An Imitation Game for Learning Semantic Parsers from User Interaction

Ziyu Yao[†], Yiqi Tang[†], Scott Yih[*], Huan Sun[†], Yu Su[†]

[†]The Ohio State University
[*]Facebook AI Research

EMNLP 2020

THE OHIO STATE UNIVERSITY

facebook
Artificial Intelligence Research

Semantic Parsing

- Parsing natural language (NL) to formal meaning representations

- Example: Text-to-SQL semantic parsing

Question:

How many CFL teams are from York College?

SQL:

```
SELECT COUNT CFL Team FROM
CFLDraft WHERE College = "York"
```

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
...	...	...	...	...

Result:

2

From WikiSQL [Zhong et al., 2017]

The Life Cycle of Semantic Parsers

- Bootstrapping

- Fine-tuning

The Life Cycle of Semantic Parsers

- Bootstrapping
  - Collect NL-semantic parse data from annotators
  - Train model to commercial-grade performance (e.g., 95% acc on a test set)
  - Semantic parsers: data-hungry; expertise required
- Fine-tuning

The Life Cycle of Semantic Parsers

- Bootstrapping
  - Collect NL-semantic parse data from annotators
  - Train model to commercial-grade performance (e.g., 95% acc on a test set)
  - Semantic parsers: data-hungry; expertise required

- Fine-tuning
  - After deployment
  - Continually analyze usage and collect new training data for emerging user needs

The Life Cycle of Semantic Parsers

- Bootstrapping
  - Collect NL-semantic parse data from annotators
  - Train model to commercial-grade performance (e.g., 95% acc on a test set)
  - Semantic parsers: data-hungry; expertise required

- Fine-tuning
  - After deployment
  - Continually analyze usage and collect new training data for emerging user needs

*PLUS: **privacy risks** when exposing user data to external developers*

This Work

- Learning semantic parsers with human users in the loop

This Work

- Learning semantic parsers with human users in the loop
  - Semantic parser as an intelligent agent:
    - being introspective of its uncertainties
    - prompt for user interaction
    - continually accumulate user feedback, improve itself and adapt for user needs

This Work

- Learning semantic parsers with human users in the loop
  - Semantic parser as an intelligent agent:
    - being introspective of its uncertainties
    - prompt for user interaction
    - continually accumulate user feedback, improve itself and adapt for user needs

reduced bootstrapping & fine-tuning cost

This Work

- Learning semantic parsers with human users in the loop
  - Semantic parser as an intelligent agent:
    - being introspective of its uncertainties
    - prompt for user interaction
    - continually accumulate user feedback, improve itself and adapt for user needs

reduced bootstrapping & fine-tuning cost

  - User:
    - answer questions covered by their background knowledge
    - get to know more about the parsing system => more trust

This Work

- Learning semantic parsers with human users in the loop
  - Semantic parser as an intelligent agent:
    - being introspective of its uncertainties
    - prompt for user interaction
    - continually accumulate user feedback, improve itself and adapt for user needs

reduced bootstrapping & fine-tuning cost

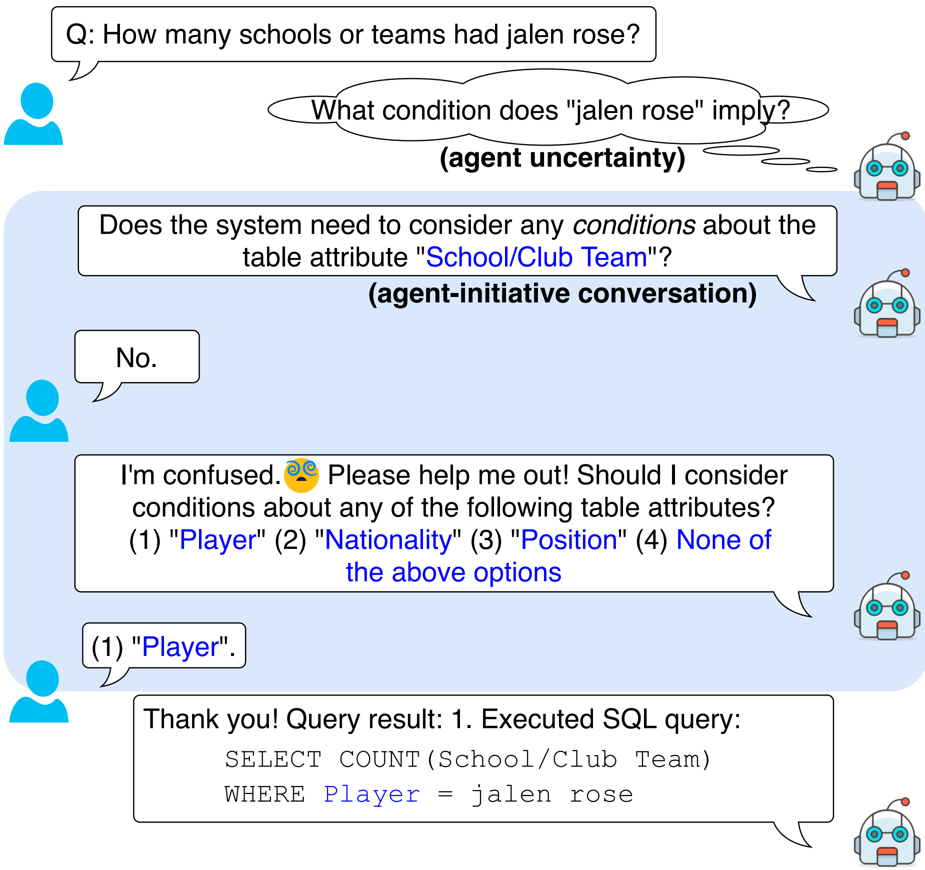
  - User:
    - answer questions covered by their background knowledge
    - get to know more about the parsing system => more trust

