

# Wizard-of-Seoul

## EDA and Data processing

---

김성동

(NAVER AI LAB)

# 1. Introduction to Wizard-of-Seoul (WOS)

- 1.1 Overview
- 1.2 Data Statistic
- 1.3 Data Construction
- 1.4 Data Format

# 2. Processing WOS Dataset

- 2.1 Implementation Flow
- 2.2 DSTInputExample
- 2.3 Slot Meta
- 2.4 DSTEvaluator

# 3. Taxonomy of DST models

- 3.1 DST Recap
- 3.2 Model Concept
- 3.3 How to represent  $p(V)$ ?
- 3.4 Taxonomy of DST models

# 1.

## Introduction to Wizard-of-Seoul

강의에서 다루게 될 데이터셋에 대해 알아보니다.

# 1.1 Overview

## Multi-Domain Dialogue State Tracking



MultiWOZ

Wizard-of-Seoul (Ours)

- MultiWOZ는 TOD의 가장 대중적인 벤치마크 데이터셋
- (영국) Cambridge 여행 Assistant 시나리오 (총 7가지의 멀티 도메인)
- **Multi-Domain DST**는 도메인 간의 전이 (Domain Transition) 상에서 나타나는 **Lexical Entailment, Co-referencing** 등을 해결하는 것이 주요 Challenge
- **Wizard-of-Seoul (WOS)**은 **KLUE Project**의 일환으로 새롭게 구축된 한국어 DST 벤치마크
- 유사하게 (대한민국) **Seoul 여행 Assistant** 시나리오 (**총 5가지의 멀티 도메인**)
  - Annotation interface를 개선하여 Typos 등 Error types 해소
  - 많은 수의 Boolean type Slot을 포함하고 있어 Abstractiveness 극대화
  - Counterfactual Goal을 사용하여 내재한 Bias를 최소화

# 1.2 Multi-Domain Dialogue State Tracking

$r_1$   
 $u_1$  안녕하세요. 남쪽에 위치한 숙소 좀 검색하려고요

$B_1$  [(숙소-지역, 서울 남쪽)]

...

$B_5$  [(숙소-지역, 서울 남쪽), (숙소-이름, 안즈 서울), ...]

$r_6$  네 예약 완료되었습니다, 더 필요한거 없으세요?  
 $u_6$  아 그리고, 제가 숙소랑 가까운 곳에 위치한 식당도 찾고 있거든요?

$B_6$  [(숙소-지역, 서울 남쪽), (숙소-이름, 안즈 서울), ..., (식당-지역, 서울 남쪽)]

...

$B_{10}$  [..., (숙소-이름, 안즈 서울), ..., (식당-이름, 식객), ...]

$r_{11}$  제가 더 도울 일은 없을까요?  
 $u_{11}$  네 그리고 숙소에서 식당으로 가는 택시 좀 잡아줘요 마지막이에요!

$B_{11}$  [..., (숙소-이름, 안즈 서울), ..., (식당-이름, 식객), ..., (택시-출발지, 안즈 서울), (택시-도착지, 식객)]

Domain Transition에 의해  
 Abstractive하게 발생하는 State  
 변화를 감지해야 하며,

이에 따라  
 Context에서 적절한 Value를  
 Co-referencing해서 Tracking하  
 야 한다.

## 1.2 Data Statistic

	MultiWOZ	WoS
Language	En	Ko
Meta Domain	Travel	Travel
# Domain	7	5
# (Informable) Slot	35	45
# Value	1983	2821
# Dialogue	10,421	10,000
# Avg turn	13.68	14.67
# tokens per turn	13.13	7.7

