

Learning to Map Context- Dependent Sentences to Executable Formal Queries

Alane Suhr, Srinivasan Iyer, Yoav Artzi



Context-Dependent Language Understanding

- **Our goal:** language understanding in long interactions
- Prior work in semantic parsing, language to code focuses on sentences in isolation
- How can we make use of interaction history when interpreting later utterances?

Context-Dependent Language Understanding

User *Show me flights from Seattle to Boston next Monday*

Context-Dependent Language Understanding

User *Show me flights from Seattle to Boston next Monday*

Context-Dependent Language Understanding

User *Show me flights from Seattle to Boston next Monday*

**SQL
Query**

```
(SELECT DISTINCT flight.flight_id FROM flight WHERE  
(flight.from_airport IN (SELECT airport_service.airport_code  
FROM airport_service WHERE airport_service.city_code IN  
(SELECT city.city_code FROM city WHERE city.city_name =  
'SEATTLE')))) AND (flight.to_airport IN (SELECT  
airport_service.airport_code FROM airport_service WHERE  
airport_service.city_code IN (SELECT city.city_code FROM  
city WHERE city.city_name = 'BOSTON')))) AND  
(flight.flight_days IN (SELECT days.days_code FROM days  
WHERE days.day_name IN (SELECT date_day.day_name FROM  
date_day WHERE date_day.year = 1993 AND  
date_day.month_number = 2 AND date_day.day_number = 8))));
```

Context-Dependent Language Understanding

User *Show me flights from Seattle to Boston next Monday*

Result Found 31 Flights: 

Context-Dependent Language Understanding

User *On American Airlines*


Result Found 2764 Flights: 

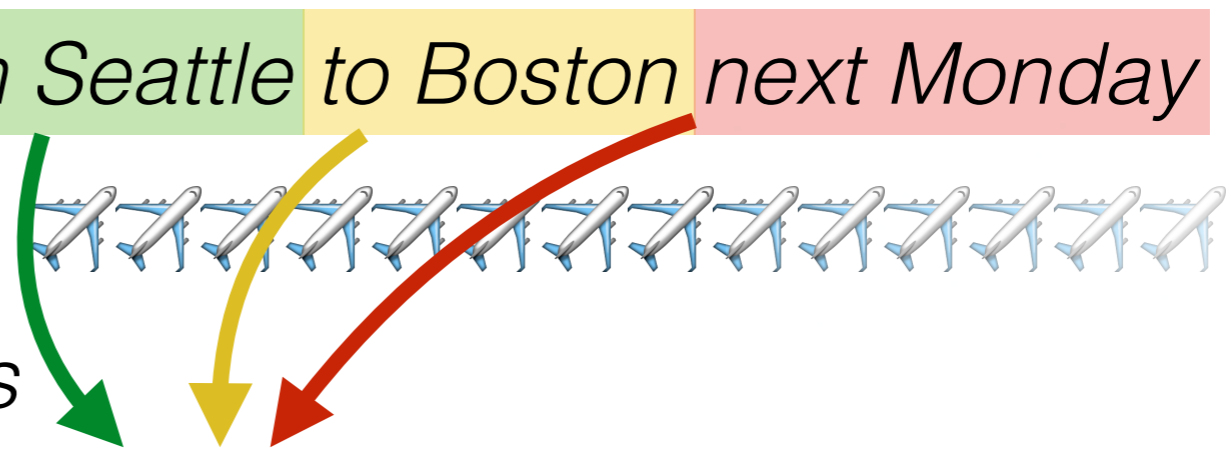
Context-Dependent Language Understanding

User Show me flights *from Seattle to Boston next Monday*

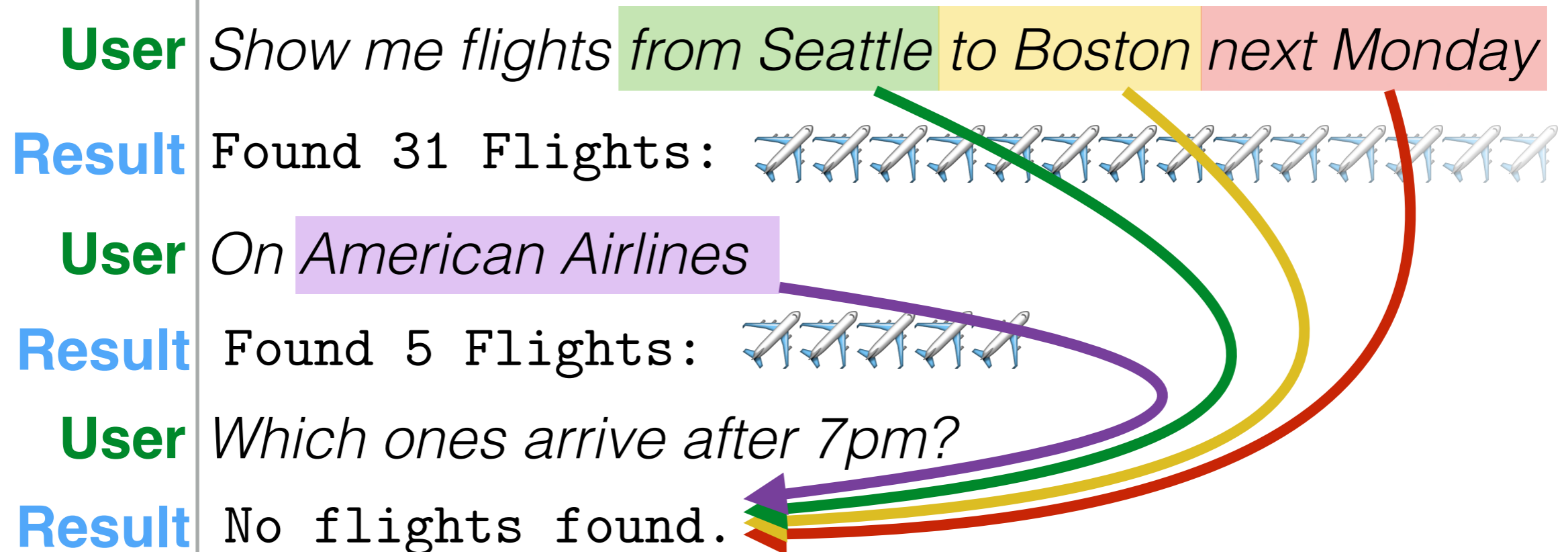
Result Found 31 Flights: 

User *On American Airlines*

Result Found 5 Flights: 



Context-Dependent Language Understanding




Context-Dependent Language Understanding

User Show me flights *from Seattle* *to Boston* *next Monday*

Result Found 31 Flights: 

User *On American Airlines*

Result Found 5 Flights: 

User *Which ones arrive after 7pm?*

Result No flights found.

User *Show me Delta flights*

Result Found 5 Flights: 



Context-Independent Prior Work

- Semantic parsing

Zelle and Money 1996, Clarke et al. 2010, Zettlemoyer and Collins 2005, Zettlemoyer and Collins 2007, Kwiatkowski et al. 2011, Artzi and Zettlemoyer 2011, Kushman and Barzilay 2013, Liang et al. 2011, Berant et al. 2013, Wang et al. 2014, Dong and Lapata 2016, Jia and Liang 2016

- Language to code

Popescu et al. 2004, Giordani and Moschitti 2012, Poon 2013, Ling et al. 2016, Zhong et al. 2017, Xu et al. 2017, Yin and Neubig 2017, Rabinovich et al. 2017, Krishnamurthy et al. 2017, Chen et al. 2017, Iyer et al. 2017

- Our approach: language understanding in interaction context

Context-Dependent Prior Work

SCONE (Long et al. 2016): micro domains focused on specific interaction phenomena

Guu et al. 2017, Fried et al. 2018, Suhr et al. 2018

ATIS (Hemphill et al. 1990, Dahl et al. 1994):

Miller et al. 1996, Zettlemoyer and Collins 2009

Use different representations; extra training & annotation.

- Our approach: single end-to-end model using only interaction data

Interaction History

- As an interaction progresses, the meaning of an utterance becomes highly dependent on the history of the interaction
- History includes both previous requests and generated SQL queries
- Two mechanisms

Incorporating Previous Requests

User *Show me all flights from Boston to Pittsburgh on Wednesday of next week which depart from Boston after 5pm*

The diagram shows a user's request: "Show me all flights from Boston to Pittsburgh on Wednesday of next week which depart from Boston after 5pm". The words "from Boston" are highlighted in light blue, "to Pittsburgh" in light green, "Wednesday of next week" in light yellow, and "after 5pm" in light orange. Four curved arrows originate from these highlights: a blue arrow from "from Boston" points to the second request; a green arrow from "to Pittsburgh" points to the second request; a yellow arrow from "Wednesday of next week" points to the third request; and an orange arrow from "after 5pm" points to the third request.