reduced privacy risks

MISP-NEIL

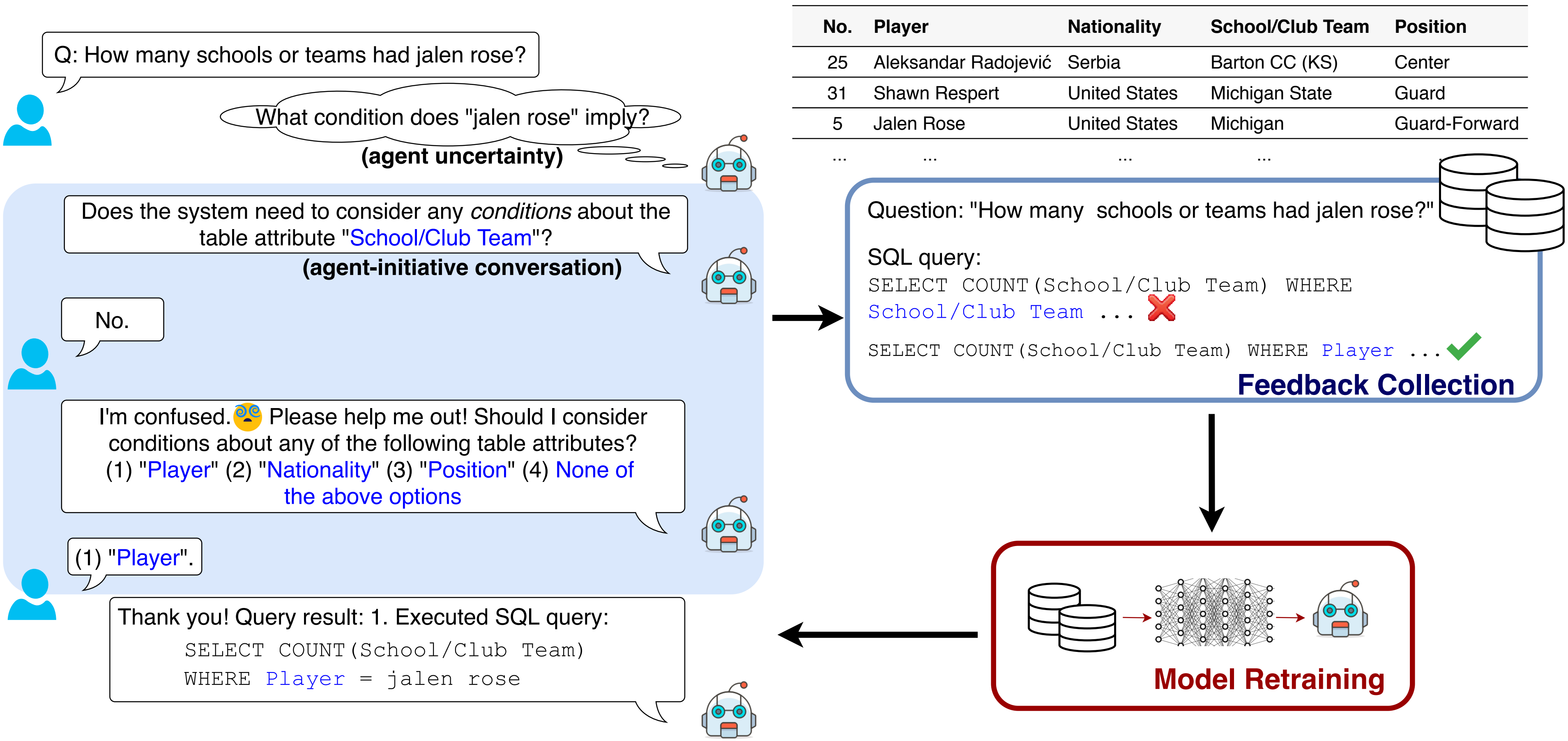
An interactive system that continually trains a semantic parser from **fine-grained user interaction** after deployment.

MISP-NEIL



No.	Player	Nationality	School/Club Team	Position
25	Aleksandar Radojević	Serbia	Barton CC (KS)	Center
31	Shawn Respert	United States	Michigan State	Guard
5	Jalen Rose	United States	Michigan	Guard-Forward

MISP-NEIL



Q: How many schools or teams had jalen rose?
What condition does "jalen rose" imply?
(agent uncertainty)
Does the system need to consider any *conditions* about the table attribute "School/Club Team"?
(agent-initiative conversation)
No.
I'm confused. Please help me out! Should I consider conditions about any of the following table attributes? (1) "Player" (2) "Nationality" (3) "Position" (4) None of the above options
(1) "Player".
Thank you! Query result: 1. Executed SQL query:
SELECT COUNT(School/Club Team) WHERE Player = jalen rose
No. | Player | Nationality | School/Club Team | Position
25 | Aleksandar Radojević | Serbia | Barton CC (KS) | Center
31 | Shawn Respert | United States | Michigan State | Guard
5 | Jalen Rose | United States | Michigan | Guard-Forward
... | ... | ... | ...
Question: "How many schools or teams had jalen rose?"
SQL query:
SELECT COUNT(School/Club Team) WHERE School/Club Team ...
SELECT COUNT(School/Club Team) WHERE Player ...
Feedback Collection
Model Retraining

Outline

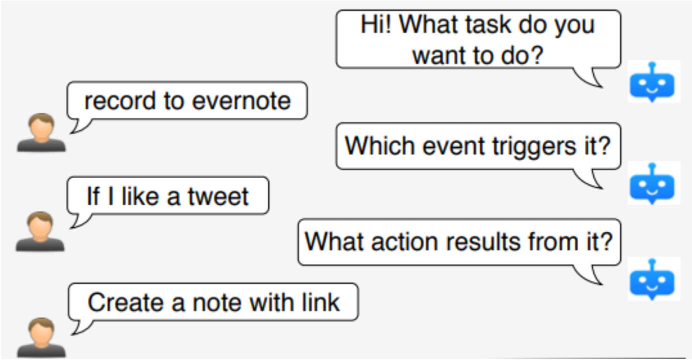
- Introduction
- MISP-NEIL architecture
  - Interactive semantic parsing with **MISP**
  - **NEIL**: aNnotation-Efficient Imitation Learning (with theoretical analysis)
- Experiments
- Future work

Interactive Semantic Parsing

- A recent idea of involving system-user interaction to improve semantic parsing

Interactive Semantic Parsing

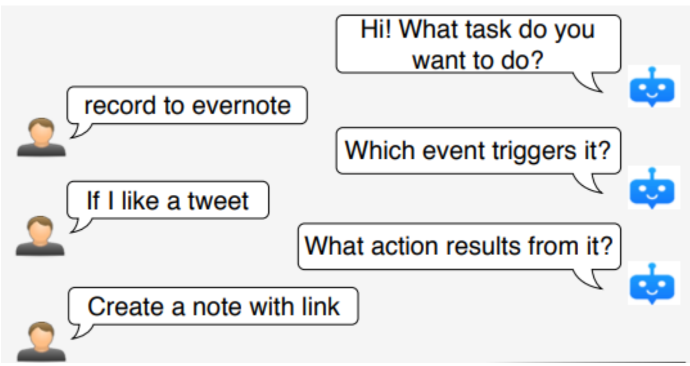
- A recent idea of involving system-user interaction to improve semantic parsing



asking for missing info [Yao et al., 2019a]

Interactive Semantic Parsing

- A recent idea of involving system-user interaction to improve semantic parsing



asking for missing info [Yao et al., 2019a]

