A Comparative Study on Schema-guided Dialog State Tracking

NAACL 2021

Jie Cao, Yi Zhang

Outlines

- Motivation
- Task Description and Datasets
- Three Comparative Studies
    i. Encoder Architectures
    ii. Supplementary training
    iii. Impact of Schema Description Style
- Q&A

Motivation

Challenges of Virtual Assistants (Task-oriented)

- Increasing number of **new services and APIs** → (new annotation, new model retraining)
- **Heterogeneous** Interfaces for **similar** services, precisely understanding overlapping functionalities.
- How to integrate **common sense and world knowledge**?

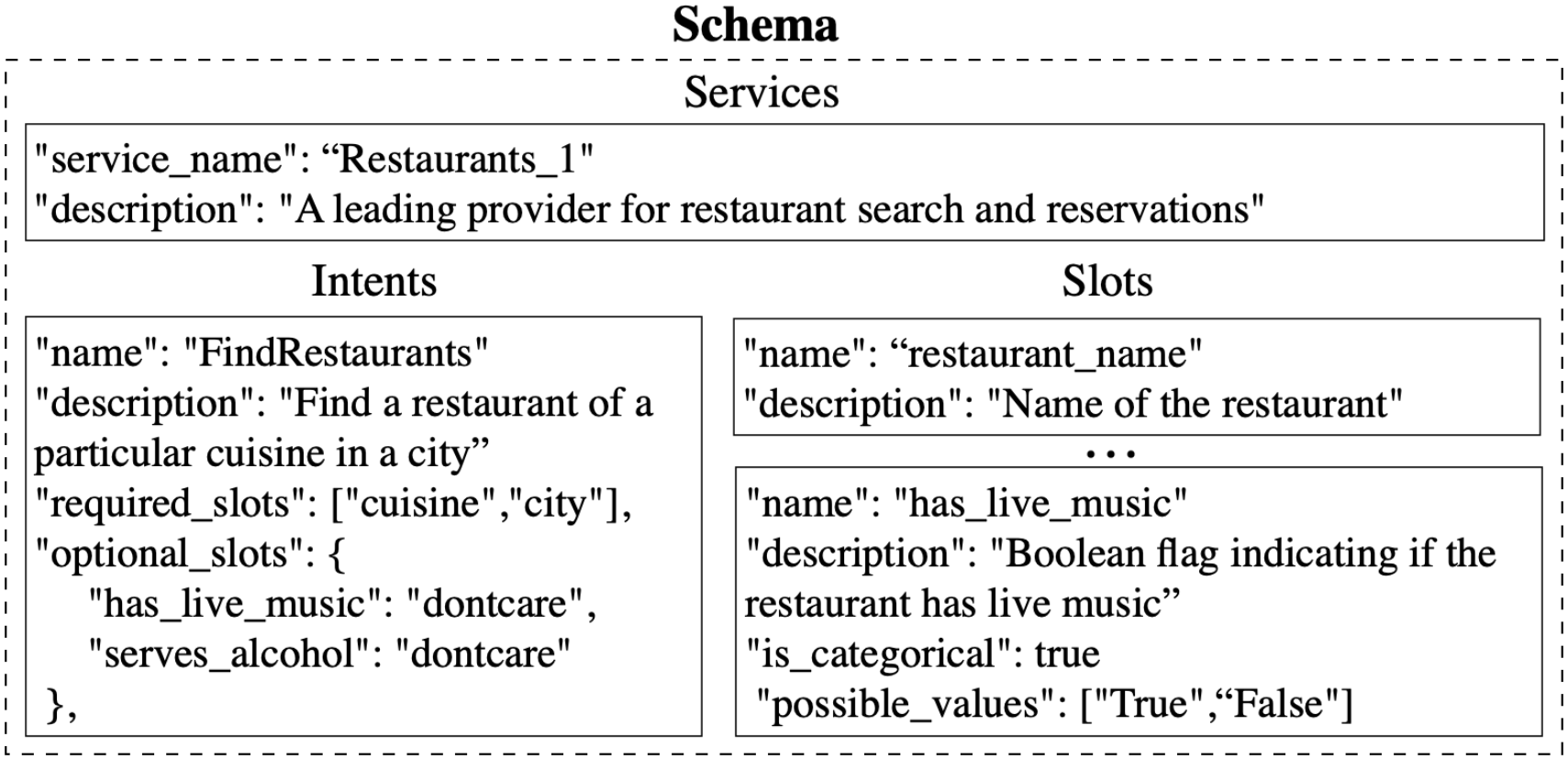
Schema-guided Dialogue State Tracking

Using **Natural language description to explain the functionalities of tags**, to help generalizing to **unseen tags** in **unseen domains.**

