Coupling distributed and symbolic execution for natural language queries

Lili Mou, Zhengdong Lu, Hang Li, Zhi Jin
doublepower.mou@gmail.com

Outline

- Introduction to neural enquirers
- Coupled approach of neural and symbolic execution
- Experimental results
- Conclusion and discussion

Semantic Parsing

Query:

How long is the game with the largest host country size?

Knowledge base (table):

Year	City	...	Area	...	Duration
...					
2000	Sydney	...	200	...	30
2004	Athens	...	250	...	20
2008	Beijing	...	350	...	25
2012	London	...	300	...	35
2016	Rio de Janeiro	...	200	...	40
...					

```
select Duration where
                area = max(area)
```

Approaches

- Traditional semantic parsing
- seq2seq models
- Neural execution
  - Fully distributed model
  - Symbolic execution

Why “Execution” is Necessary?

Q: How long is the game with the largest host country size?

Year	City	Area	Duration
2000	Sydney	200	30
2004	Athens	250	20
2008	Beijing	350	25
2012	London	300	35
2016	Rio de Janeiro	200	40

Why “Execution” is Necessary?

Q: How long is the game with the largest host country size?

Year	City	**Area**	Duration
2000	Sydney	200	30
2004	Athens	250	20
2008	Beijing	350	25
2012	London	300	35
2016	Rio de Janeiro	200	40

Why “Execution” is Necessary?

Q: How long is the game with the largest host country size?

Year	City	Area	Duration
2000	Sydney	200	30
2004	Athens	250	20
2008	Beijing	350	25
2012	London	300	35
2016	Rio de Janeiro	200	40

Why “Execution” is Necessary?

Q: How long is the game with the largest host country size?

Year	City	Area	**Duration**
2000	Sydney	200	30
2004	Athens	250	20
2008	Beijing	350	25
2012	London	300	35
2016	Rio de Janeiro	200	40

Why “Execution” is Necessary?

- Think of a more complicated example:
  - “How long is the last game which has smaller country size than the game whose host country GDP is 250?”
- Such compositionality of queries necessitates multiple steps of execution.

Neural Enquirer (Yin et al., 2016)

- Everything is an embedding and everything is done by neural information processing
- Differentiable => High learning efficiency
- Low execution efficiency because of neural information processing
- Low interpretability

Neural Symbolic Machine
(Liang et al., 2016)

- Discrete operators
- Differentiable controller
- REINFORCE algorithm (“trial-and-error”)