User: *Can you create a meeting with Megan right before that starts?*

```
createEvent(EventSpec(
  end=start(refer(Constraint[Event]())),
  attendee=PersonSpec(name='Megan')
))
```

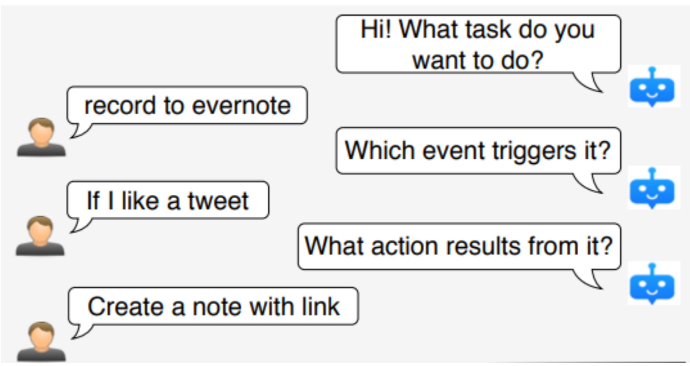
Agent: *Which person named Megan did you mean?*

User: *Megan Bowen.*

disambiguation [Semantic Machines 2020]

Interactive Semantic Parsing

- A recent idea of involving system-user interaction to improve semantic parsing



asking for missing info [Yao et al., 2019a]

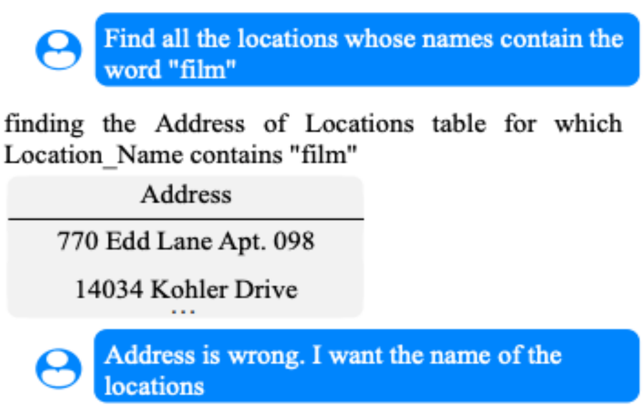
User: *Can you create a meeting with Megan right before that starts?*

```
createEvent(EventSpec(
  end=start(refer(Constraint[Event]())),
  attendee=PersonSpec(name='Megan')
))
```

Agent: *Which person named Megan did you mean?*

User: *Megan Bowen.*

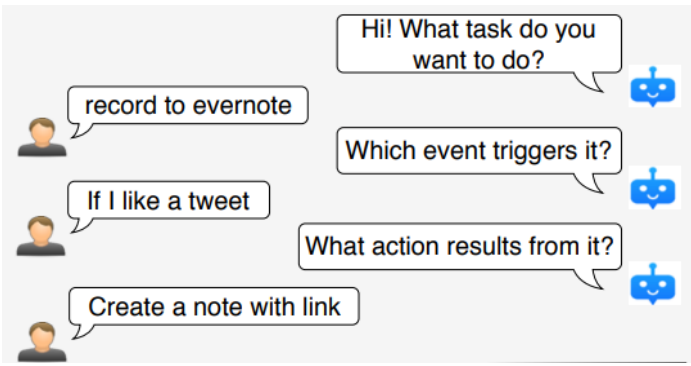
disambiguation [Semantic Machines 2020]



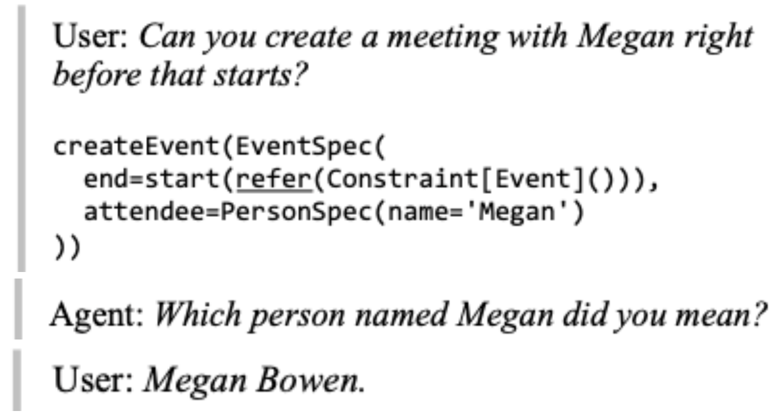
post correction [Elgohary et al., 2020]

Interactive Semantic Parsing

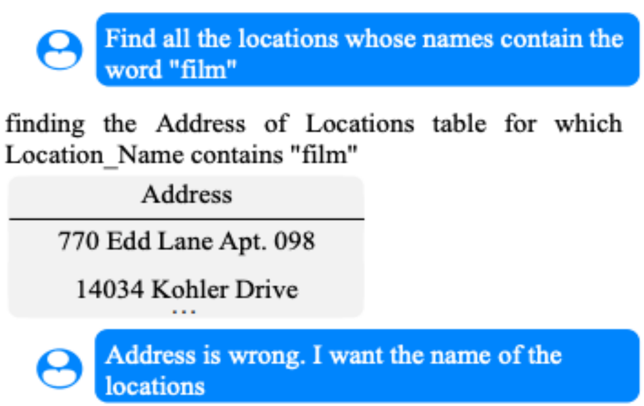
- A recent idea of involving system-user interaction to improve semantic parsing



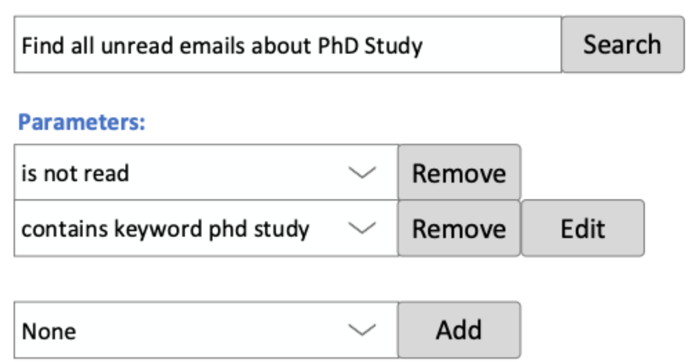
asking for missing info [Yao et al., 2019a]



disambiguation [Semantic Machines 2020]



post correction [Elgohary et al., 2020]



user post edit via GUI [Su et al., 2018]

Interactive Semantic Parsing

- MISP (Model-based Interactive Semantic Parser) [Yao et al., 2019b]
  - A general, unified framework
  - Generalization:
    - can be used with various semantic parser architectures & logical forms
  - User-friendly:
    - fine-grained natural language questions (generally covered by user background knowledge)

Interactive Semantic Parsing

- MISP (Model-based Interactive Semantic Parser) [Yao et al., 2019b]
  - A general, unified framework
  - Generalization:
    - can be used with various semantic parser architectures & logical forms
  - User-friendly:
    - fine-grained natural language questions (generally covered by user background knowledge)

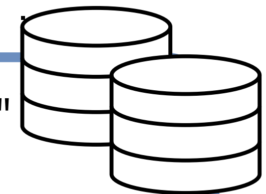
Please refer to [Yao et al., 2019b] for more details.