	Flight Service 1	**Flight Service 2**
Adding Intent Description	{ "name": "SearchFlight", "description": "Find a flight itinerary between cities for a given date", "required_slots": ….. },	{ "name": "FindFlight", "description": "Search for flights to a desitinaion", "required_slots":…… },
Adding Slot Description, Type, Possible values	{ "name": "is_direct", "description": "**Whether the flight directly arrive without any stop**", "is_categorical": True, "possible_values": ["true","false"] },	{ "name": "num_stops", "description": "**Number of layovers in the flight**", "is_categorical": true, "possible_values": ["0","1","2"] },

Schema-guided Dialogue State Tracking

Given **service schema with description** and **dialogue history,**
predict **dialog states after each user turn.**



Schema-guided Dialogue State Tracking

Dialogue

User: I am looking for **Asian** food in **SFO**
System: I found 10 restaurants in **San Francisco.**
User: Any popular one? By the way, I also want to **buy a drink**.
System: There is a nice restaurant called Butterfly Restaurant.
User: Do they have **live music**? **Where** are they **located**?
System: They do not have live music. They are at 33 The Embarcadero.

Dialogue State Tracking

User Turn 1

intent	FindRestanurants
req_slots	N/A
slot_values	
city	**SFO**
cuisine	**Asian**

User Turn 2

intent	FindRestanurants
req_slots	N/A
slot_values	
city	**SFO,** **San Francisco**
cuisine	**Asian**
alcohol	**TRUE**

User Turn 3

intent	FindRestanurants
req_slots	**has_live_music,** **street_address**
slot_values	
city	**SFO,** **San Francisco**
cuisine	**Asian**
alcohol	**TRUE**

Four Subtasks Each Turn:

1. Active Intent Classification
2. Requested Slot
3. Categorical Slot Values
   a. Boolean
   b. Predefined Values
      i. Numeric
      ii. Text
4. Non-Categorical Slot Values
   a. Span-based Value

Two Datasets:

1. Google SG-DST
2. MultiWOZ 2.2

Datasets

Datasets	Splits	Dialog	Domains	Services	Zero-shot Domains	Zero-shot Services	Function Overlapp	Collecting Method
SG-DST	Train	16142	16	26	-	-	Across-domain Within-domain	M2M
	Dev	2482	16	17	1	8		
	Test	4201	18	21	3	11		
MULTIWOZ 2.2	Train	9617	3	3	-	-	Across-domain	H2H
	Dev	2455	5	5	2	2		
	Test	2969	8	8	5	5		

SG-DST has more overlapping functionalities than MultiWOZ 2.2

Challenges: Three Comparative Studies

Q1: How to encode the dialog and schema?

- For each turn, matching the same dialog history with all schema descriptions **multiple times**.
- **Sentence-pair(SNLI)** and **Token-level classification(QA)**

