Reasoning about Intent for Ambiguous Requests

Irina Saparina and **Mirella Lapata**
School of Informatics
University of Edinburgh



Irina Saparina



This is Irina's work; I was her thesis advisor.

She could not travel to the conference,
so I am presenting it on her behalf.

Ambiguity in Text-to-SQL Semantic Parsing

A question can map to **multiple valid answers** depending on interpretation.

restaurants				
id	**city**	**address**	opened	franchise
1	Edinburgh	Princes St	2011	1
2	London	Oxford St	2009	2

Where do desserts sell best?

Ambiguity in Text-to-SQL Semantic Parsing

A question can map to **multiple valid answers** depending on interpretation.

restaurants

id	city	address	opened	franchise
1	Edinburgh	Princes St	2011	1
2	London	Oxford St	2009	2

Where do desserts sell best?

```
SELECT address FROM restaurants ...
```
→ **Princes St**

```
SELECT city FROM restaurants ...
```
→ **Edinburgh**

```
SELECT city , address FROM restaurants ...
```
→ **Edinburgh, Princes St**

Ambiguity in Text-to-SQL Semantic Parsing

Models need to **recognise and handle ambiguity**, but they often **silently commit** to a single interpretation, overlooking alternatives.

Where do desserts sell best?

Ambiguity in Text-to-SQL Semantic Parsing

Models need to **recognise and handle ambiguity**, but they often **silently commit** to a single interpretation, overlooking alternatives.

Where do desserts sell best?

Reasoning trace

The user wants to know where desserts sell best... "Where" refers to a location, so I'll return the **city**...

Ambiguity in Text-to-SQL Semantic Parsing

Models need to **recognise and handle ambiguity**, but they often **silently commit** to a single interpretation, overlooking alternatives.

Where do desserts sell best?

Reasoning trace

The user wants to know where desserts sell best... "Where" refers to a location, so I'll return the **city**...

Answer

```
SELECT city FROM restaurants ...
```

Reasoning ignores intent: tokens go to the wrong interpretation, wasting compute and reinforcing the error.

Reasoning about Intent

We propose training models with **RL to reason about intent** before committing to an answer.

Where do desserts sell best?

Reasoning about Intent

We propose training models with **RL to reason about intent** before committing to an answer.

Reasoning trace

Where do desserts sell best?

The question . . . is ambiguous because “where” refers to a location, which could mean either “**address**” or “**city**”. . .

Reasoning about Intent

We propose training models with **RL to reason about intent** before committing to an answer.

Where do desserts sell best?

Reasoning trace

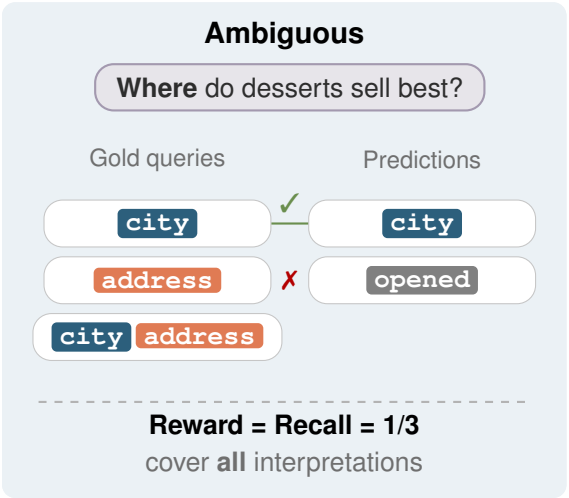
The question ... is ambiguous because "where" refers to a location, which could mean either "**address**" or "**city**"...

Interpretation-answer pairs

1. **Address** of the restaurant ...
   `SELECT address FROM restaurants...`
2. **City** is the restaurant ...
   `SELECT city FROM restaurants...`
3. **City** and **address** is the restaurant ...
   `SELECT city , address FROM restaurants...`

All interpretations and answers in a single response!

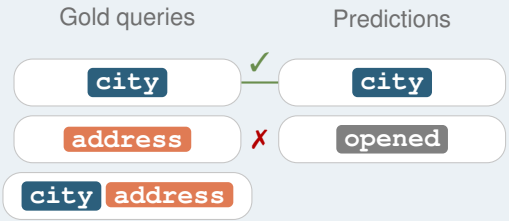
Dual Reward Function



Dual Reward Function

Ambiguous

Where do desserts sell best?

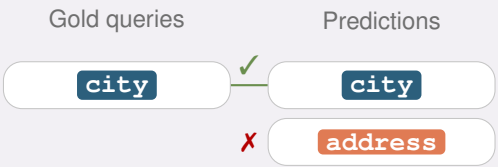


Reward = Recall = 1/3

cover **all** interpretations

Unambiguous

In which **city** do desserts sell best?