⋮ (3 turns)

User *Please describe the class of service Y*

⋮ (5 turns)

User *Show the cost of tickets on flight US 345*

Incorporating Previous Requests

- Relevant but elided information was mentioned many turns before
- User may change focus during interaction
- **Solution:** implicit mechanism for carrying information from beginning to end of interaction

Incorporating Previous Queries

User *Show me flights from Seattle to Boston next Monday*

SQL Query

```
(SELECT DISTINCT flight.flight_id FROM flight WHERE  
(flight.from_airport IN (SELECT airport_service.airport_code  
FROM airport_service WHERE airport_service.city_code IN  
(SELECT city.city_code FROM city WHERE city.city_name =  
'SEATTLE')))) AND (flight.to_airport IN (SELECT  
airport_service.airport_code FROM airport_service WHERE  
airport_service.city_code IN (SELECT city.city_code FROM  
city WHERE city.city_name = 'BOSTON')))) AND  
(flight.flight_days IN (SELECT days.days_code FROM days  
WHERE days.day_name IN (SELECT date_day.day_name FROM  
date_day WHERE date_day.year = 1993 AND  
date_day.month_number = 2 AND date_day.day_number = 8))));
```


Incorporating Previous Queries

User *On American Airlines*

**SQL
Query**

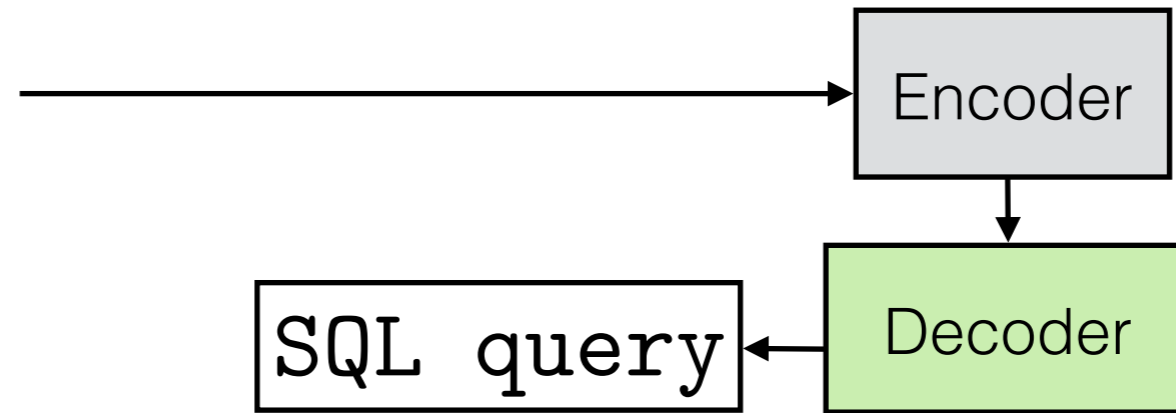
```
(SELECT DISTINCT flight.flight_id FROM flight WHERE  
(flight.airline_code = 'AA') AND (flight.from_airport IN  
(SELECT airport_service.airport_code FROM airport_service  
WHERE airport_service.city_code IN (SELECT city.city_code  
FROM city WHERE city.city_name = 'SEATTLE')))) AND  
(flight.to_airport IN (SELECT airport_service.airport_code  
FROM airport_service WHERE airport_service.city_code IN  
(SELECT city.city_code FROM city WHERE city.city_name =  
'BOSTON')))) AND (flight.flight_days IN (SELECT days.days_code  
FROM days WHERE days.day_name IN (SELECT date_day.day_name  
FROM date_day WHERE date_day.year = 1993 AND  
date_day.month_number = 2 AND date_day.day_number = 8))));
```

Incorporating Previous Queries

- Segments corresponding to earlier constraints appear in later queries
- **Solution:** explicit mechanism for composing later SQL queries from segments of previous ones