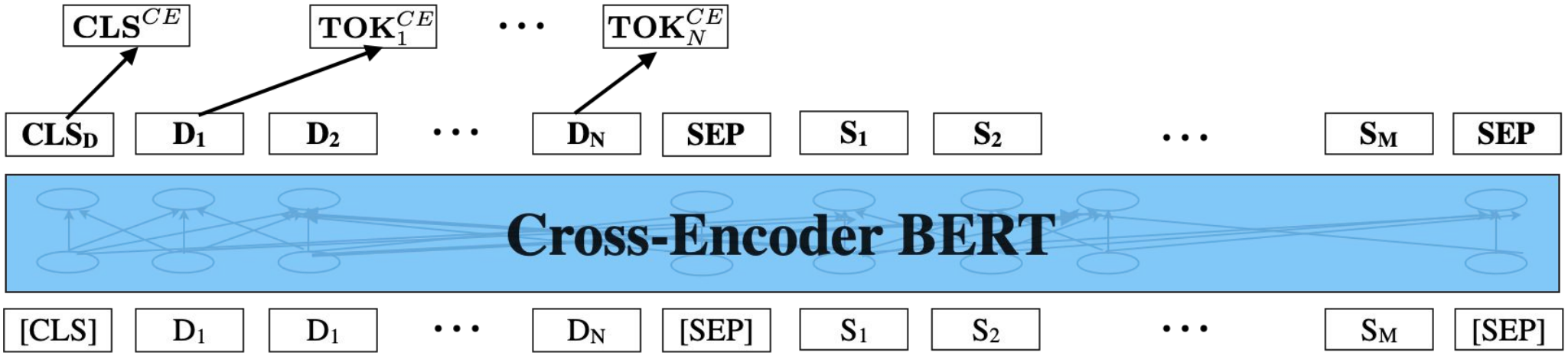
Q2: How do different supplementary trainings help?

- **Zero-shot learning** for unseen services

Q3: How the model performs on various description styles?

- Unseen service may have heterogeneous styles.

Q1: How to encode the dialog and schema? (Cross-Encoder)

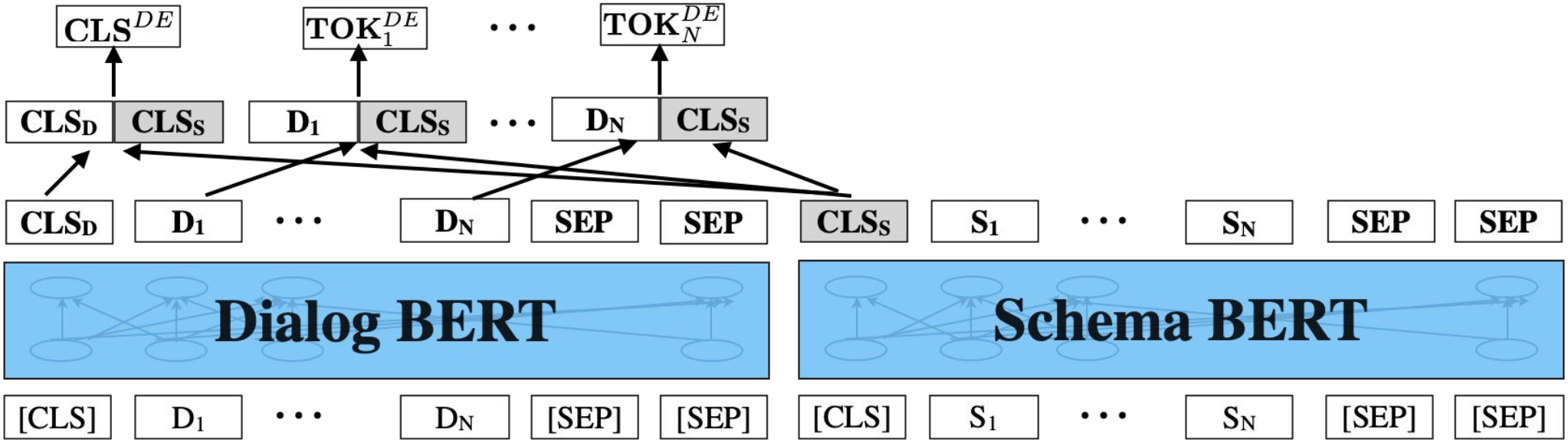


Pros:
Accurate, each representation is contextualized via full-attention

Cons: a lot of recomputing, slow

a. Dialog encoded **multiple times** within the same turn
b. Schema encoded **multiple times** across different turns.

