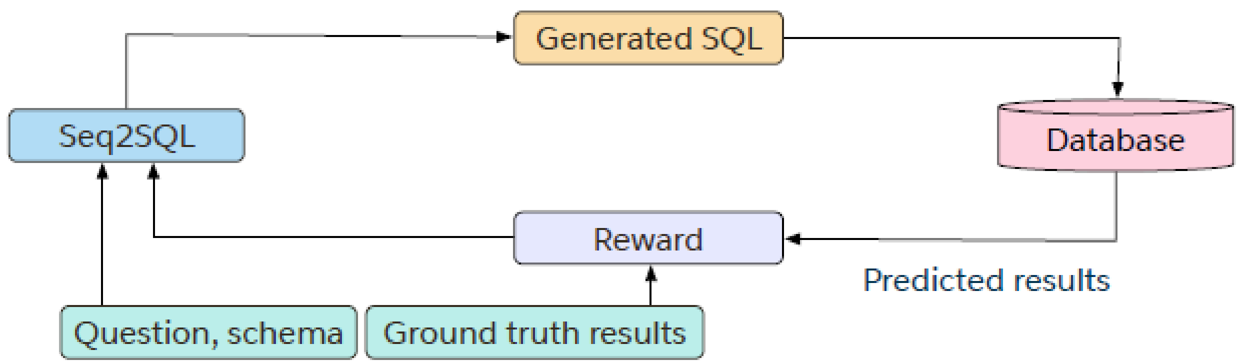


Figure 1: Seq2SQL takes as input a question and the columns of a table. It generates the corresponding SQL query, which, during training, is executed against a database. The result of the execution is utilized as the reward to train the reinforcement learning algorithm.

Table: CFLDraft

Pick #	CFL Team	Player	Position	College
27	Hamilton Tiger-Cats	Connor Healy	DB	Wilfrid Laurier
28	Calgary Stampeders	Anthony Forgone	OL	York
29	Ottawa Renegades	L.P. Ladouceur	DT	California
30	Toronto Argonauts	Frank Hoffman	DL	York
...	...	...	...	...

Question:

How many CFL teams are from York College?

SQL:

```
SELECT COUNT CFL Team FROM
CFLDraft WHERE College = “York”
```

Result:

2

Figure 2: An example in WikiSQL. The inputs consist of a table and a question. The outputs consist of a ground truth SQL query and the corresponding result from execution.

Three Components

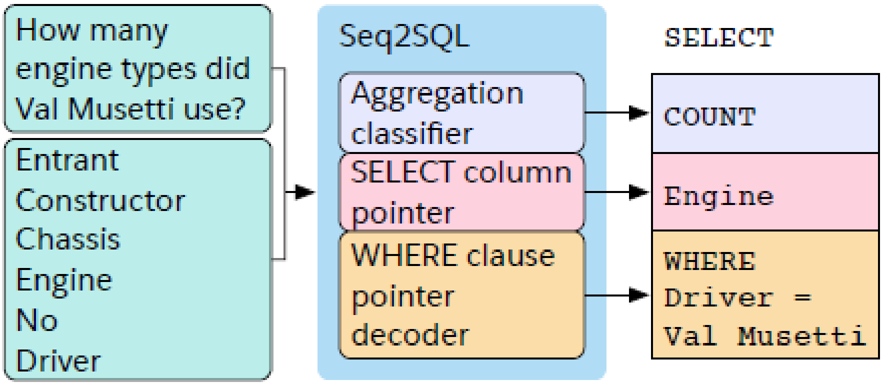


Figure 3: The Seq2SQL model has three components, corresponding to the three parts of a SQL query (right).

$$R\left(q\left(y\right), q_g\right) = \begin{cases} -2, & \text{if } q\left(y\right) \text{ is not a valid SQL query} \\ -1, & \text{if } q\left(y\right) \text{ is a valid SQL query and executes to an incorrect result} \\ +1, & \text{if } q\left(y\right) \text{ is a valid SQL query and executes to the correct result} \end{cases}$$

Recent Studies

Model	RNN-type	Data	REINFORCE
Seq2Tree	LSTM	Geo, Atis, IFTTT	NA
Neural Programmer	LSTM	WikiTableQuestions	Fail
Neural Enquirer	GRU	MIXED dataset	N/A
Mou et al., (2017)	GRU?	MIXED dataset	Success
NSM	GRU	WebQuestionsSP	Success
Seq2SQL	LSTM, PTR	WikiSQL	Success

Summary

- 과거에는 규칙이나 문법에 의존하여 수행하던 질의어 시맨틱 파싱에 RNN 기반 심층 신경망 모형을 응용하려는 시도가 증가하고 있다.
- 더 나아가 강화학습 메커니즘을 시맨틱 파싱의 일부 과정에 이용하여 파싱의 정확성을 올리는데 사용하고 있다.
- 자연어 쿼리 인코딩에는 주로 Seq2Seq 모형이 이용되고 있으나, 디코딩 과정에서 자연어 문장의 위계적 구조를 제대로 파싱하기 위해 기호처리 로직을 모형에 적극 반영하고 있다.
- 강화학습 알고리즘으로 주로 policy gradient 알고리즘을 사용하나, 학습을 위한 조건이 까다로운 것으로 보고하고 있다. → Curriculum learning과 같은 초기학습 과정을 통제하는 식의 방법으로 대처한다.

References

- Andreas, J., Rohrbach, M., Darrell, T., & Klein, D. (2016). Learning to Compose Neural Networks for Question Answering. *ArXiv:1601.01705 [Cs]*.
- Dong, L., & Lapata, M. (2016). Language to Logical Form with Neural Attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 33–43). Berlin, Germany: Association for Computational Linguistics
- Liang, C., Berant, J., Le, Q., Forbus, K. D., & Lao, N. (2017). Neural Symbolic Machines: Learning Semantic Parsers on Freebase with Weak Supervision. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 23–33). Vancouver, Canada: Association for Computational Linguistics.
- Mou, L., Lu, Z., Li, H., & Jin, Z. (2017). Coupling distributed and symbolic execution for natural language queries. In *Proceedings of the 34th International Conference on Machine Learning*. Sydney, Australia.
- Neelakantan, A., Le, Q. V., Abadi, M., McCallum, A., & Amodei, D. (2017). Learning a Natural Language Interface with Neural Programmer. In *International Conference on Learning Representations*. Toulon, France.
- Yin, P., Lu, Z., Li, H., & Ben, kao. (2016). Neural Enquirer: Learning to Query Tables in Natural Language. In *Proceedings of the Workshop on Human-Computer Question Answering* (pp. 29–35). San Diego, California: Association for Computational Linguistics
- Zaremba, W., & Sutskever, I. (2015). Reinforcement Learning Neural Turing Machines - Revised. *ArXiv:1505.00521 [Cs]*.
- Zhong, V., Xiong, C., & Socher, R. (2017). Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning. *ArXiv:1709.00103 [Cs]*.