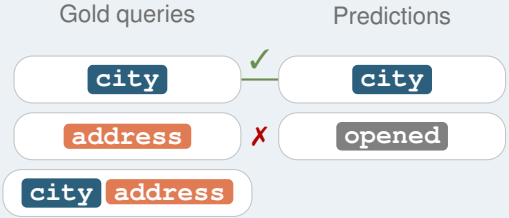
Reward = Precision = 1/2

don't invent interpretations

Dual Reward Function

Ambiguous

Where do desserts sell best?

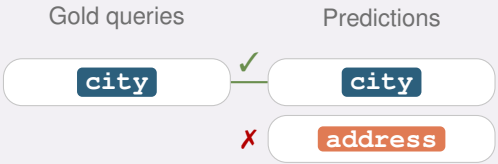


Reward = Recall = 1/3

cover **all** interpretations

Unambiguous

In which **city** do desserts sell best?



Reward = Precision = 1/2

don't invent interpretations

Only multiple valid answers required for training, no interpretations or clarification questions!

Experimental Setup

- **RL objective:** DAPO (Yu et al., 2025)
  oversample ambiguous examples to handle the difficulty bias
- **Model:** Qwen3 4B Instruct (Yang et al., 2025)

Text-to-SQL parsing

Ambrosia Saparina & Lapata (2024)	train + in-domain test
AmbiQT Bhaskar et al. (2023)	out-of-domain test

Conversational QA

Abg-CoQA Guo et al. (2021)	train + test

Model Comparisons

- **CoT:** chain-of-thought prompting
  Qwen3 4B and 235B-A22B, Instruct & Thinking
- **SFT** with gold interpretations
  Qwen3 4B Instruct
- **IntentRL:** our RL training
  Qwen3 4B Instruct

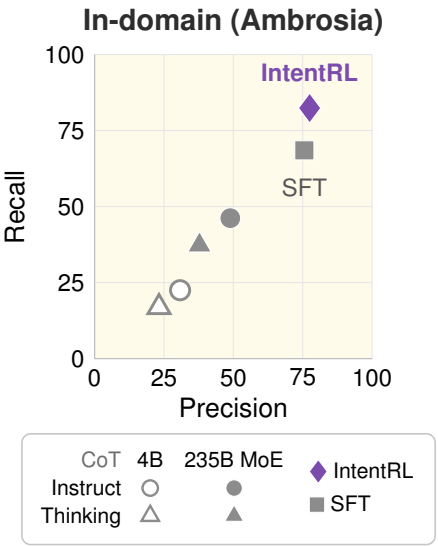
Prompt template

Analyse the question to **identify different possible** interpretations, then provide the corresponding answer for each interpretation . . .

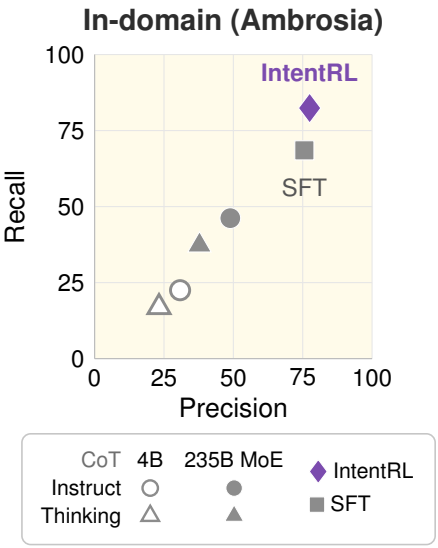
Context:
`[Context]`

Question to interpret and answer:
`[Question]`

Results: Precision vs. Recall (All Datasets)

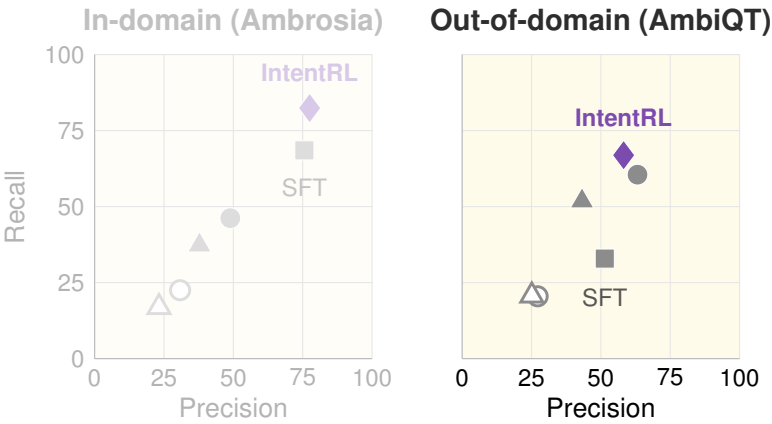


Results: Precision vs. Recall (All Datasets)