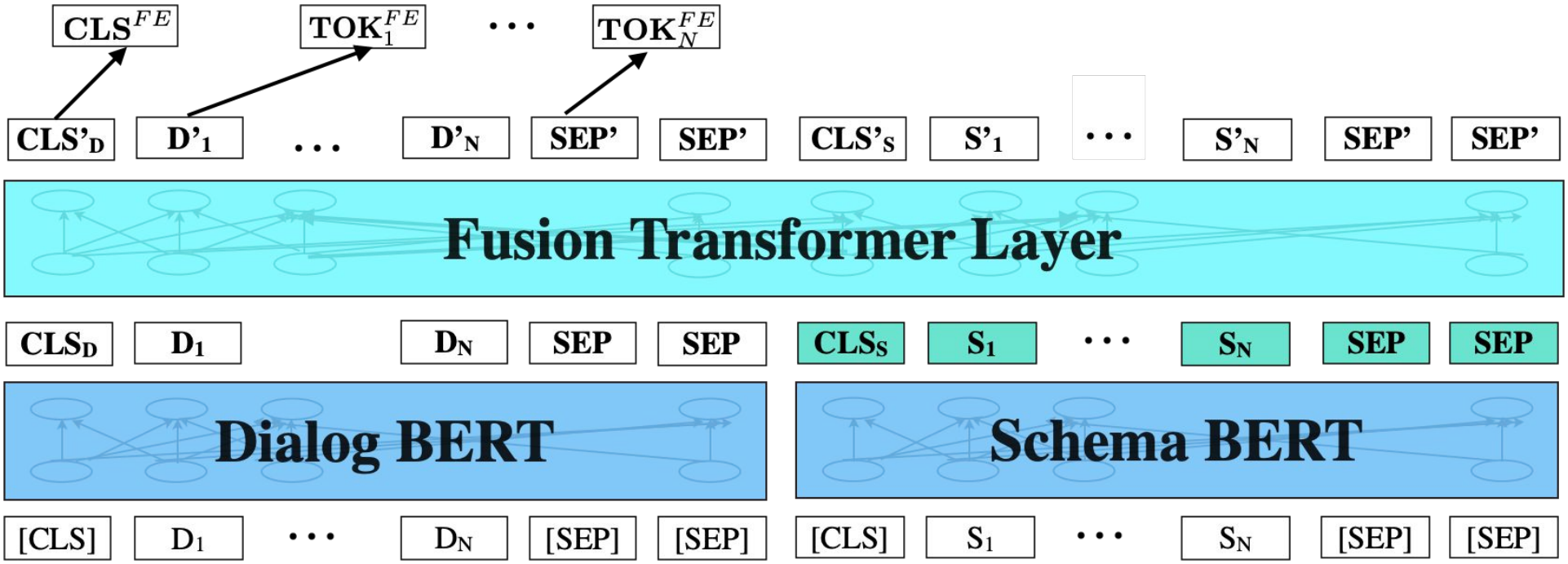
Q1: How should the dialog and schema be encoded? (Dual-Encoder)



Pros:
Encoding dialog history and schema independently, can be percomputed once and cached. Fast inference.

Cons:
a. Local self-attention
b. inaccurate

Q1: How to encode the dialog and schema?



Pros: moderate inference
Still independent precomputing.
but a thin full-attention fusion layer for better performance