Open source: https://github.com/sunlab-osu/MISP

Outline

- Introduction
- MISP-NEIL architecture
  - Interactive semantic parsing with **MISP**
  - **NEIL**: aNnotation-Efficient Imitation Learning (with theoretical analysis)
- Experiments
- Future work

Recall: user feedback in **MISP-NEIL**



Question: "How many schools or teams had jalen rose?"

SQL query:

```
SELECT COUNT(School/Club Team) WHERE
School/Club Team ...
```

```
SELECT COUNT(School/Club Team) WHERE Player ...
```

Feedback Collection

predicting the table attribute "Player" after generating the keyword "WHERE" (called "user demonstrations")

NEIL: aNnotation-Efficient Imitation Learning

- Imitation learning: training the semantic parser to *imitate* “user demonstrations” collected during interaction

NEIL: aNnotation-Efficient Imitation Learning

- Imitation learning: training the semantic parser to *imitate* “user demonstrations” collected during interaction

- “annotation-efficient”
  - The agent needs to avoid asking too many questions to the user
  - **Challenge**: *sparse* user demonstrations

NEIL: aNnotation-Efficient Imitation Learning

- Imitation learning: training the semantic parser to *imitate* "user demonstrations" collected during interaction
- "annotation-efficient"
  - The agent needs to avoid asking too many questions to the user
  - **Challenge**: *sparse* user demonstrations
  - **Solution**: collecting both user demonstrations and agent-confident actions (without user validation) as training labels

NEIL: aNnotation-Efficient Imitation Learning

- A DAGGER-liked algorithm [Ross et al., 2011]
  - Iteratively aggregate demonstrations as new training labels and retrain the parser (called *"policy"*)

```
For each iteration i=1 to N:
    Receive user questions {q};
    New training labels ← Parse&Collect(question q, policy_i);
    Aggregate new training labels;
    Train policy_{i+1} on aggregated training data (including the
    pre-training data).
Return the best policy_i on validation.
```

NEIL: aNnotation-Efficient Imitation Learning

- A DAGGER-liked algorithm [Ross et al., 2011]
  - Iteratively aggregate demonstrations as new training labels and retrain the parser (called *"policy"*)

For each iteration `i=1` to `N`:
    Receive user questions `{q}`;
    New training labels ← `Parse&Collect`(question `q`, `policy_i`);
    Aggregate new training labels;
    Train `policy_{i+1}` on aggregated training data (including the pre-training data).
Return the best `policy_i` on validation.

NEIL: aNnotation-Efficient Imitation Learning

- A DAGGER-liked algorithm [Ross et al., 2011]
  - Iteratively aggregate demonstrations as new training labels and retrain the parser (called *"policy"*)

For each iteration `i=1` to `N`:
    Receive user questions `{q}`;
    New training labels ← `Parse&Collect`(question `q`, `policy_i`);
    Aggregate new training labels;
    Train `policy_{i+1}` on aggregated training data (including the pre-training data).
Return the best `policy_i` on validation.

*including user-demonstrated and agent-confident actions

NEIL: aNnotation-Efficient Imitation Learning

- A DAGGER-liked algorithm [Ross et al., 2011]
  - Iteratively aggregate demonstrations as new training labels and retrain the parser (called *"policy"*)

```
For each iteration i=1 to N:
    Receive user questions {q};
    New training labels ← Parse&Collect(question q, policy_i);
    Aggregate new training labels;
    Train policy_{i+1} on aggregated training data (including the
    pre-training data).
Return the best policy_i on validation.
```

NEIL: aNnotation-Efficient Imitation Learning

- A DAGGER-liked algorithm [Ross et al., 2011]
  - Iteratively aggregate demonstrations as new training labels and retrain the parser (called *"policy"*)

```
For each iteration i=1 to N:
    Receive user questions {q};
    New training labels ← Parse&Collect(question q, policy_i);
    Aggregate new training labels;
    Train policy_{i+1} on aggregated training data (including the
    pre-training data).
Return the best policy_i  on validation.
```

Theoretical Analysis

- NEIL is annotation-efficient, but would it lead to much worse semantic parsers?
  - vs. *fully-supervised* approach

Theoretical Analysis

- NEIL is annotation-efficient, but would it lead to much worse semantic parsers?
  - vs. *fully-supervised* approach
- A new "cost function" for semantic parsing tasks
  - smaller cost, better algorithm

Theoretical Analysis

- NEIL is annotation-efficient, but would it lead to much worse semantic parsers?
  - vs. *fully-supervised* approach

- A new "cost function" for semantic parsing tasks
  - smaller cost, better algorithm

Theorem 5.1. *For supervised approach, let* $\epsilon_N = \min_{\pi \in \Pi} \mathbb{E}_{s \sim d_{\pi^*}}[l(s, \pi)]$, *then* $J(\hat{\pi}_{sup}) = T\epsilon_N$.

Theorem 5.2. *For the proposed* NEIL *algorithm, if* N *is* $\tilde{O}(T)$, *there exists a policy* $\hat{\pi} \in \hat{\pi}_{1:N}$ *s.t.* $J(\hat{\pi}) \leq T\left[\epsilon_N + \underline{\frac{2T\ell_{max}}{N}\sum_{i=1}^{N} e_i}\right] + O(1)$.

e_i: probability of confident but wrong actions

Theoretical Analysis

- NEIL is annotation-efficient, but would it lead to much worse semantic parsers?
  - vs. *fully-supervised* approach
- A ne ks
  - sn

Key factors to reduce NEIL's performance loss:

(1) more accurate **confidence estimation**;
=> decision probability with a high confidence threshold

Theorem 5.2. *For the proposed* NEIL *algorithm, if N is $\tilde{O}(T)$, there exists a policy $\hat{\pi} \in \hat{\pi}_{1:N}$ s.t.* $J(\hat{\pi}) \leq T\left[\epsilon_N + \frac{2T\ell_{max}}{N} \sum_{i=1}^{N} e_i\right] + O(1)$.

e_i: probability of confident but wrong actions

Theoretical Analysis

- NEIL is annotation-efficient, but would it lead to much worse semantic parsers?
  - vs. *fully-supervised* approach
- A ne ks
  - sn