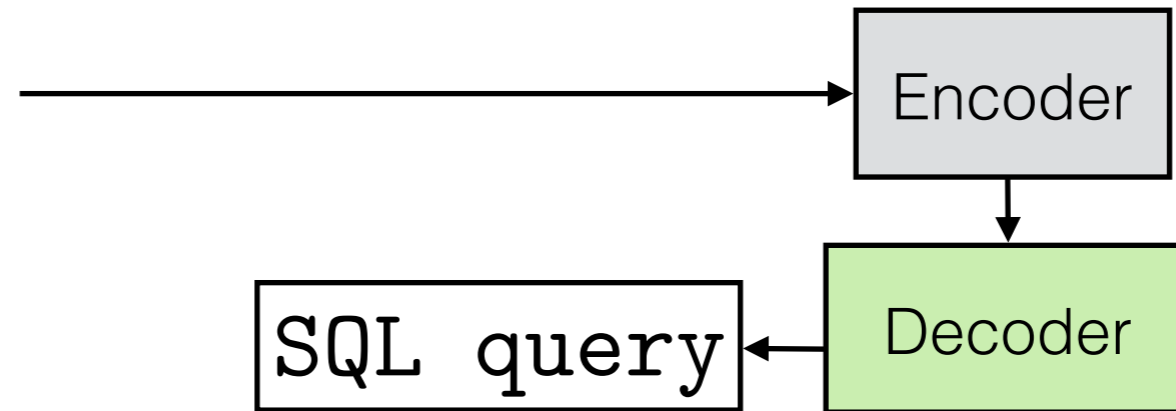
Model Overview

*Show me flights from
Seattle to Boston next
Monday*

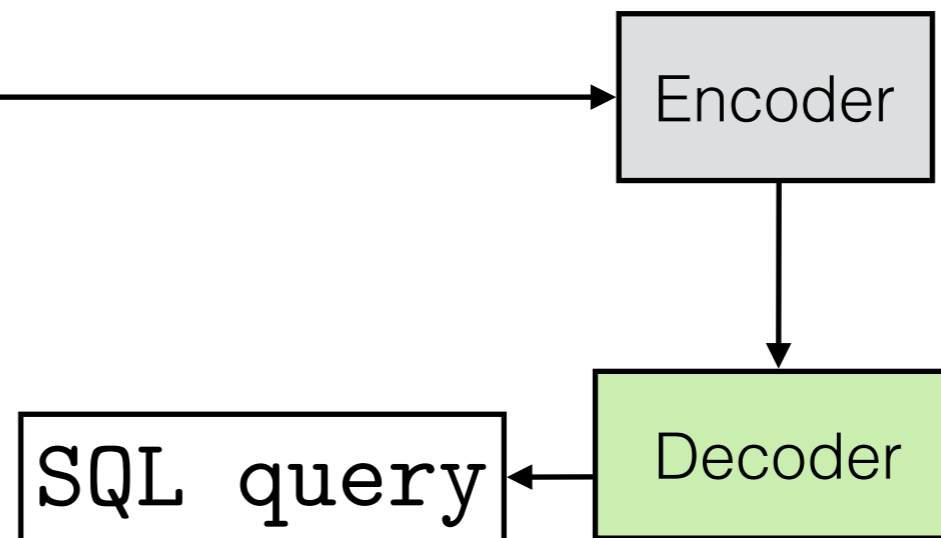


Model Overview

*Show me flights from
Seattle to Boston next
Monday*

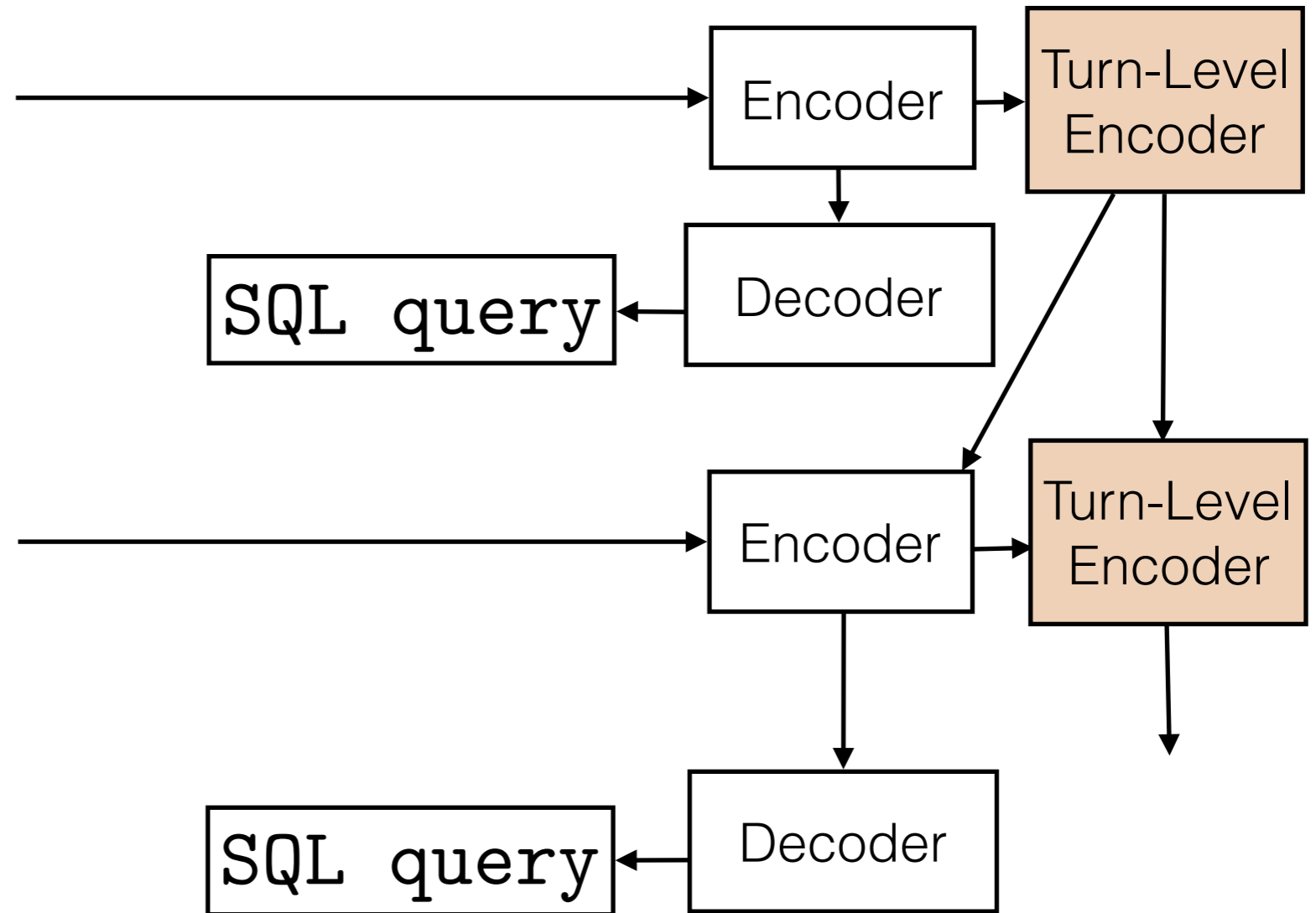


On American Airlines



Model Overview

*Show me flights from
Seattle to Boston next
Monday*

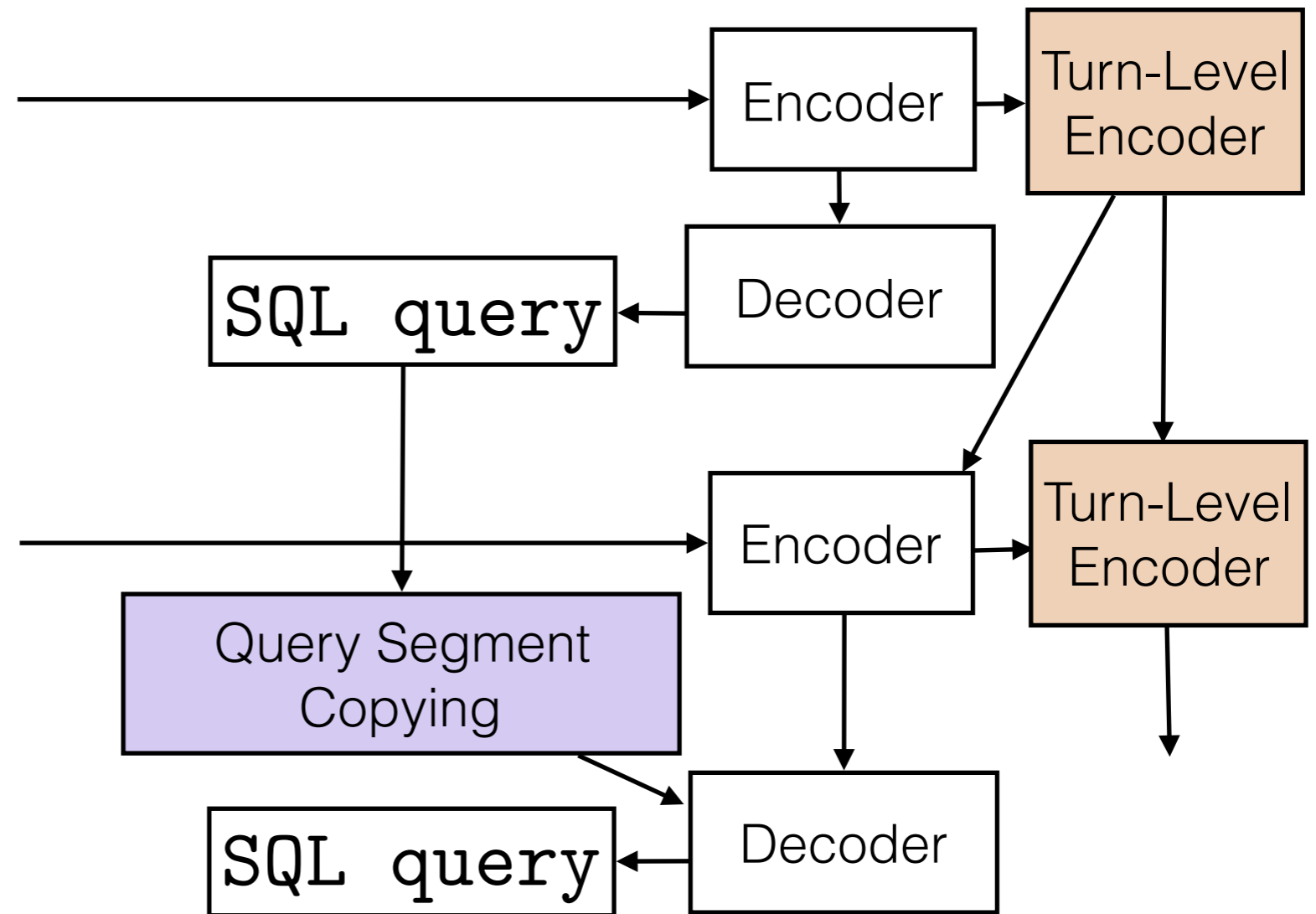


Mechanism 1 Previous Requests: Turn-level Encoder

Model Overview

*Show me flights from
Seattle to Boston next
Monday*

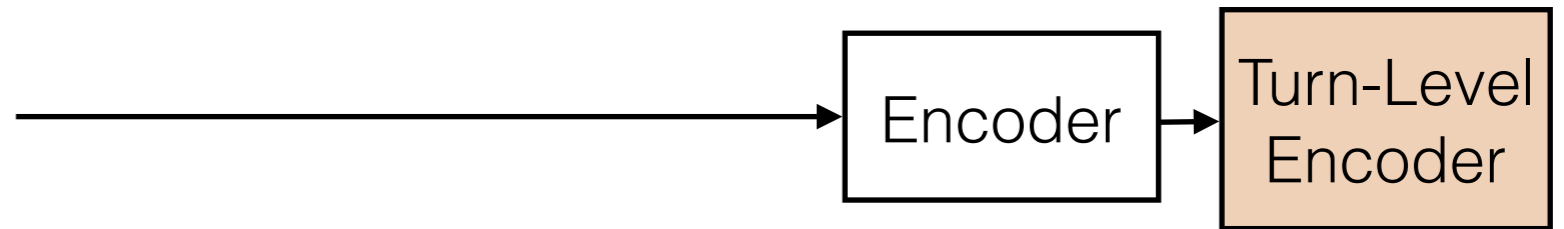
On American Airlines



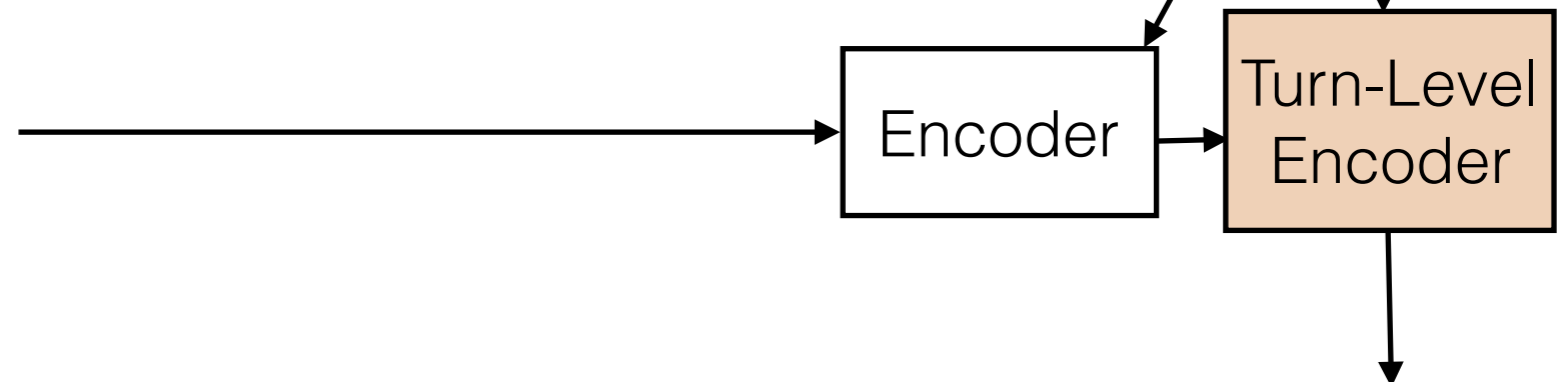
- Mechanism 1** Previous Requests: Turn-level Encoder
- Mechanism 2** Previous Queries: Query Segment Copying

Turn-level Encoder

*Show me flights from
Seattle to Boston next
Monday*



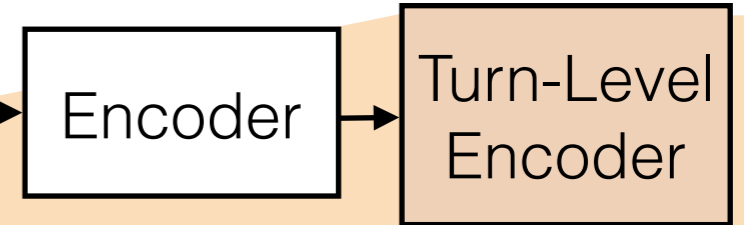
On American Airlines



Mechanism 1 Previous Requests: Turn-level Encoder

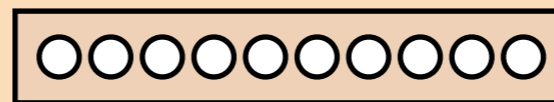
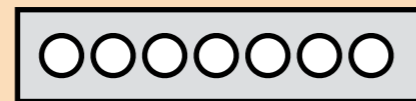
Turn-level Encoder