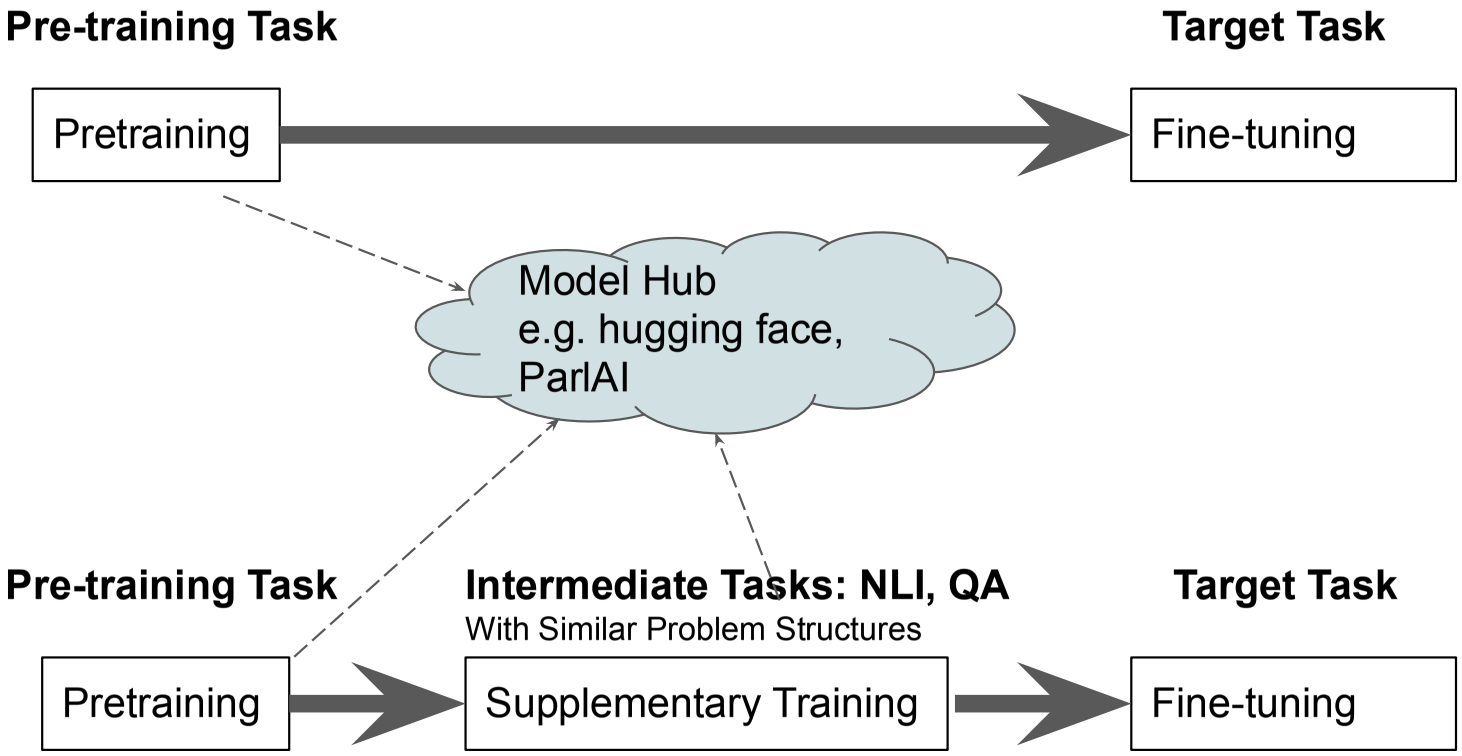
Cons:
Moderate accuracy

Q1: How to encode the dialog and schema(for 4 subtasks) ?

	SG-DST					MULTIWOZ 2.2		
Method/Task	Acc	F1	Joint Acc			Joint Acc		
	Intent	Req	Cat	NonCat	All	Cat	NonCat	All
Seen Services								
Dual-Encoder	94.51	99.62	87.92	47.77	43.20	79.20	79.34	65.64
Fusion-Encoder	94.90	**99.69**	88.94	48.78	58.52	81.37	80.58	67.43
Cross-Encoder	**95.55**	99.59	**93.68**	**91.85**	**87.58**	**85.99**	**81.02**	**71.93**
Unseen Services								
Dual-Encoder	89.73	95.20	42.44	31.62	19.51	56.92	50.82	31.83
Fusion-Encoder	90.47	95.95	48.79	35.91	22.85	57.01	52.23	33.64
Cross-Encoder	**93.84**	**98.26**	**71.55**	**74.13**	**54.54**	**59.85**	**59.62**	**38.46**

By caching the token embedding instead of the single CLS embedding, a simple partial-attention ***Fusion-Encoder*** can achieve much better performance than ***Dual-Encoder***, while still infers two times faster than ***Cross-Encoder***

Q2: How do different supplementary trainings help?



Q2: How do different supplementary trainings help?

	SG-DST							
	Intent		Req		Cat		NonCat	
	seen	unseen	seen	unseen	seen	unseen	seen	unseen
Δ_{SNLI}	+0.02	**+0.68**	+0.38	-0.38	-2.87	-1.23	-0.1	-6.25
Δ_{SQuAD}	-0.17	-1.32	-0.01	-0.33	-3.02	-5.17	-1.79	**+3.25**

- **SNLI only** helps for Intent (emphasizing the whole sentence entailment), although Req and Cat are also sentence-pair classification tasks.
- **SQuAD** consistently helps for non-categorical slot identification tasks, due to span-based retrieving
- Supplementary training helps more on unseen services

Q3: How the model performs on various description styles?