Key factors to reduce NEIL's performance loss:

(1) more accurate **confidence estimation**;
=> decision probability with a high confidence threshold

(2) moderate **policy initialization.**
=> verify in experiments

Theorem 5.2. *For the proposed* NEIL *algorithm, if N is $\tilde{O}(T)$, there exists a policy $\hat{\pi} \in \hat{\pi}_{1:N}$ s.t.* $J(\hat{\pi}) \leq T\left[\epsilon_N + \underline{\frac{2T\ell_{max}}{N}\sum_{i=1}^{N} e_i}\right] + O(1)$.

e_i: probability of confident but wrong actions

Outline

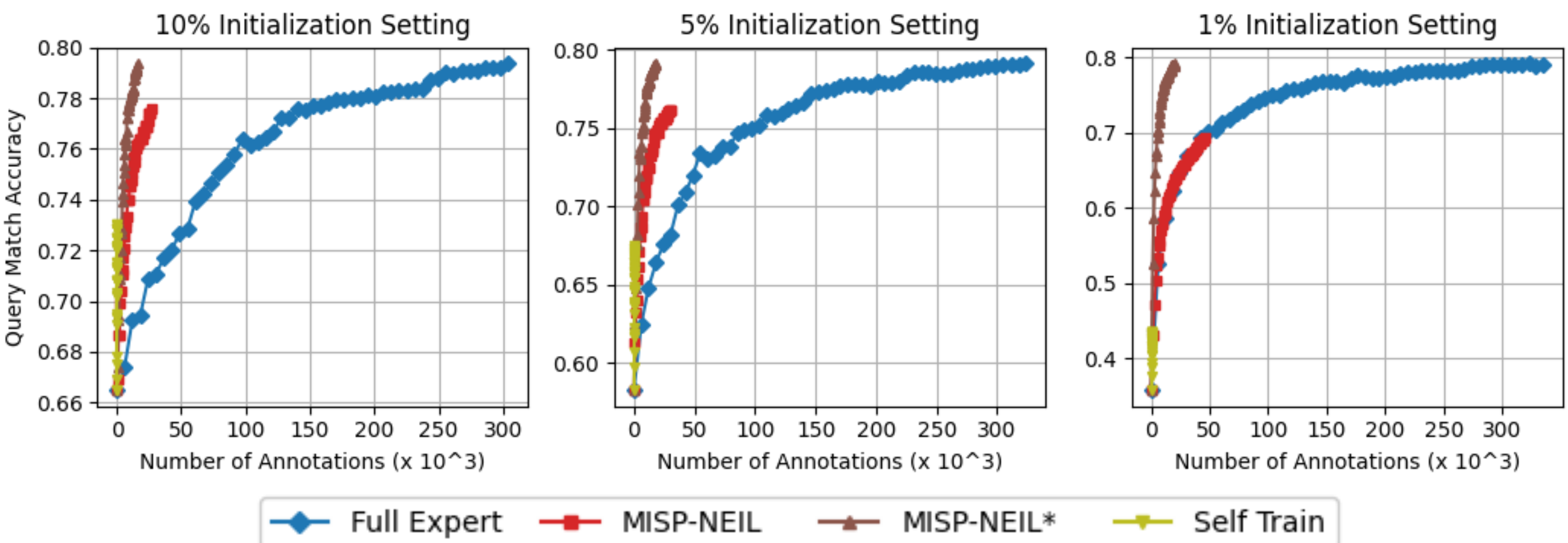
- Introduction
- MISP-NEIL architecture
  - Interactive semantic parsing with **MISP**
  - **NEIL**: aNnotation-Efficient Imitation Learning (with theoretical analysis)
- Experiments
- Future work

Experimental Setup

- Benchmark dataset: WikiSQL [Zhong et al., 2017]
- Base semantic parser: SQLova [Hwang et al., 2019]
- Three parser initialization settings
  - using 10% (around 5K), 5% and 1% (around 500) of the training data
- Iterative parser learning
  - In each iteration, simulate 1K (unlabeled) user questions
  - Simulated user interaction/feedback

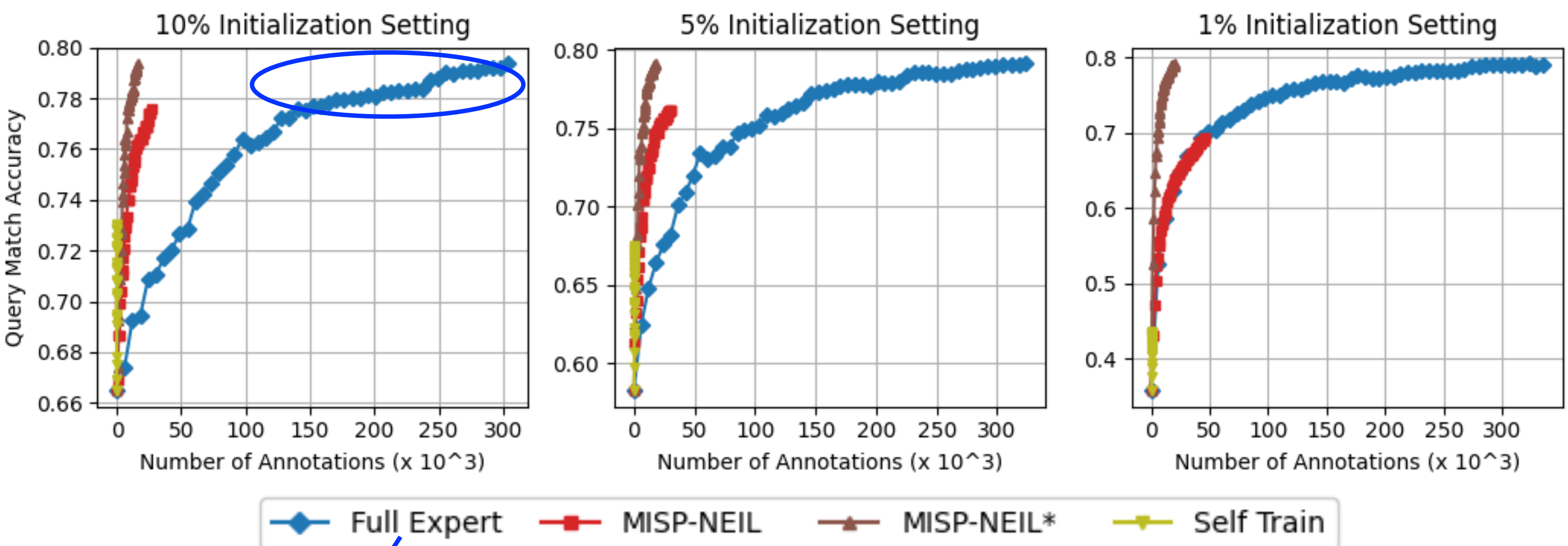
Comparison on Annotation Efficiency

- Parser's test-time accuracy when each system has consumed a certain number of annotations in training