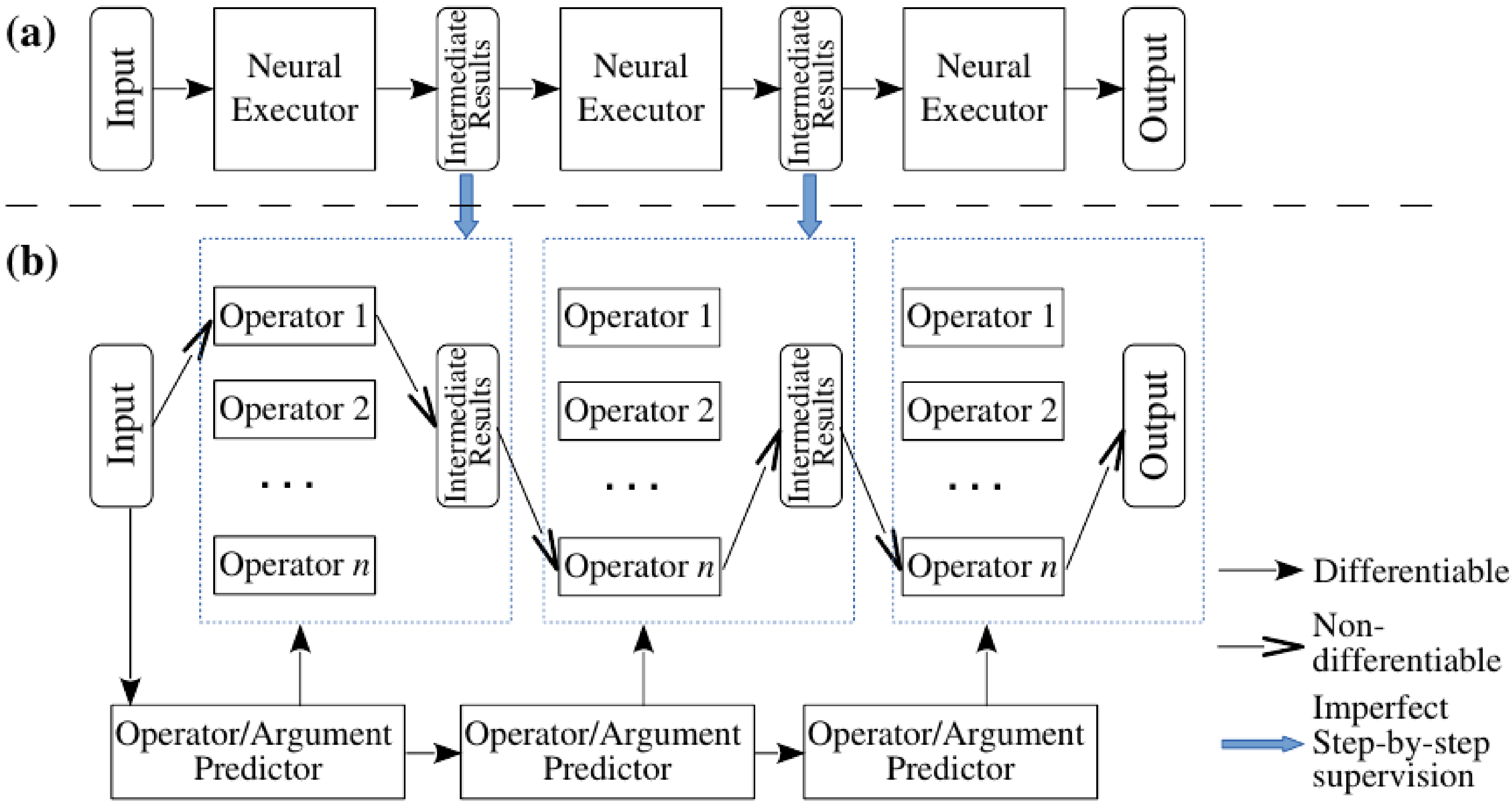
Comparison

	Neuralized	Symbolic	Wanted (Our approach)
Learning efficiency	High	Very low	(Comparatively) High
Execution efficiency	Low	High	High
Interpretability	Low	High	High
Performance	Low	Low	High

Outline

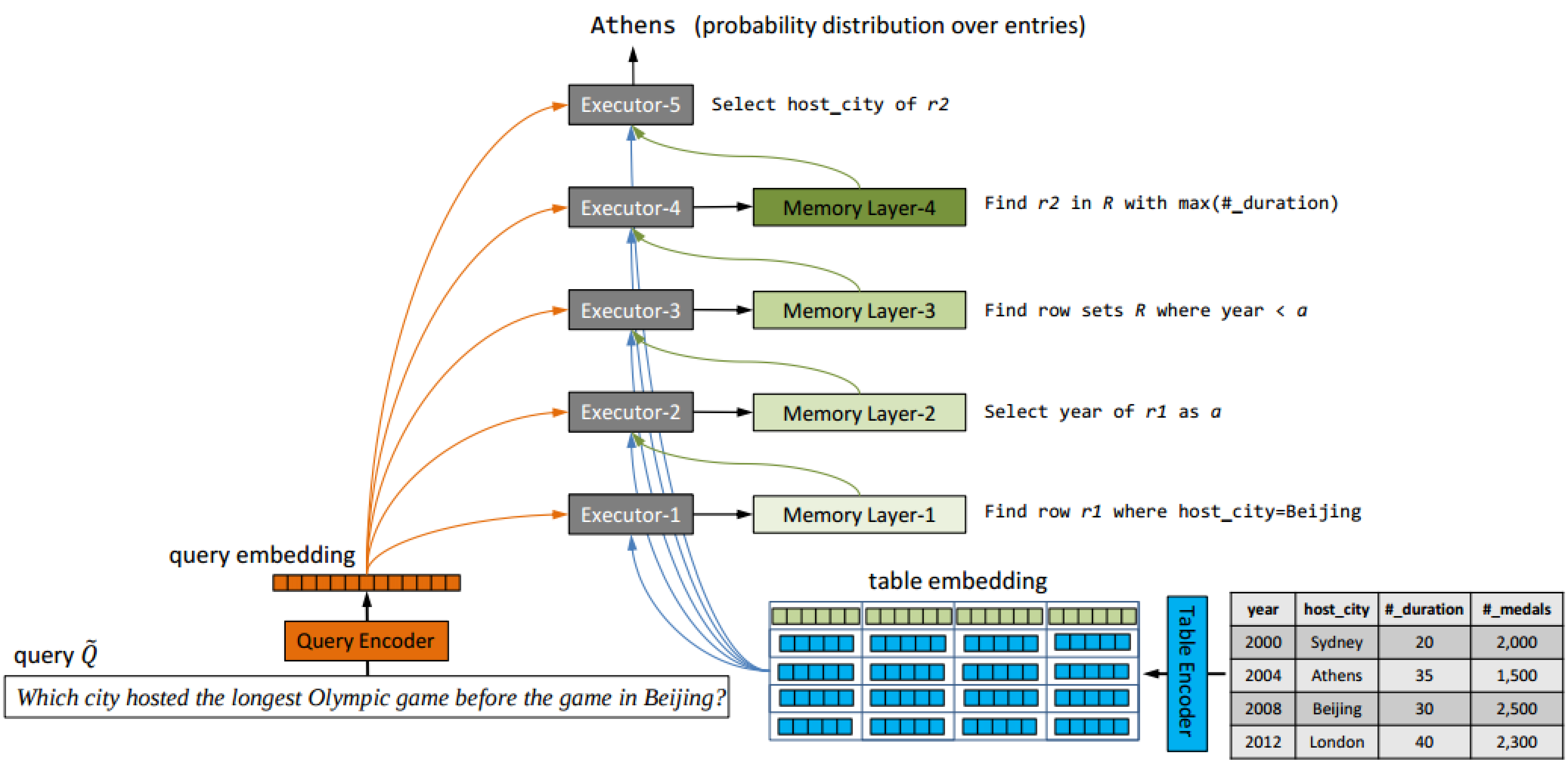
- Introduction to neural enquirers
- Coupled approach of neural and symbolic execution
  - Distributed enquirer
  - Symbolic executor
  - A Unified View
- Experimental results
- Conclusion and discussion

Overview


(a)
Input
Neural Executor
Intermediate Results
Neural Executor
Intermediate Results
Neural Executor
Output
(b)
Input
Operator 1
Operator 2
. . .
Operator n
Intermediate Results
Operator 1
Operator 2
. . .
Operator n
Intermediate Results
Operator 1
Operator 2
. . .
Operator n
Output
Operator/Argument Predictor
Operator/Argument Predictor
Operator/Argument Predictor
Differentiable
Non-differentiable
Imperfect Step-by-step supervision