*Show me flights from
Seattle to Boston next
Monday*

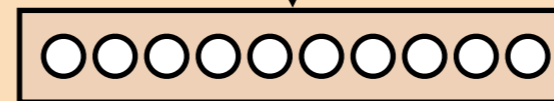


1. State Update

Encoded request



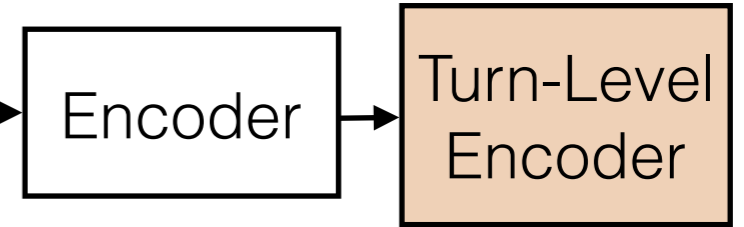
Discourse-level vector state



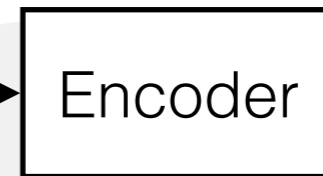
New
discourse-level vector state

Turn-level Encoder

*Show me flights from
Seattle to Boston next
Monday*

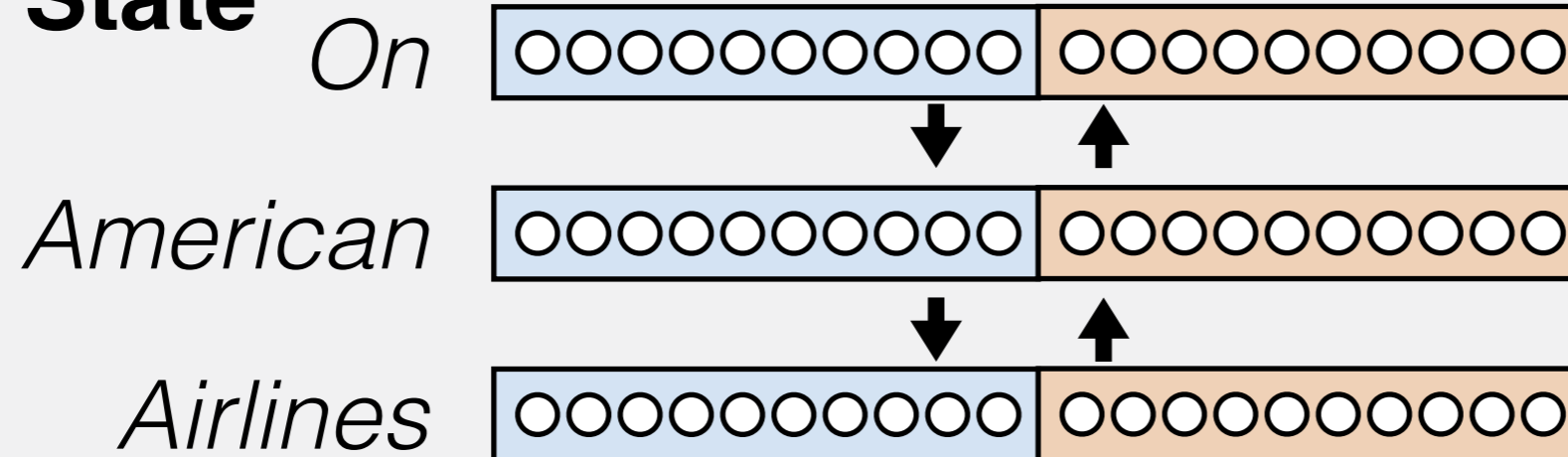


On American Airlines



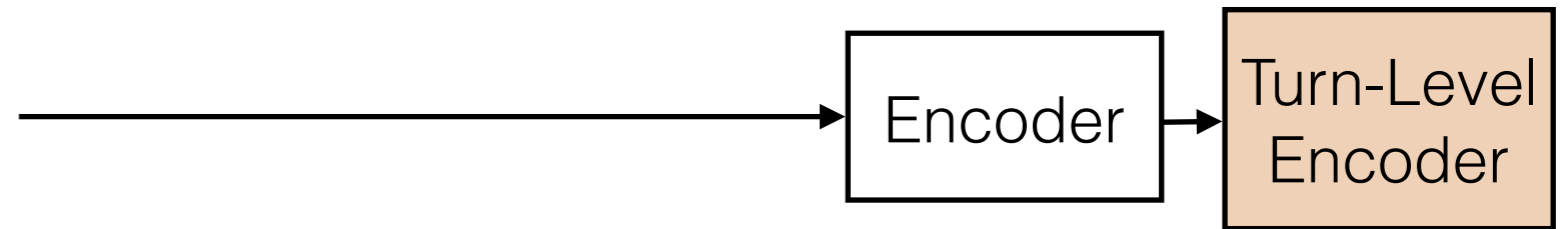
2. Using State

Word embeddings Discourse-level state

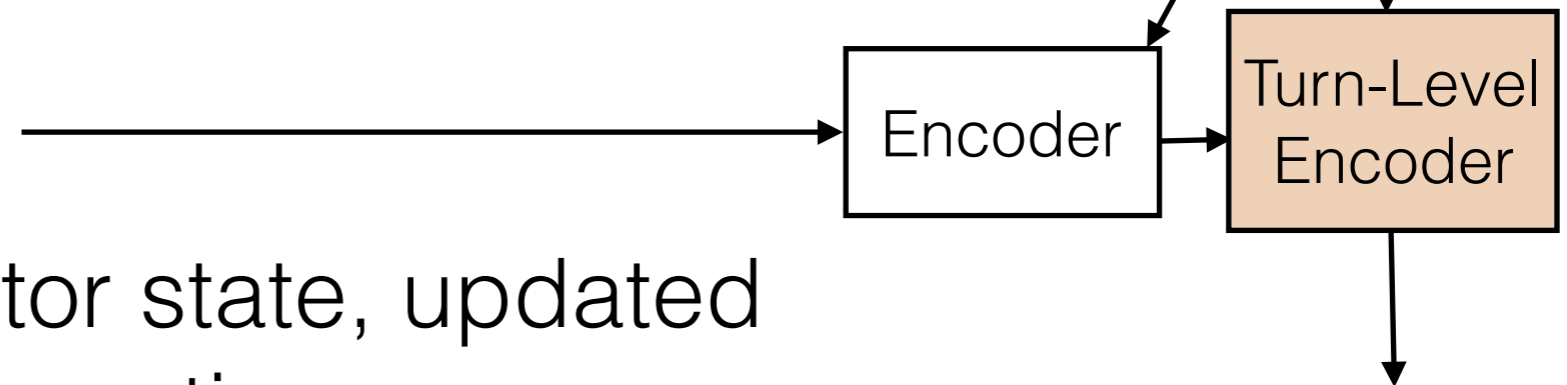


Turn-level Encoder

*Show me flights from
Seattle to Boston next
Monday*



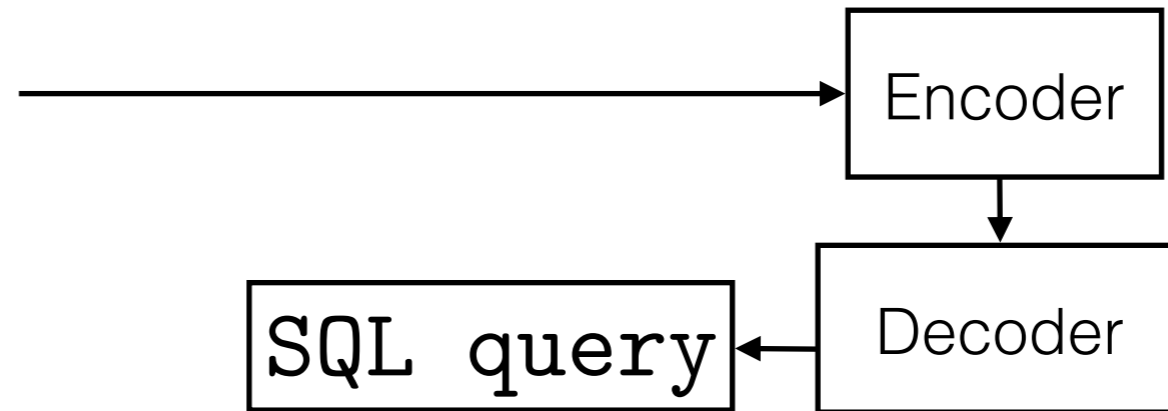
On American Airlines



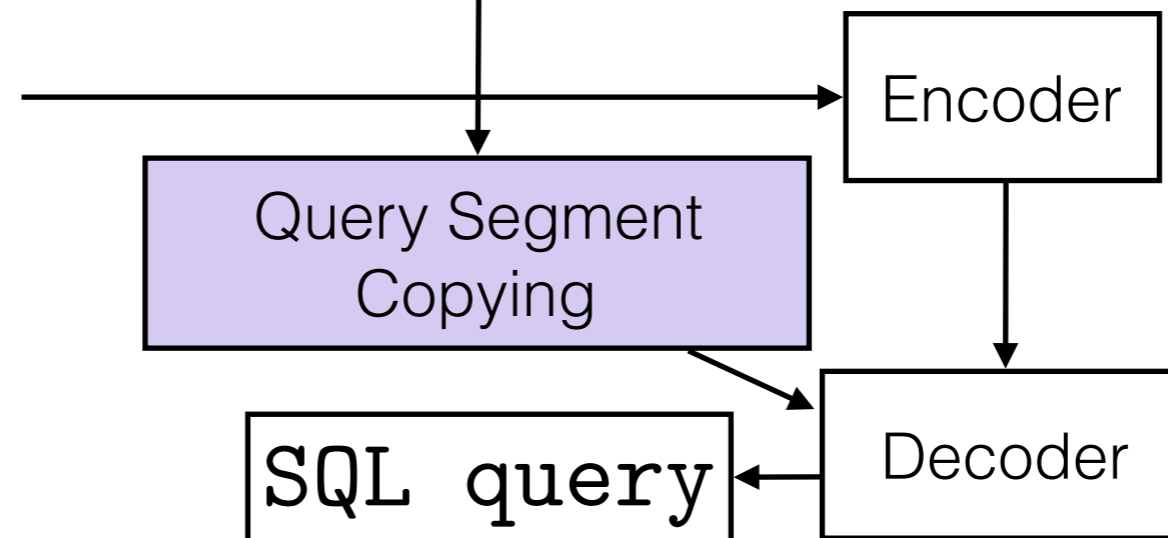