Comparison on Annotation Efficiency

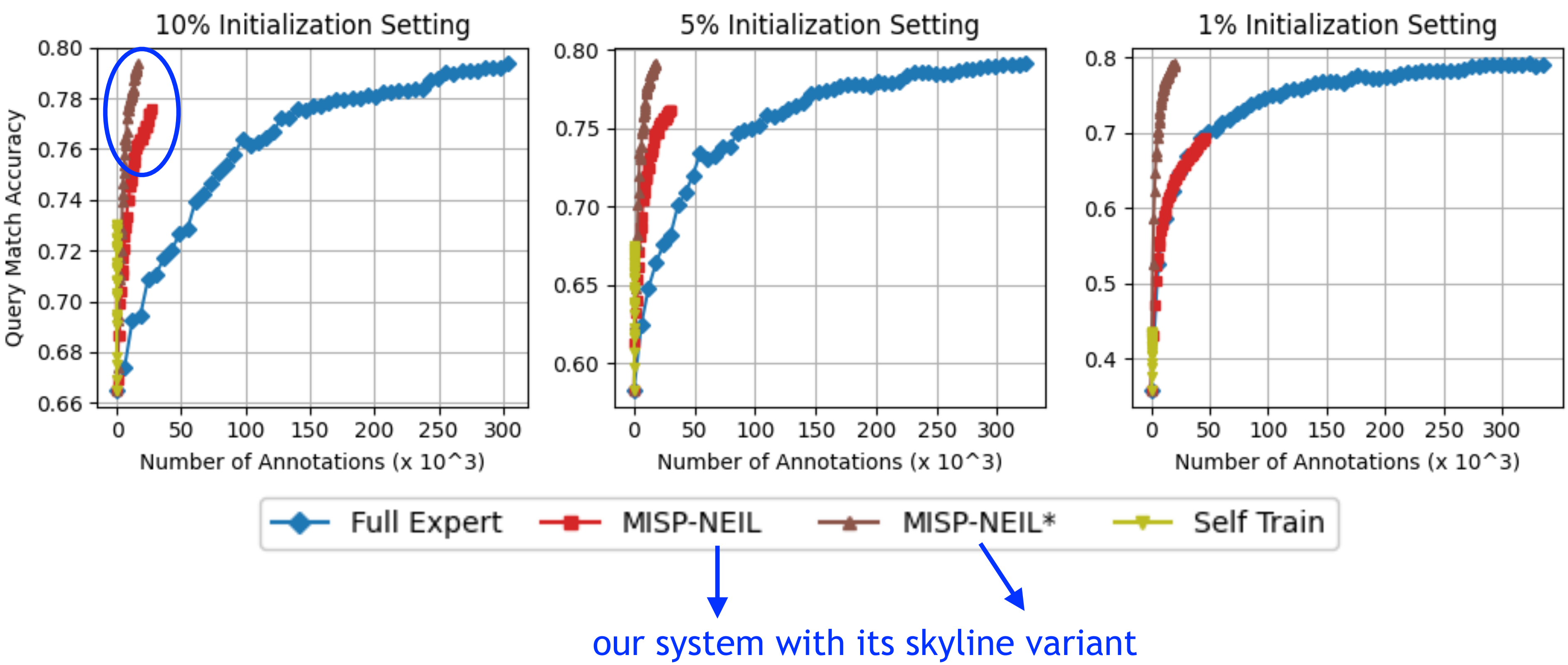
- Parser's test-time accuracy when each system has consumed a certain number of annotations in training



fully-supervised approach

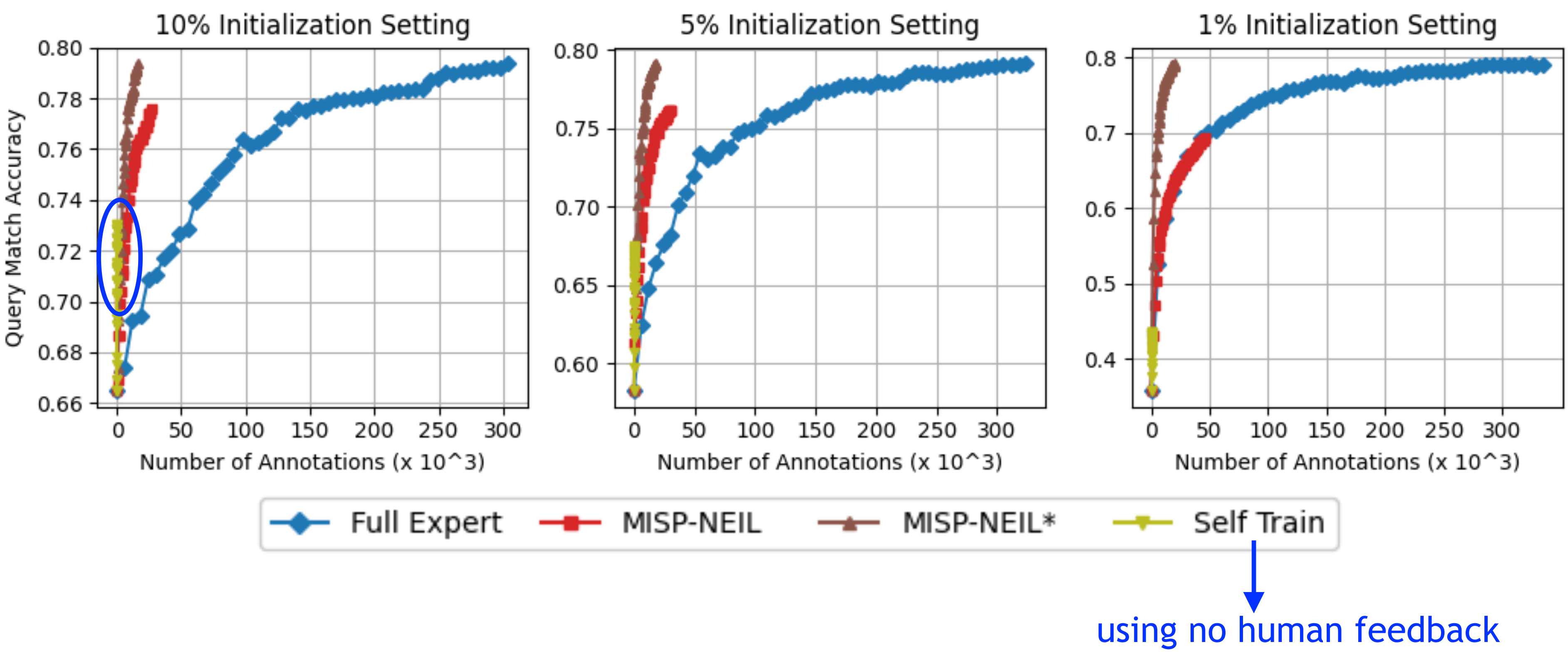
Comparison on Annotation Efficiency

- Parser's test-time accuracy when each system has consumed a certain number of annotations in training