Distributed Enquirer (Yin et al., 2016)

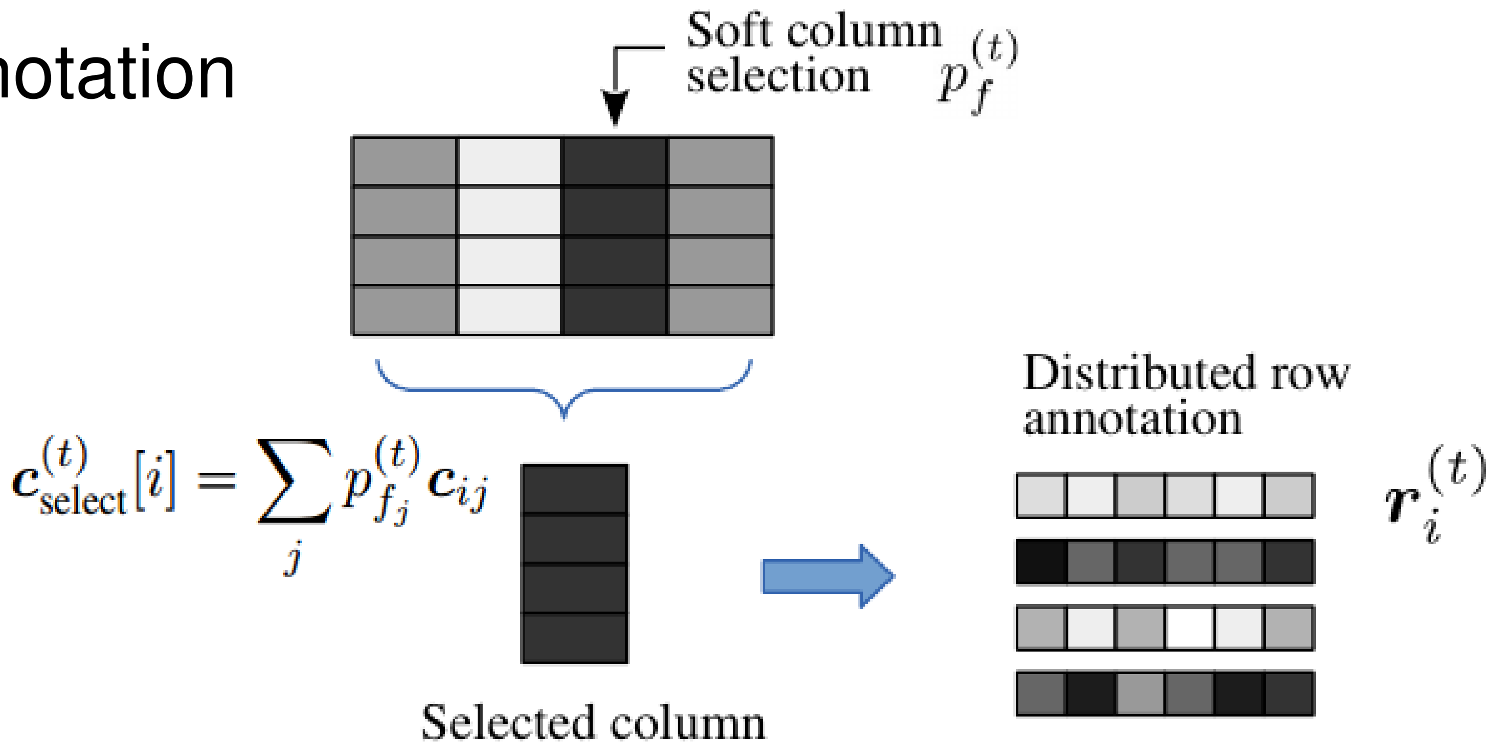
- Query encoder
  - Bidirectional RNN
- Table encoder
  - Concatenation of cell and field embeddings
  - Further processed by a multi-layer perceptron
- Executor
  - Column attention (soft selection)
  - Row annotation (distributed selection)



Athens (probability distribution over entries)
Executor-5
Select host_city of r2
Executor-4
Memory Layer-4
Find r2 in R with max(#_duration)
Executor-3
Memory Layer-3
Find row sets R where year < a
Executor-2
Memory Layer-2
Select year of r1 as a
Executor-1
Memory Layer-1
Find row r1 where host_city=Beijing
query embedding
Query Encoder
query $\tilde{Q}$
Which city hosted the longest Olympic game before the game in Beijing?
table embedding
Table Encoder
year host_city #_duration #_medals
2000 Sydney 20 2,000
2004 Athens 35 1,500
2008 Beijing 30 2,500
2012 London 40 2,300

Executor

- The result of one-step execution softmax attention over columns and a distributed
  - Column attention
  - Row annotation



Details

- Let $\boldsymbol{r}_i^{(t-1)}$ be the previous step's row annotation results, where the subscript i indexes a particular row. We summa-

- Last step's execution information

$$\boldsymbol{g}^{(t-1)} = \mathrm{MaxPool}_i \left\{ \boldsymbol{r}_i^{(t-1)} \right\}$$

- Current step
  - Column attention $p_{f_j}^{(t)} = \mathrm{softmax}\left(\mathrm{MLP}\left([\boldsymbol{q}; \boldsymbol{f}_j; \boldsymbol{g}^{(t-1)}]\right)\right)$
  - Row annotation

$$\boldsymbol{c}_{\mathrm{select}}^{(t)}[i] = \sum_j p_{f_j}^{(t)} \boldsymbol{c}_{ij}$$

$$\boldsymbol{r}_i^{(t)} = \mathrm{MLP}\left(\left[\boldsymbol{q}, \boldsymbol{g}^{(t-1)}, \boldsymbol{r}^{(t-1)}, \boldsymbol{c}_{\mathrm{select}}^{(t)}[i]\right]\right)$$

Symbolic Execution

- Intuition: A more natural way for semantic parsing is symbolic execution
  - E.g., `max`(.), `less_than`(.)