Background:

1. To compatible with previous tag-based DST system, many previous papers show **simply adding question format to those tags** may help.
   a. Is name-based description enough?
   b. Does question format helps?
2. **Unseen services may use different description style**
   a. **heterogeneous evaluation?**

style	Intent Description	Slot Description
Identifer	intent_1	slot_4
NameOnly	CheckBalance	account_type
Q-Name	Is the user intending to CheckBalance?	What is the value of acctount_type ?
Orig	Check the amount of money in a user's bank account	The account type of the user
Q-Orig	Does the user want to check the amount of money in the bank account ?	What is the account type of the user ?
Orig-Para	Check the balance of the user's bank account	Type of the user account

Q3: How the model performs on various description styles? (homogeneous)

Style\Task	SG-DST				MULTIWOZ 2.2	
	Intent	Req	Cat	NonCat	Cat	NonCat
Identifer	61.16	91.48	62.47	30.19	34.25	52.28
NameOnly	**94.24**	98.84	74.01	75.63	53.72	56.18
Q-Name	93.31	**98.86**	74.36	74.86	**54.19**	56.17
Orig	93.01	98.55	74.51	75.76	52.19	57.20
Q-Orig	93.42	98.51	**76.64**	**76.60**	53.61	**57.80**

Is named-based description enough?

- Most name are meaningful, and perform **not bad,** especially **on** Intent/Req subtasks
- Rich description outperforms the name-based on **NonCat**, but inconsistent on other tasks.

Q3: How the model performs on various description styles? (homogeneous)

Style/Dataset	SG-DST		MULTIWOZ 2.2	
	seen	unseen	seen	unseen
Orig	-1.79	**+3.25**	-2.21	**+4.27**
Q-Orig	-2.01	**+8.84**	-1.28	**+3.06**
NameOnly	-1.49	-0.11	+0.58	**+1.77**
Q-Name	-2.98	**+1.04**	-0.32	**+1.25**

Is question-format helpful?

- It generally helps on Cat/NonCat
- Adding it to rich description will benefit more from SQuAD2 supplementary training on unseen. However, not on MultiWOZ.

Q3: How the model performs on various description styles? (Heterogeneous)

Style\Task	SG-DST							
	Intent(Acc)		Req(F1)		Cat(Joint Acc)		NonCat(Joint Acc)	
	mean	Δ	mean	Δ	mean	Δ	mean	Δ
NameOnly	82.47	-11.47	96.92	-1.64	61.37	-5.54	56.53	-14.68
Q-Name	**93.27**	+0.58	**97.88**	-0.76	68.55	+2.63	62.92	-6.30
Orig	79.47	-12.70	97.42	-0.74	**68.58**	-0.3	**66.72**	-3.11
Q-Orig	84.57	-8.24	96.70	-1.45	68.40	-2.89	56.17	-15.00
	para	Δ	para	Δ	para	Δ	para	Δ
NameOnly	**92.22**	-1.74	97.69	-0.87	67.39	-0.7	67.17	-4.04
Orig	91.54	**-0.63**	**98.42**	**+0.26**	**71.74**	**+2.86**	**67.68**	**-2.16**

What if unseen service in different description styles?

- For unseen styles, all tasks surfer from inconsistencies, though to varying degrees
- For paraphrased styles, richer description are relatively more robust than named-based descriptions.

Takeaways

1. Cross-Encoder > Fusion-Encoder > Dual-Encoder in accuracy, while opposite on inference speed.

2. To support low-resource unseen services, we quantified the gain via supplementary training on different subtasks.

3. Simple named-based description are actually meaningful, and they perform not bad, but not as robust as rich description in most cases.

4. All subtasks suffers from inconsistencies when using heterogeneous description on unseen services, which requires future work on more robust cross-style schema-guided dialog modeling.

Q&A?

Thanks