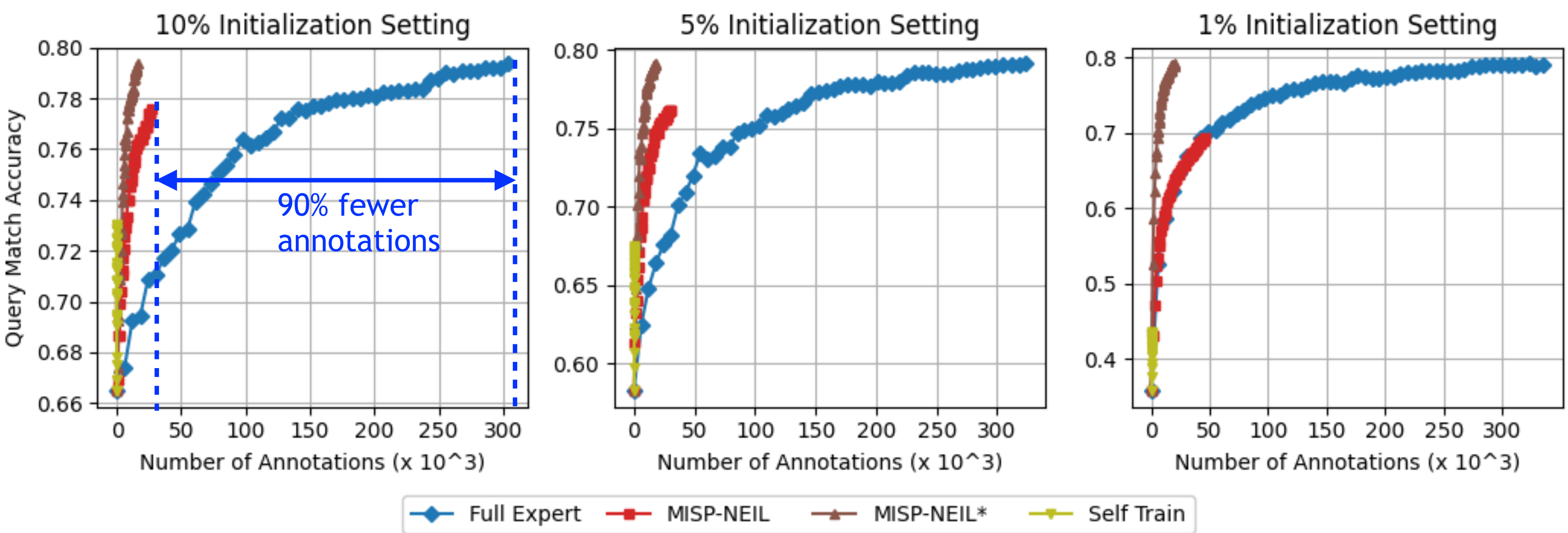
Comparison on Annotation Efficiency

- Parser's test-time accuracy when each system has consumed a certain number of annotations in training



Comparison on Annotation Efficiency

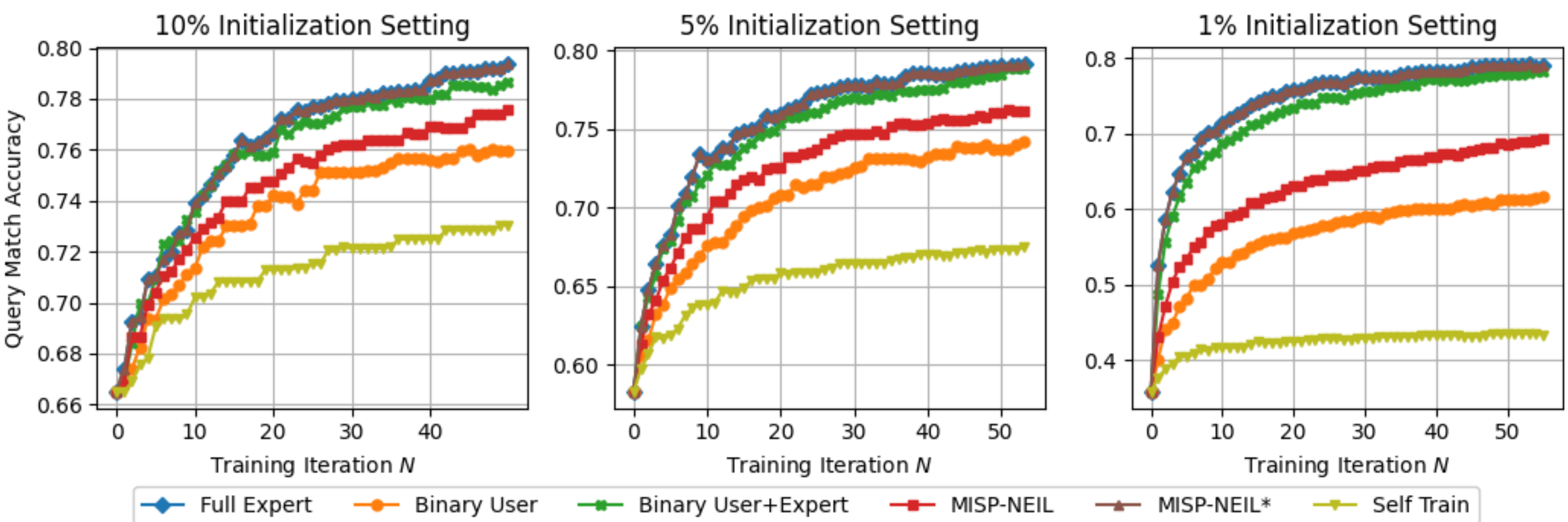
- Parser's test-time accuracy when each system has consumed a certain number of annotations in training



Observation: MISP-NEIL enjoys the best annotation efficiency (PLUS collecting annotations from *users* rather than *experts*)