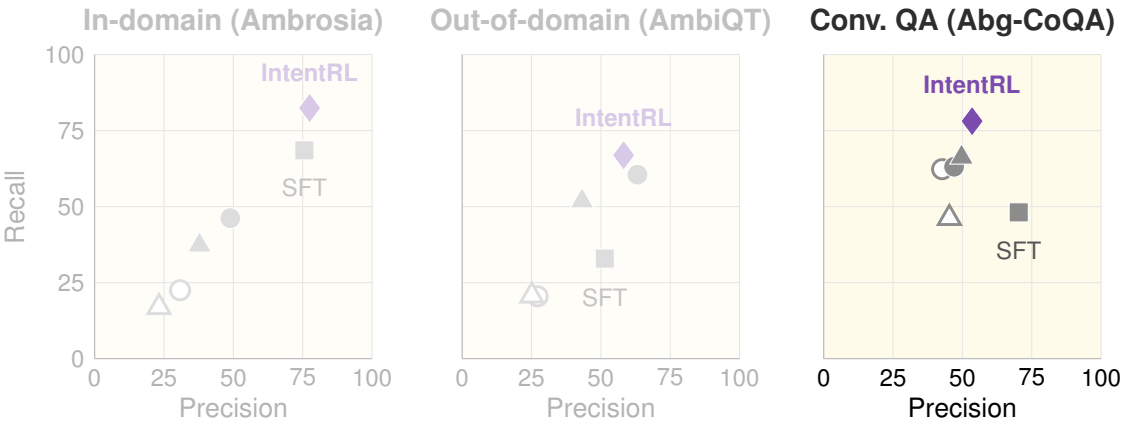
In-domain: IntentRL is best on both recall and precision

Results: Precision vs. Recall (All Datasets)



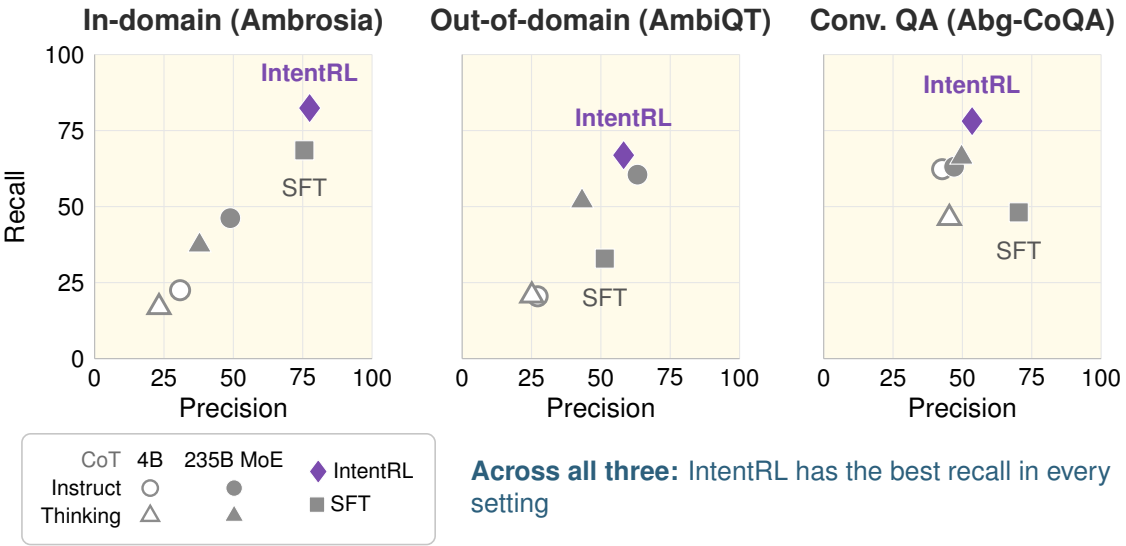
Out-of-domain: IntentRL keeps the best recall; SFT's recall drops to 32.9

Results: Precision vs. Recall (All Datasets)



Beyond SQL: IntentRL has the best recall; SFT is more precise but misses interpretations

Results: Precision vs. Recall (All Datasets)



Across all three: IntentRL has the best recall in every setting

Interpretation Alignment

- **Human evaluation** of IntentRL's interpretation-answer pairs
  30 ambiguous examples per dataset (92 pairs for Ambrosia, 80 for Abg-CoQA), 3 annotators
- **Task:** is the interpretation **compatible** with its answer?
  judged even when the answer is wrong

Ambrosia

91.7%

alignment

annotator agreement 84.0%

Abg-CoQA

90.0%

alignment

annotator agreement 81.0%

Over 90% of IntentRL's interpretations match their answers, with high annotator agreement

Key Takeaways

- **Reasoning about user intent** improves ambiguity resolution
- **IntentRL** produces **interpretation-answer pairs in a single response**
- Training requires **only multiple valid answers**
- **Best recall** on all datasets and **over 90% interpretation alignment**
- Generalises **out of domain** (AmbiQT) and **works beyond SQL** (conversational QA)

More in the paper: unambiguous questions, comparison with SOTA, transfer across models, clarification question generation

Code and models available

Poster: I-1-57