- Methodology
  - Primitive operators
  - Controller (operator/argument predictor)

Primitive Operators

Operator	**Explanation**
`select_row`	Choose a row whose value of a particular column is mentioned in the query
`argmin`	Choose the row from previously selected candidate rows with the minimum value in a particular column
`argmin`	Choose the row from previously selected candidate rows with the maximum value in a particular column
`greater_than`	Choose rows whose value in a particular column is greater than a previously selected row
`less_than`	Choose rows whose value in a particular column is less than a previously selected row
`select_value`	Choose the value of a particular column and of the previously selected row
`EOE`	Terminate, indicating the end of execution

Example

Q: How long is the game with the largest host country size?

Year	**City**	**Area**	**Duration**
2000	Sydney	200	30
2004	Athens	250	20
2008	Beijing	350	25
2012	London	300	35
2016	Rio de Janeiro	200	40

operator = `argmax`   field = `Area`

Example

Q: How long is the game with the largest host country size?

Year	City	**Area**	Duration
2000	Sydney	200	30
2004	Athens	250	20
2008	Beijing	350	25
2012	London	300	35
2016	Rio de Janeiro	200	40

operator = `argmax` field = `Area`

Example

Q: How long is the game with the largest host country size?

Year	City	Area	Duration
2000	Sydney	200	30
2004	Athens	250	20
2008	Beijing	350	25
2012	London	300	35
2016	Rio de Janeiro	200	40

operator = `select_value`   field = `Duration`

Example

Q: How long is the game with the largest host country size?

Year	City	Area	**Duration**
2000	Sydney	200	30
2004	Athens	250	20
2008	Beijing	350	25
2012	London	300	35
2016	Rio de Janeiro	200	40

operator = $\texttt{EOE}$

A More Complicated Example

- Q: How long is the last game which has smaller country size than the game whose host country GDP is 250?

1. `select_row`: select the row where the column is *GDP* and the value is mentioned in the query.
2. `less_than`: select rows whose country size is less than that of the previously selected row.
3. `argmax`: select the row whose year is the largest among previously selected rows.
4. `select_value`: choose the value of the previously selected row with the column being *Duration*.

Controller: Operator/Argument Predictor

- Jordan-type RNNs
  - Operator predictor

$$\boldsymbol{h}_{\mathrm{op}}^{(t-1)} = \mathrm{sigmoid}(W_{\mathrm{op}}^{(\mathrm{rec})} \boldsymbol{h}_{\mathrm{op}}^{(t-1)})$$

$$p_{\mathrm{op}_i}^{(t)} = \mathrm{softmax}\left\{ \boldsymbol{w}_{\mathrm{op}_i}^{(\mathrm{out})\top} \boldsymbol{h}_{\mathrm{op}}^{(t-1)} \right\}$$

  - Field predictor

$$\boldsymbol{h}_{\mathrm{field}}^{(t-1)} = \mathrm{sigmoid}(W_{\mathrm{field}}^{(\mathrm{rec})} \boldsymbol{h}_{\mathrm{field}}^{(t-1)})$$

$$p_{f_j}^{(t)} = \mathrm{softmax}\left\{ \boldsymbol{f}_j^{\top} \boldsymbol{h}_{\mathrm{field}}^{(t-1)} \right\}$$

The Problems

- Non-differentiable
- No step-by-step supervision

A Unified View

- Two worlds of execution
  - Fully neuralized enquirer
    - End-to-end learnable
  - Symbolic enquirer
    - High execution efficiency
    - High interpretability
- We propose to take advantage of the both worlds
  - Plus high performance

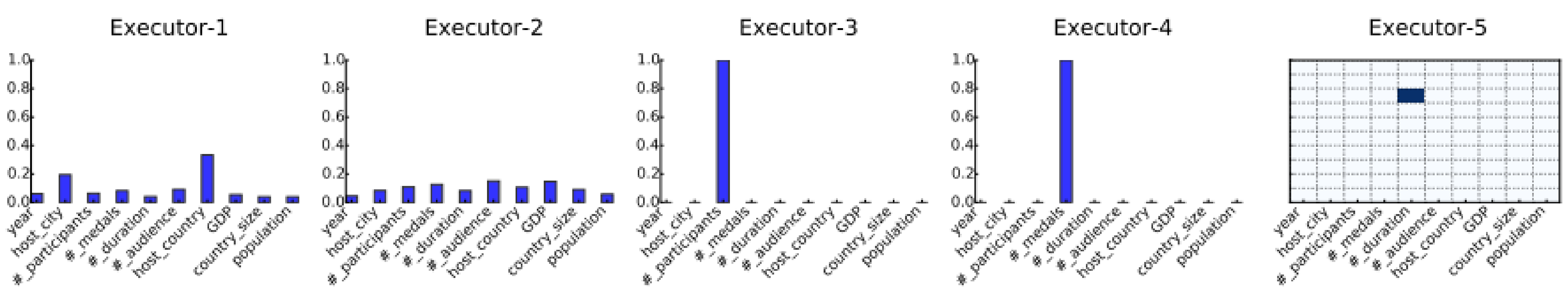
Intuition

- The fully neuralized enquirer also exhibits some (imperfect) interpretability
- The field attention generally aligns with column selection

$$p_{f_j}^{(t)} = \mathrm{softmax}\left(\mathrm{MLP}\left([\boldsymbol{q}; \boldsymbol{f}_j; \boldsymbol{g}^{(t-1)}]\right)\right)$$

$$p_{f_j}^{(t)} = \mathrm{softmax}\left\{\boldsymbol{f}_j^\top \boldsymbol{h}_{\text{field}}^{(t-1)}\right\}$$