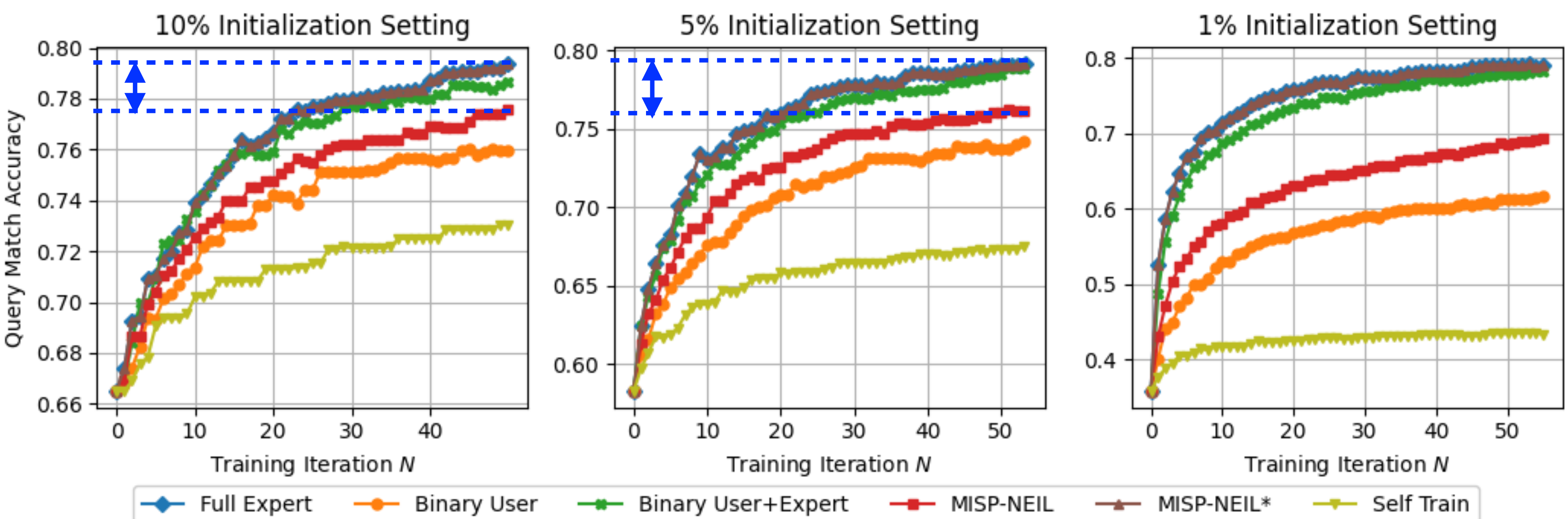
Comparison on Training Effectiveness

- Parser's test-time accuracy when each system has trained the parser for the same number of iterations



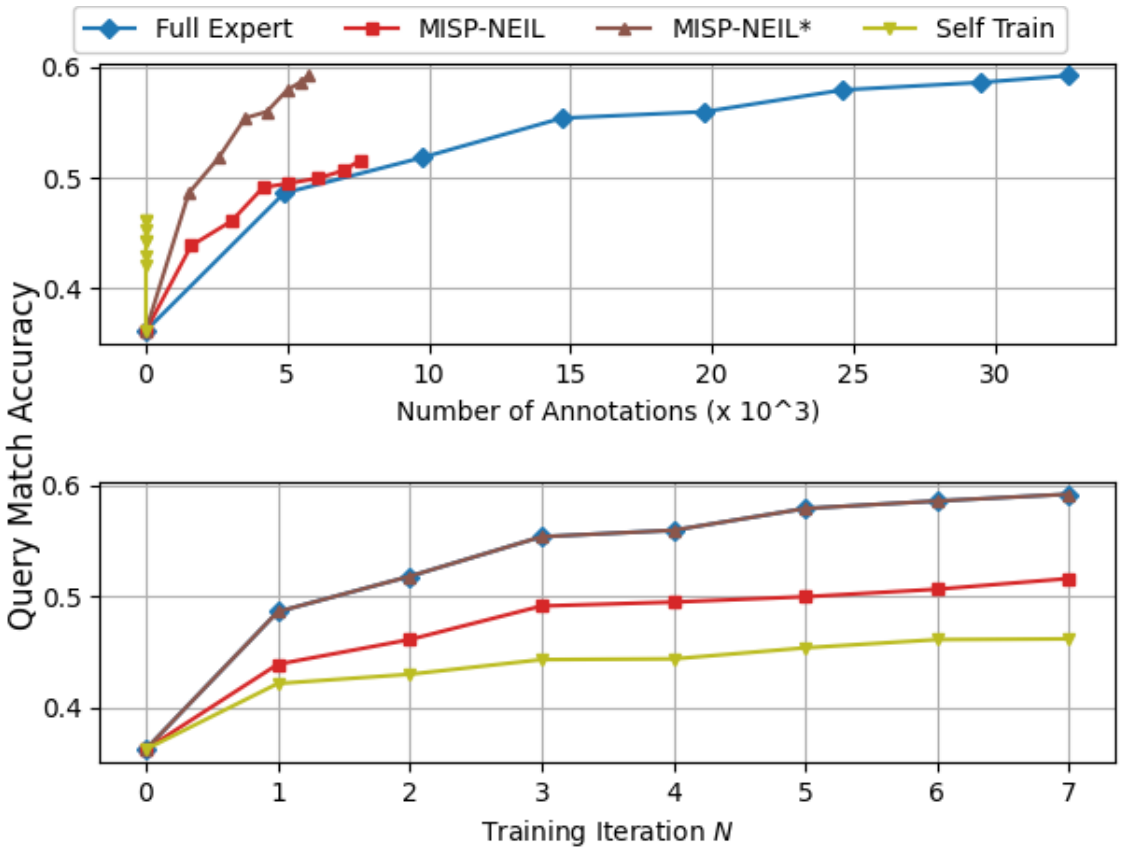
Comparison on Training Effectiveness

- Parser's test-time accuracy when each system has trained the parser for the same number of iterations



(1) When the parser is **moderately initialized** (10%/5% setting), MISP-NEIL is comparable with Full Expert (only 2% Acc loss) while being annotation-efficient;
(2) MISP-NEIL also **outperforms** other learning-from-user systems.

Experimental Results on Spider



Please check out our paper for more details

Outline

- Introduction
- MISP-NEIL architecture
  - Interactive semantic parsing with **MISP**
  - **NEIL**: aNnotation-Efficient Imitation Learning (with theoretical analysis)
- Experiments
- Future work

Future Work

- Large-scale user study
  - MISP is shown helpful for end-users in a small user test [Yao et al., 2019]
  - We aim at a more realistic test with *crowd workers*

- More accurate uncertainty estimation
  - Neural semantic parsers tend to be overconfident
  - Possible solutions: neural network calibration [Guo et al., 2017], using machine learning modules [Zhao et al., 2017; Fang et al., 2017]

- NEIL for saving annotations for low-resource tasks

Acknowledgement

Code is available at: https://github.com/sunlab-osu/MISP

Thank you!