- Persistent vector state, updated throughout interaction
- Encode information from beginning to end of interaction
- Completely learned

Query Segment Copying

*Show me flights from
Seattle to Boston next
Monday*

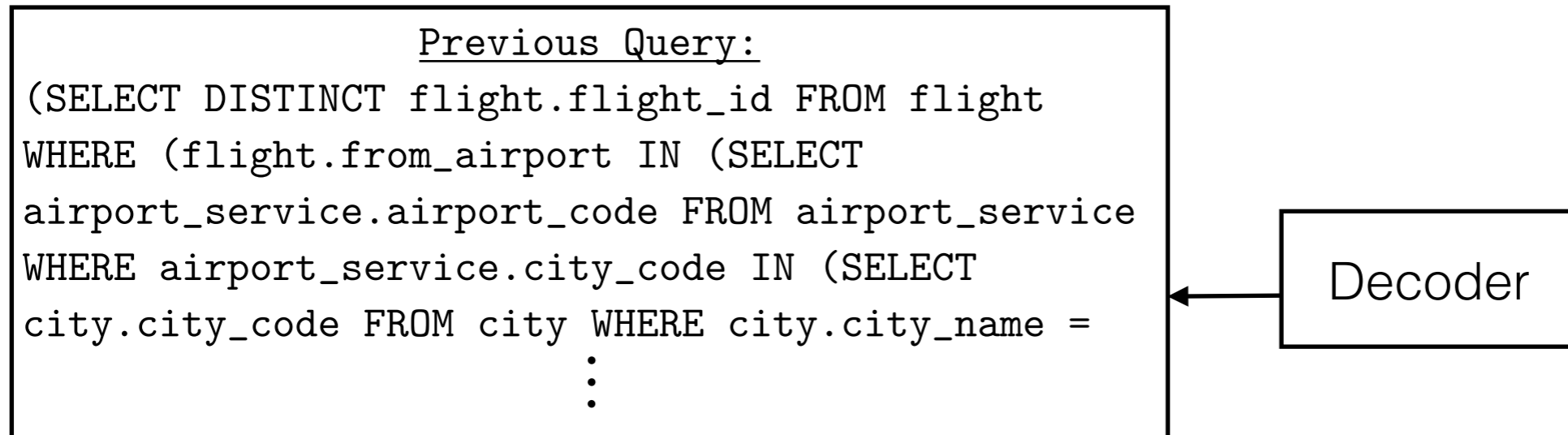


On American Airlines



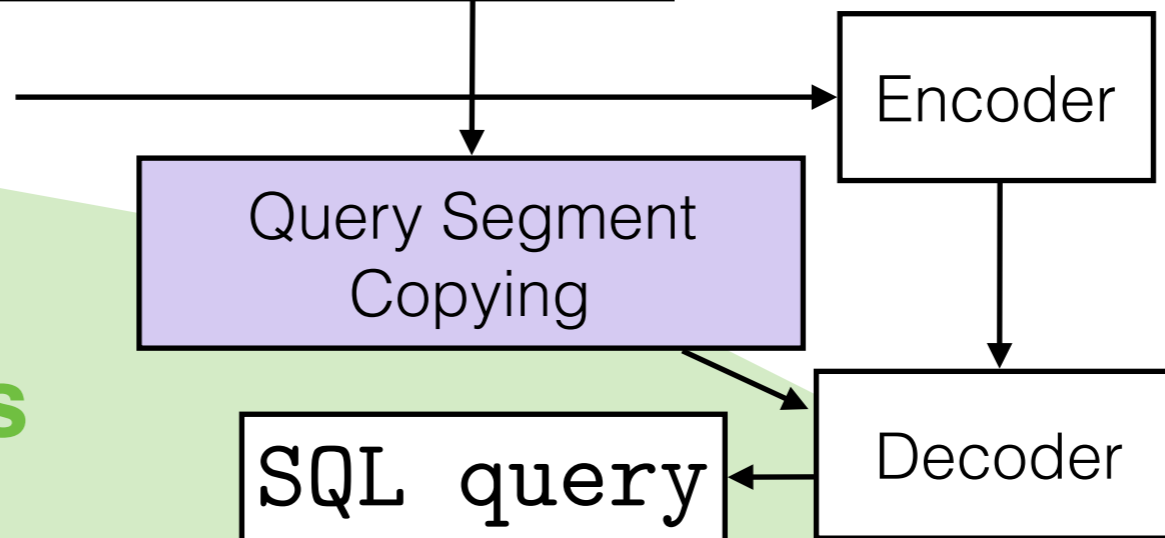
Mechanism 2 Previous Queries: Query Segment Copying

Query Segment Copying



On American Airlines

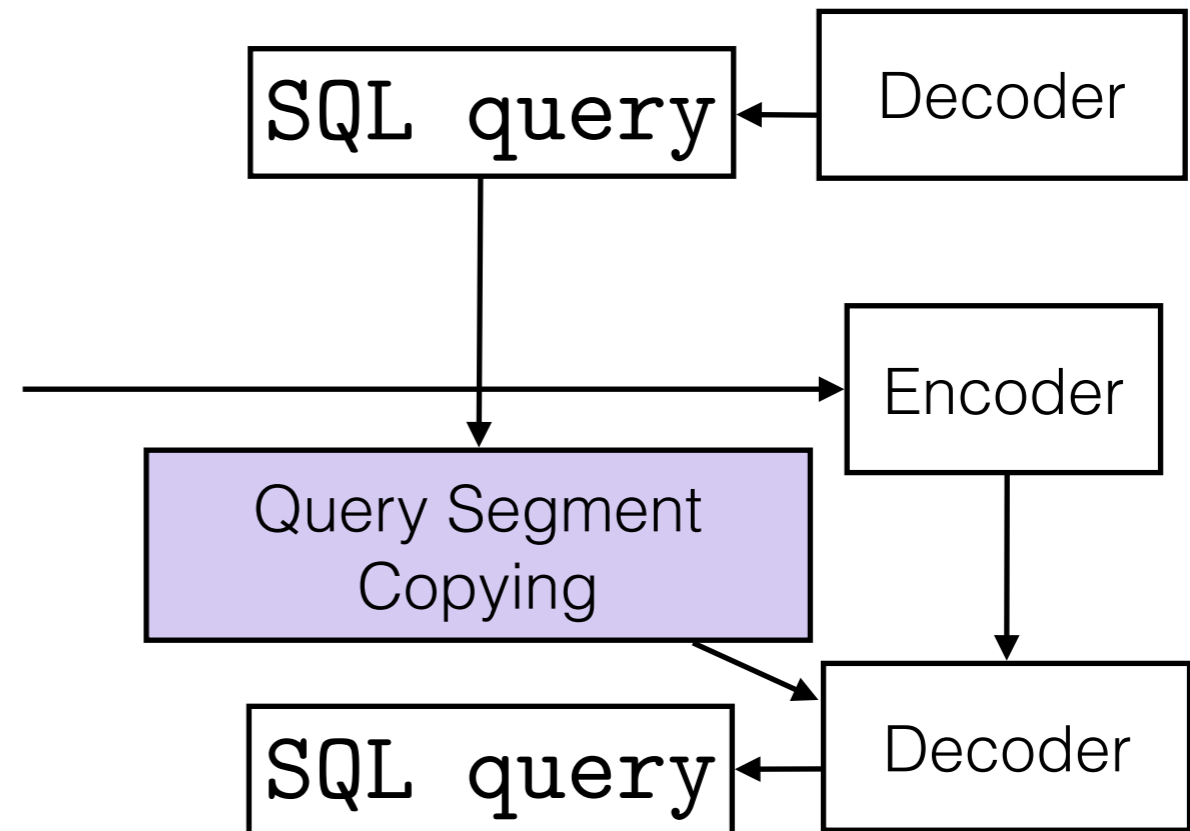
3. Generating Query Segments



Probability of query segment
computed using its vector state

Query Segment Copying

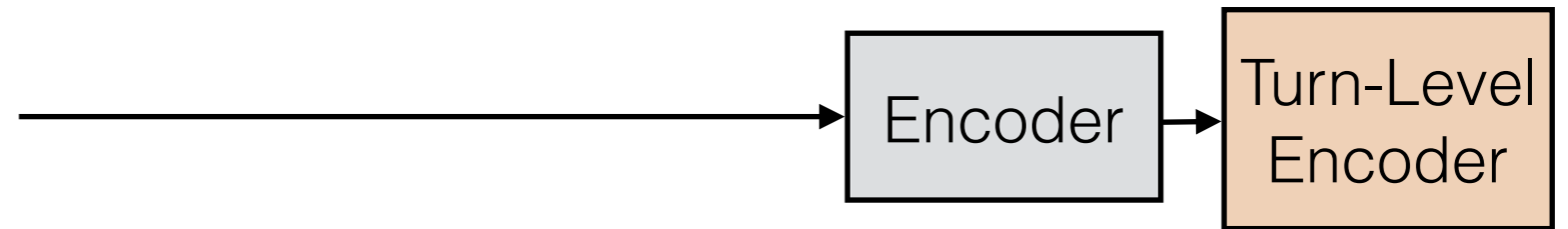
On American Airlines



- Explicit mechanism for copying previous constraints
- Encoding and generating segments learned with the rest of the model