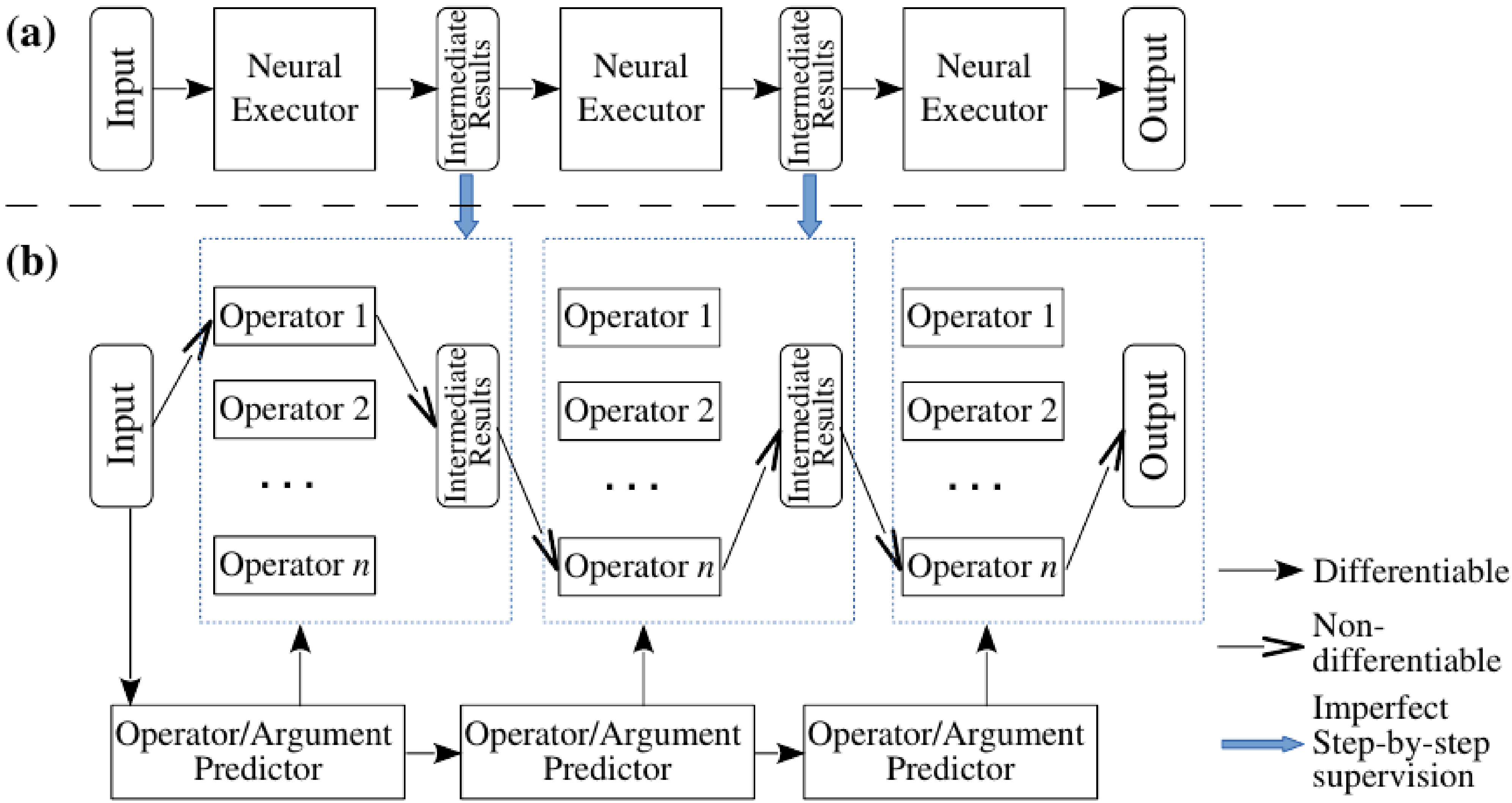
Q_5: *How long is the game with the most medals that has fewer than 3,000 participants?*



Solution

- Use neural networks' (imperfect) intermediate results to pretrain the symbolic executor's policy in a step-by-step fashion
- Improve the policy by reinforcement learning

Overview


(a)
Input
Neural Executor
Intermediate Results
Neural Executor
Intermediate Results
Neural Executor
Output
(b)
Input
Operator 1
Operator 2
...
Operator n
Intermediate Results
Operator 1
Operator 2
...
Operator n
Intermediate Results
Operator 1
Operator 2
...
Operator n
Output
Operator/Argument Predictor
Operator/Argument Predictor
Operator/Argument Predictor
Differentiable
Non-differentiable
Imperfect Step-by-step supervision