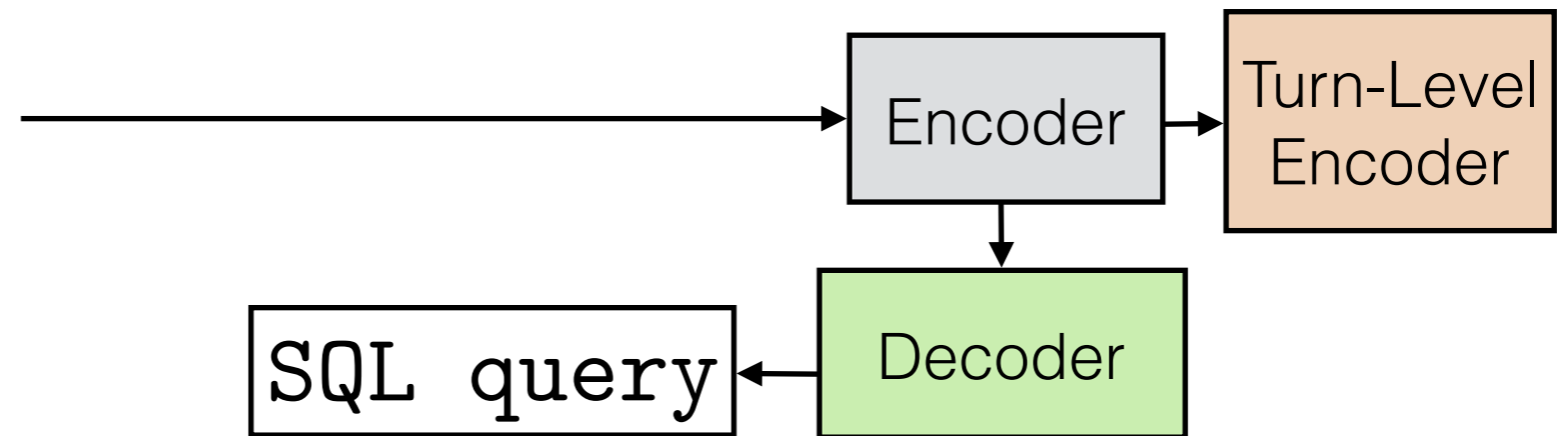
Inference

*Show me flights from
Seattle to Boston next
Monday*



Inference

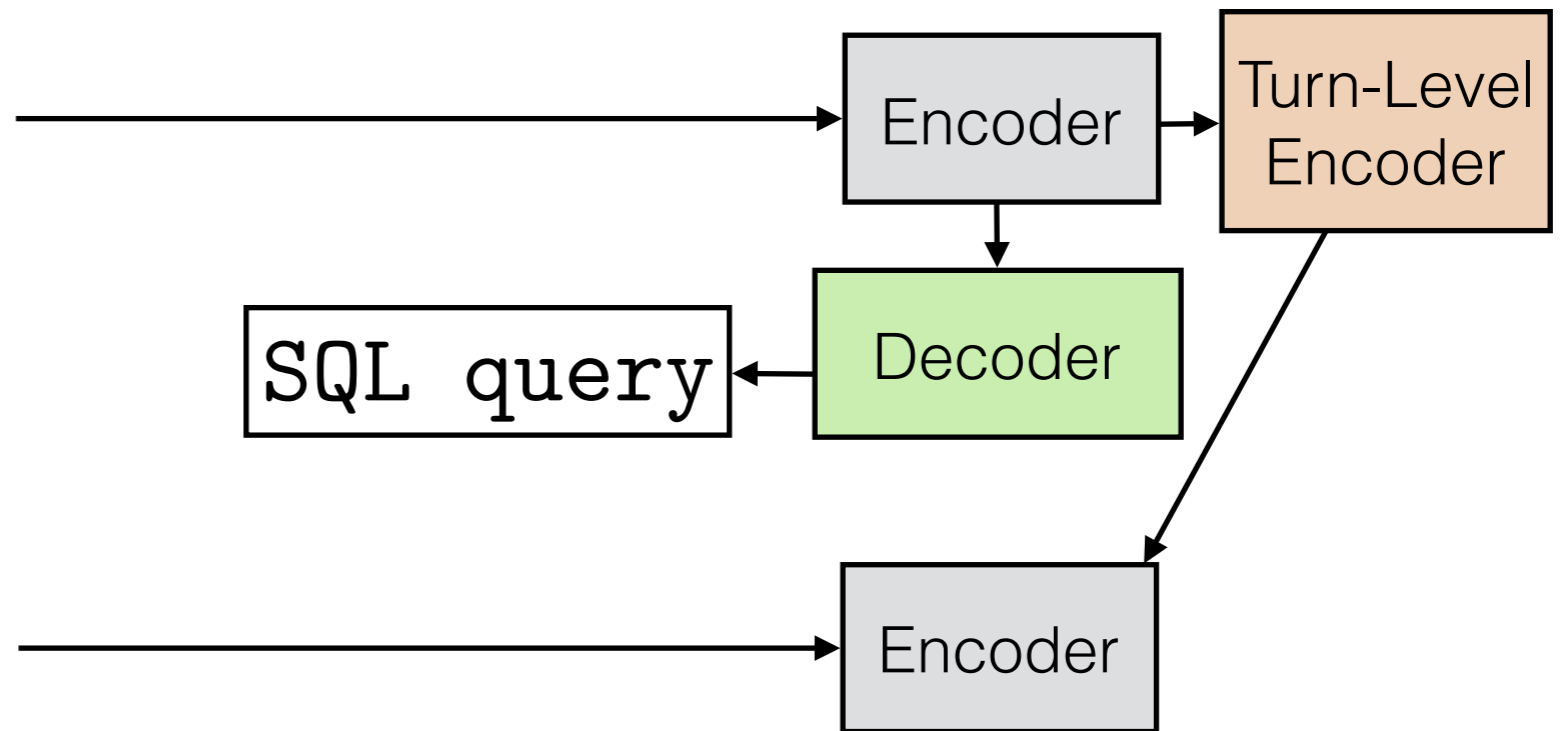
*Show me flights from
Seattle to Boston next
Monday*



Inference

*Show me flights from
Seattle to Boston next
Monday*

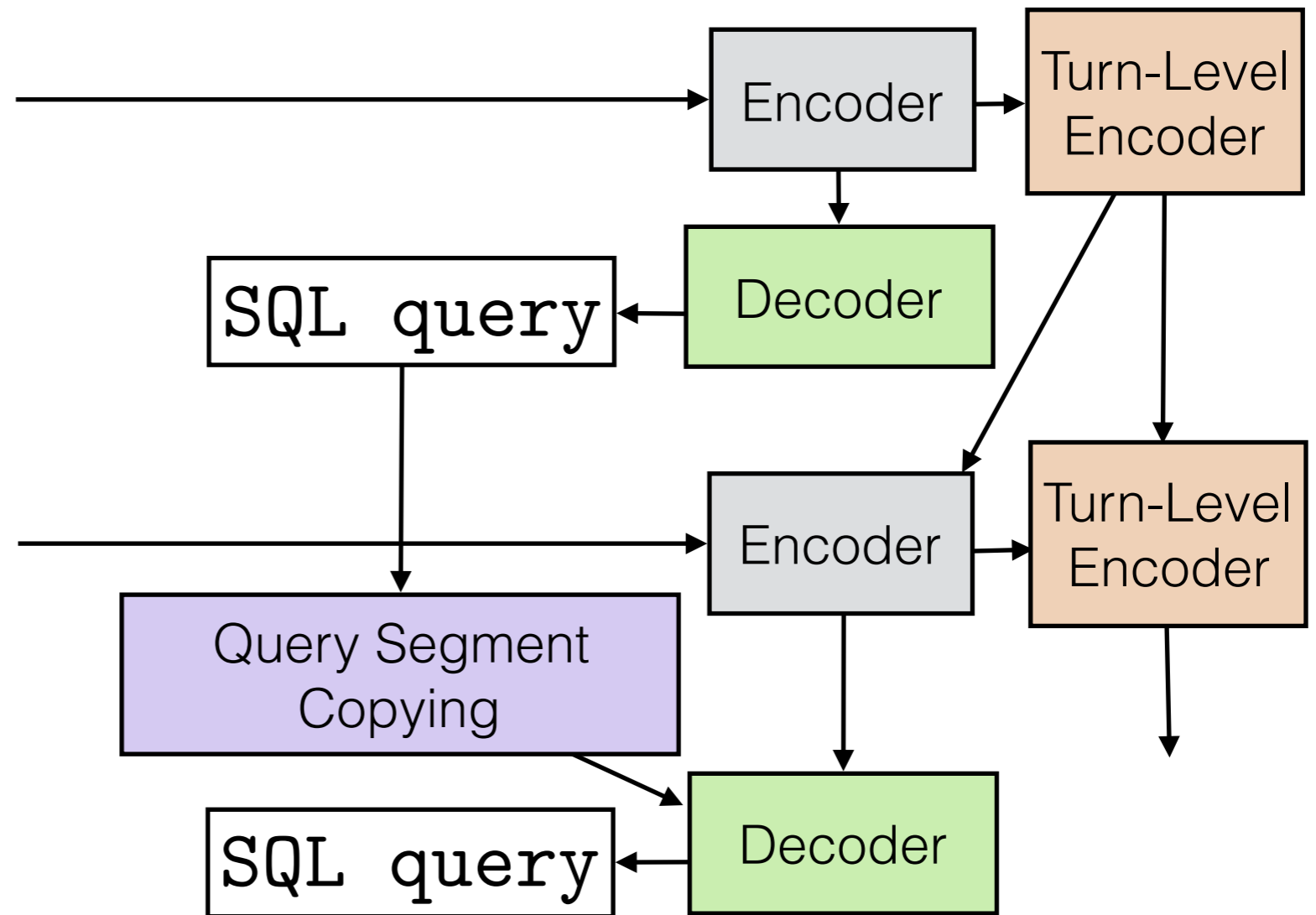
On American Airlines



Inference

*Show me flights from
Seattle to Boston next
Monday*

On American Airlines



Learning

- Training data: interactions with request-SQL pairs
- Objective: minimize token-level cross-entropy loss
- All learned components updated together

ATIS

(Hemphill et al. 1990, Dahl et al. 1994)

- Flight information, 27 tables, 162K entries
- Small corpus: <2000 interactions
- Long interactions: average 7 turns; maximum: 64
- Complex queries: average 102.9 tokens each; 93% reference >3 tables

Handling Entities

- Need to generalize to rare or unseen constraints
- Can take advantage of database context
- Apply entity identification, anonymization techniques

Handling Entities

User *Show me flights from Seattle to Boston next Monday*

User *Show me flights from CITY1 to CITY2 YEAR MONTH DAY*

SQL Query `... city.city_name = CITY1 ...`

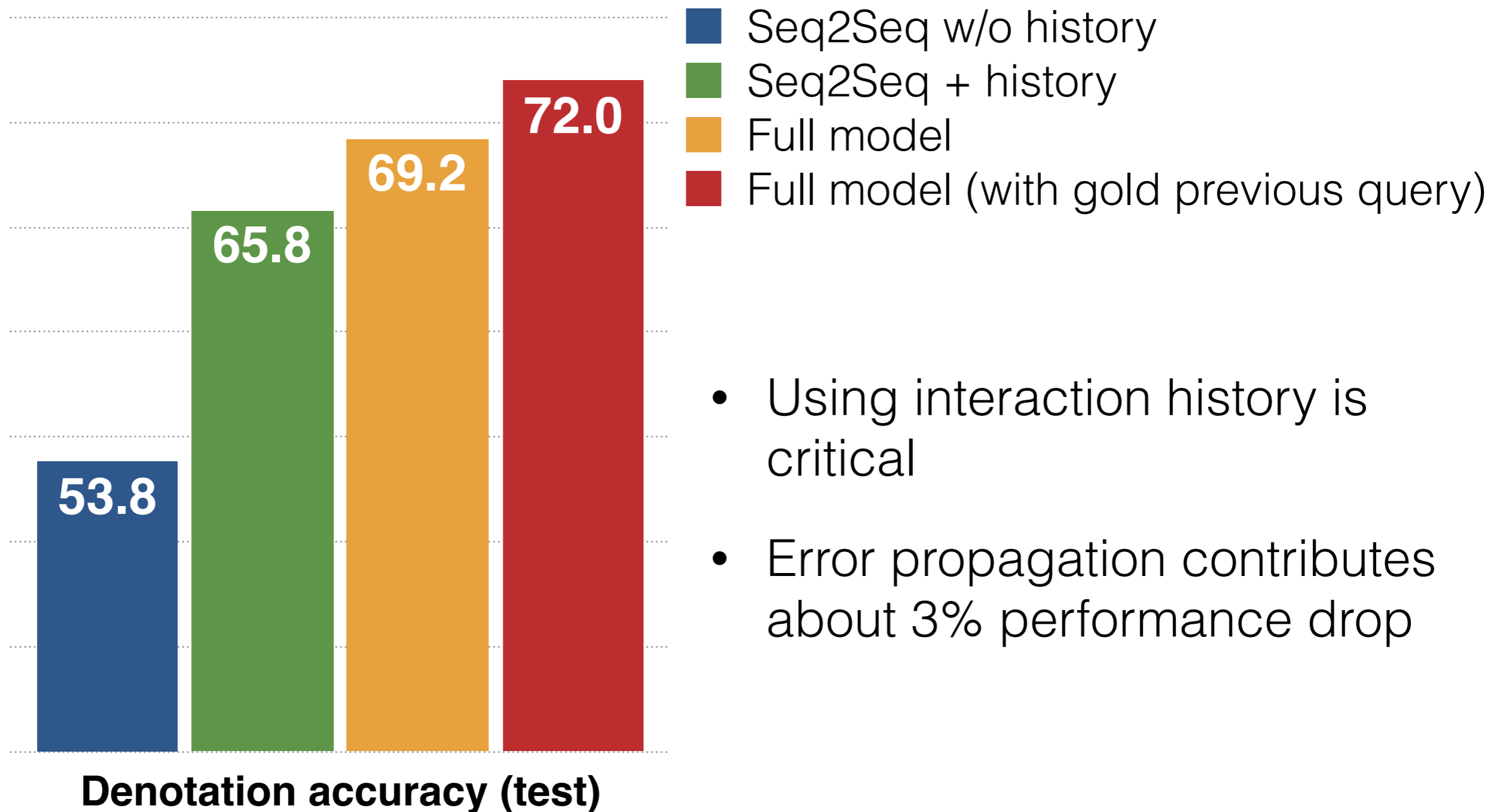
SQL Query `... city.city_name = 'SEATTLE' ...`

Experiments

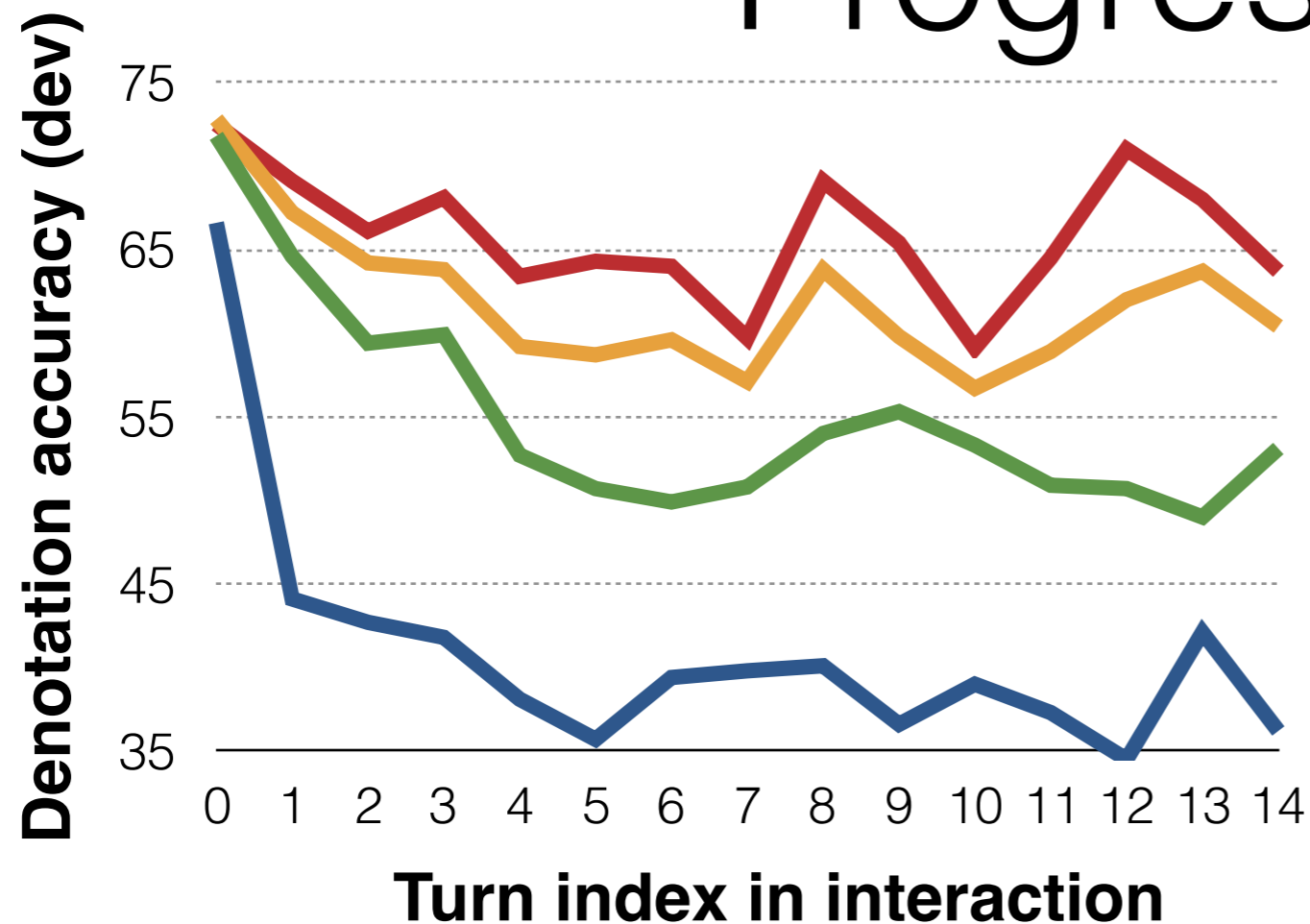
- **Seq2Seq w/o history**
seq2seq on current utterance only
- **Seq2Seq + history**
seq2seq by concatenating last four utterances
- **Full model**
use turn-level encoder
and query segment copying
- **Measure effect of error propagation:**
full model with access to gold previous query