Pretraining

- Let *m* be the number of actions to pretrain

Imperfect supervision signal

$$J = -\sum_{i=1}^{m} \sum_{j=1}^{n_{\text{label}}^{(i)}} \hat{t}_j^{(i)} \log p_j^{(i)}$$

Step-by-step supervision

REINFORCE Policy Improvement

- $J = -\mathbb{E}_{a_1, a_2, \cdots, a_n \sim \theta}[R(a_1, a_2, \cdots, a_n)]$
- Gradient $$\frac{\partial J}{\partial \boldsymbol{o}_i} = \tilde{R} \cdot (\boldsymbol{p}_i - \mathbf{1}_{a_i})$$
- Reward R: 1=correct result, 0 = incorrect result
- Tricks
  - Exploring with a small probability (0.1)
  - Subtracting the mean (reinforcement comparison)
  - Truncate negative reward (reward-inaction)

REINFORCE

Ranzato, Marc'Aurelio, et al. "Sequence Level Training with Recurrent Neural Networks." ICLR, 2016.

- Define an external cost function on a generated sequence
- Generate words by sampling
- Take the derivative of generated samples

$$L_\theta = -\sum_{w_1^g,\ldots,w_T^g} p_\theta(w_1^g,\ldots,w_T^g) r(w_1^g,\ldots,w_T^g) = -\mathbb{E}_{[w_1^g,\ldots w_T^g]\sim p_\theta} r(w_1^g,\ldots,w_T^g)$$

∂p(w)=p(w)∂log p(w) because p(w)=exp{log p(w)}

- $\partial J = \sum_{\mathbf{w}} [\partial\, p(\mathbf{w}|\ldots)]\, r(\mathbf{w}) = \sum_{\mathbf{w}} p(\mathbf{w})[\partial \log p(\mathbf{w})]\, r(\mathbf{w})$

||

$$\left(p_\theta(w_{t+1}|w_t^g, \mathbf{h}_{t+1}, \mathbf{c}_t) - \mathbf{1}(w_{t+1}^g)\right)$$

where $\mathbf{o}_t$ is the input to the softmax.

REINFORCE in a nutshell

- Sample from your current policy distribution
- Obtain the reward
- Update according to the gradient of the sampled actions
  - Extremely difficult to get started
  - Poor local optima (also sensitive to initial policy)
  - Fortunately, the distributed worlds makes life much easier.

Outline

- Introduction to neural enquirers
- Coupled approach of neural and symbolic execution
- Experimental results
- Conclusion and discussion

Experimental Settings

- Dataset: Yin et al. (2016)
  - Synthesized data
  - 25k samples (different queries and tables)
- Hyperparamters
  - Mostly derived from previous work
  - 40 epochs of pretraining before REINFORCE

Experiments

- Performance

Query type	SEMPRE†	Denotation Distributed†	Symbolic	Coupled
SelectWhere	93.8	96.2	99.2	**99.6**
Superlative	97.8	98.9	**100.0**	**100.0**
WhereSuperlative	34.8	80.4	51.9	**99.9**
NestQuery	34.4	60.5	52.5	**100.0**
Overall	65.2	84.0	75.8	**99.8**

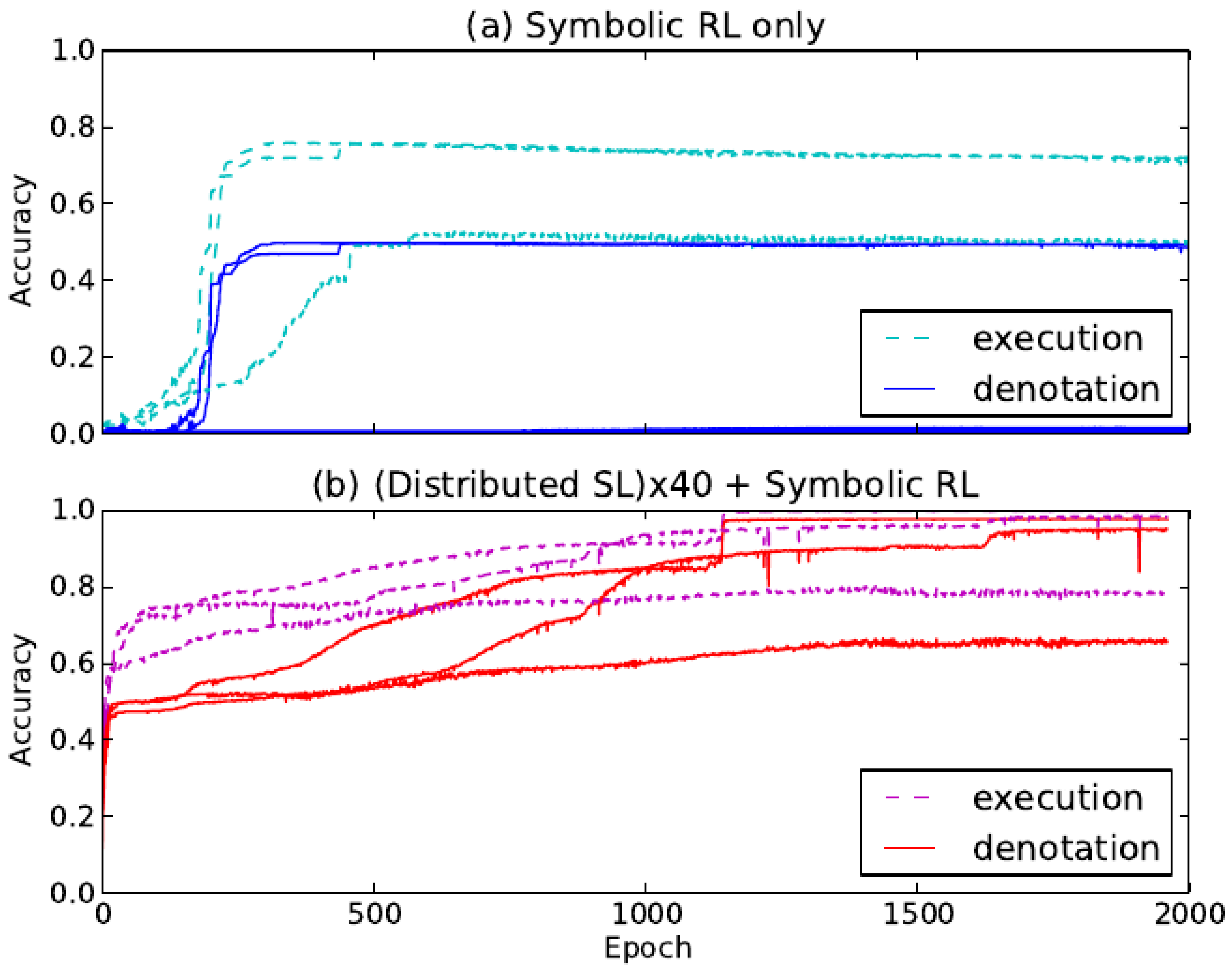
Experiments

- Interpretability

Query type	SEMPRE†	Denotation Distributed†	Denotation Symbolic	Denotation Coupled	Execution Distributed	Execution Symbolic	Execution Coupled
SelectWhere	93.8	96.2	99.2	**99.6**	–	99.1	**99.6**
Superlative	97.8	98.9	**100.0**	**100.0**	–	**100.0**	**100.0**
WhereSuperlative	34.8	80.4	51.9	**99.9**	–	0.0	**91.0**
NestQuery	34.4	60.5	52.5	**100.0**	–	0.0	**100.0**
Overall	65.2	84.0	75.8	**99.8**	–	49.5	**97.6**

Experiments

- Learning efficiency



Experiments

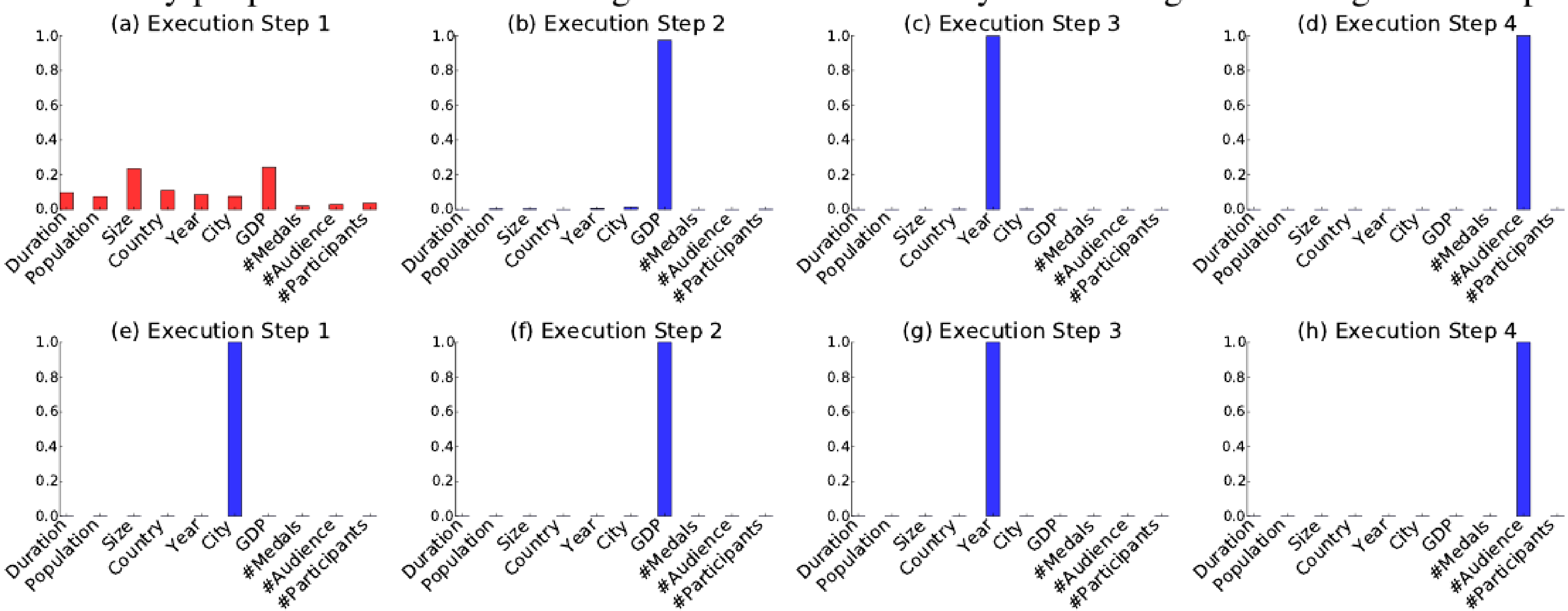
- Execution efficiency

	Fully Distributed	**Our approach**		
		Op/Arg Pred.	Symbolic Exe.†	Total
CPU	13.86	2.65	0.002	2.65
GPU	1.05	0.44		0.44

Feeding back/Co-training

Training Method	Accuracy (%)
End-to-end (w/ denotation labels)†	84.0
Step-by-step (w/ execution labels)†	96.4
Feeding back	96.5

Query: How many people watched the earliest game whose host country GDP is larger than the game in Cape Town?