Evaluation metric:
Denotation accuracy
(compare tables)

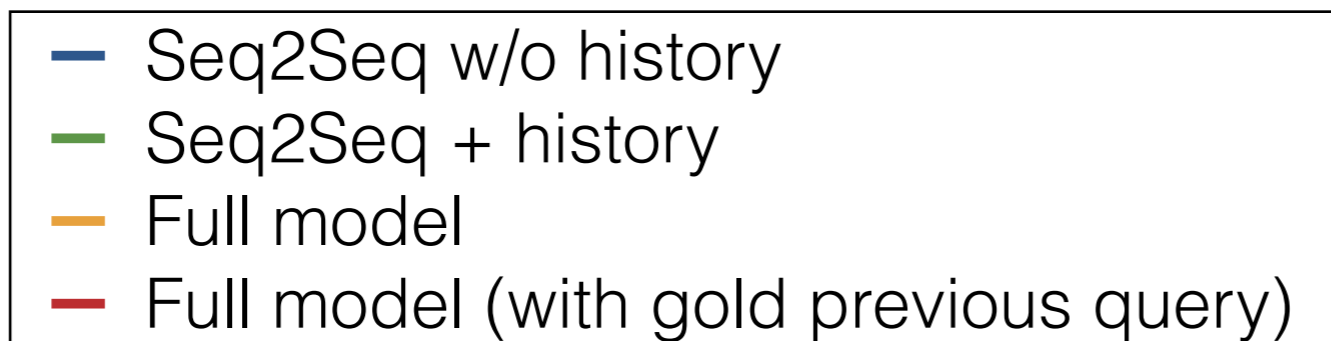
Results



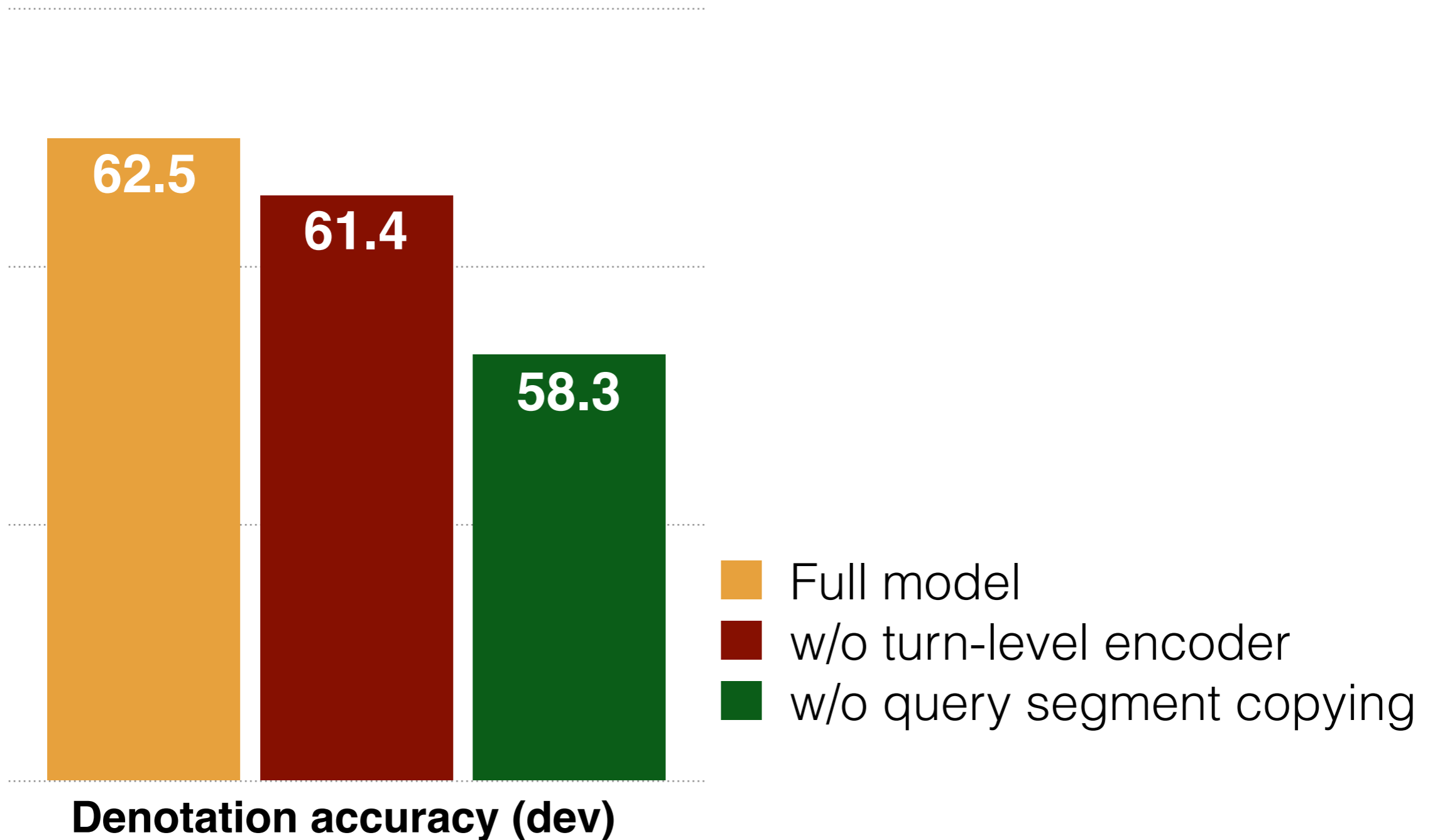
Performance as Interactions Progress



- Without interaction history, performance drops immediately
- Our model: relatively stable



Ablation Results



Error Propagation

User *Which ones arrive around 7pm?*

**SQL
Query**

```
( SELECT DISTINCT flight.flight_id FROM flight WHERE  
( flight.from_airport IN ( SELECT  
airport_service.airport_code FROM airport_service WHERE  
airport_service.city_code IN ( SELECT city.city_code FROM  
city WHERE city.city_name = 'ATLANTA' ) ) ) AND  
( flight.to_airport IN ( SELECT airport_service.airport_code  
FROM airport_service WHERE airport_service.city_code IN  
( SELECT city.city_code FROM city WHERE city.city_name =  
'BALTIMORE' ) ) ) AND ( flight.flight_days IN ( SELECT  
days.days_code FROM days WHERE days.day_name IN ( SELECT  
date_day.day_name FROM date_day WHERE date_day.year = 1991  
AND date_day.month_number = 9 AND date_day.day_number =  
6 ) ) ) AND ( flight.arrival_time >= 1630 AND  
flight.arrival_time <= 1730 ) ) ) ) ;
```

Error Propagation

User *Which ones arrive around 7pm?*

**SQL
Query**

Error: looking for flights
around 5pm

```
flight.arrival_time >= 1630 AND  
flight.arrival_time <= 1730
```

Error Propagation

User *Which kind of airplane is that?*

**SQL
Query**

```
( SELECT DISTINCT aircraft.aircraft_code FROM aircraft WHERE  
aircraft.aircraft_code IN ( SELECT  
equipment_sequence.aircraft_code FROM equipment_sequence  
WHERE equipment_sequence.aircraft_code_sequence IN ( SELECT  
flight.aircraft_code_sequence FROM flight WHERE  
( flight.arrival_time >= 1630 AND flight.arrival_time <=  
1730 AND ( flight.from_airport IN ( SELECT  
airport_service.airport_code FROM airport_service WHERE  
airport_service.city_code IN ( SELECT city.city_code FROM  
city WHERE city.city_name = 'ATLANTA' ) ) ) )
```

⋮

Error Propagation

- Selecting an incorrect segment
- Previous generated query didn't contain a necessary segment

Future work: how to mitigate error propagation?
New training procedures?

- Language understanding in long, complex interactions
- Turn-level encoder: implicit mechanism for reasoning about previous requests
- Query segment copying: explicitly derive meaning of request (SQL query) from interaction history

Thank you!

<https://github.com/clic-lab/atis>