Outline

- Introduction to neural enquirer
- Coupled approach of neural and symbolic execution
- Conclusion and discussion

Conclusion

- Propose to couple distributed and symbolic execution for natural language queries
- The distributed enquirer exhibits some (imperfect) interpretability
- We use the distributed model's step-by-step signal to pretrain the symbolic one to acquire a fairly meaningful initial policy.
- Improve the policy by REINFORCE.
- The coupled model achieves high learning efficiency, high execution efficiency, high interpretability, as well as high performance.

Future Work

- Couple more actions
- Better use the information
  - Using the full distribution information?
  - Sampling from the distribution predicted by the neural enquirer?
  - Inducing operators and pretraining the operation predictors

Discussion

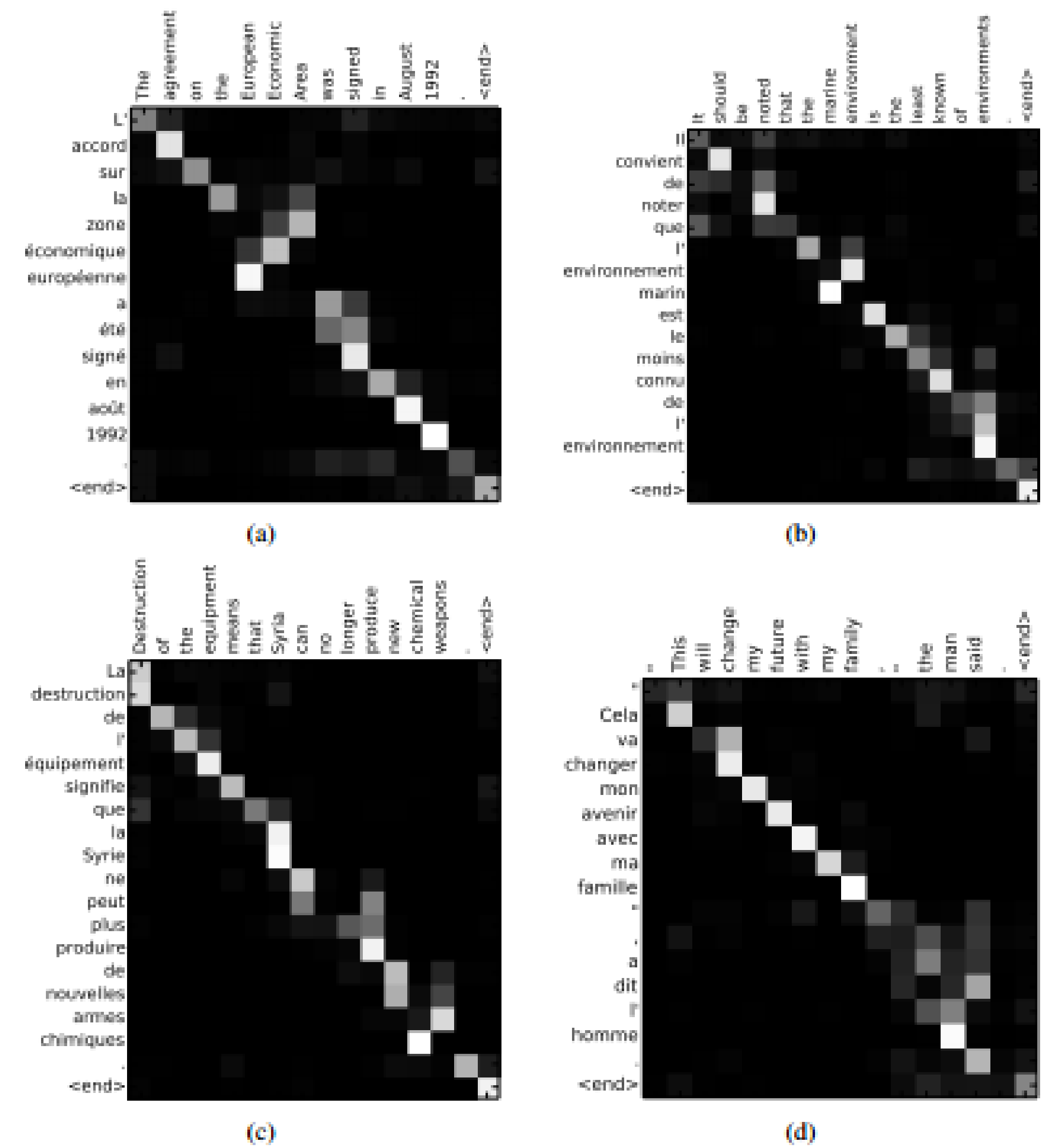
- Previous work on incorporating neural networks with external (somewhat) symbolic systems
  - Hu et al. (2016) harness knowledge of a rule-based system by inducing a probability distribution from it as the training objective.
  - Lei et al. (2016) propose a sparse, hard gating approach to force a neural network to focus on relevant information.
  - Mi et al. (2016) use alignment heuristics to train the attention signal of neural machine translation in a supervised manner.

The Uniqueness of Our Work

- First train a neural network in an end-to-end fashion
- Then guide a symbolic system to achieve a meaningful initial policy

Attention as Alignment

(Bahdanau et al., ICLR 2015)



Q & A?

Thank you for listening!