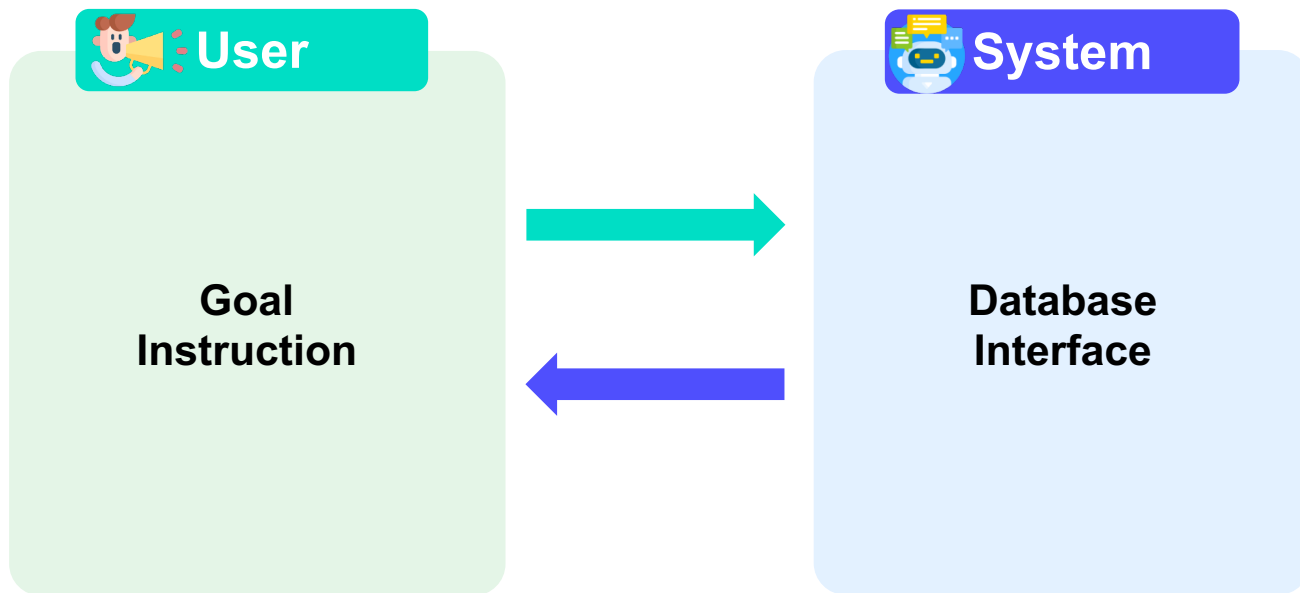
# 1.2 Data Statistics

Task Schema == Predefined Scenario

	숙소	식당	관광	택시	지하철
# Informable Slot	15	13	9	5	3
# Boolean Type	8	6	6	0	0
# Requestable Slot	8	10	8	3	6
Booking (예약)	O	O	X	X	X

# 1.3 Data Construction

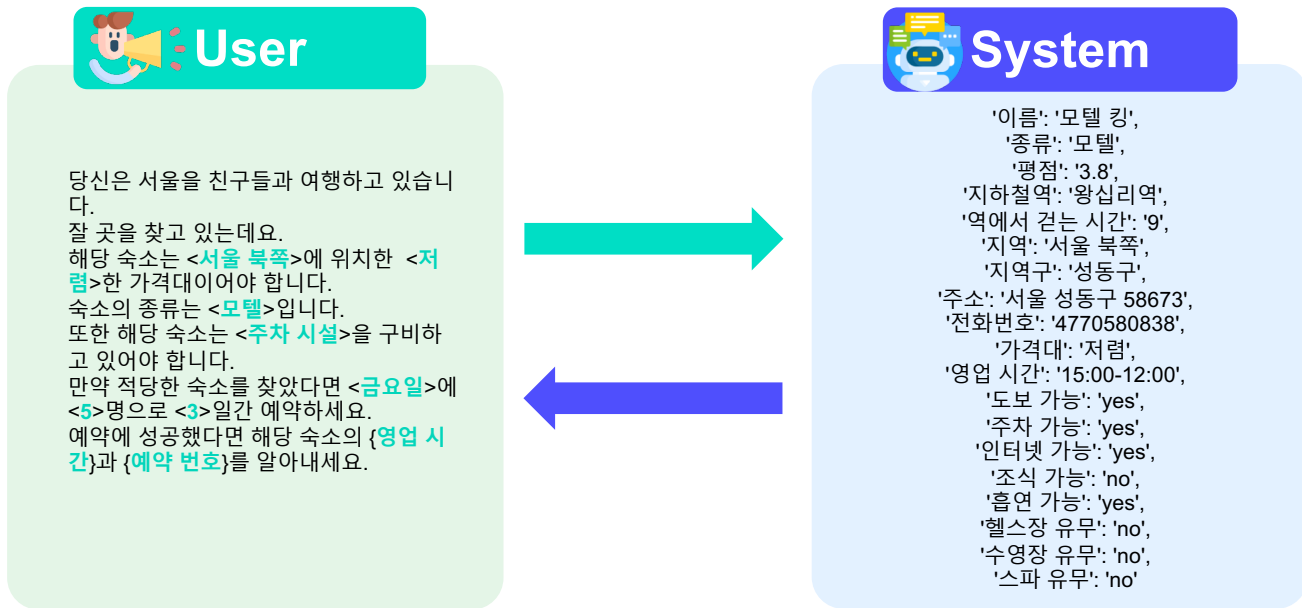
Wizard-of-Oz framework





# 1.3 Data Construction

## Wizard-of-Oz framework



# 1.3 Data Construction

## Easy-to-Follow Goal Instruction

**user** 서울 북쪽에 있는 저렴한 가격대의 모텔에 예약하고 싶어요

**sys** 안녕하세요. 더 원하시는 조건이 있으신가요?

**user** 네. 주차장이 있는 곳이면 좋겠어요.

**sys** 네. 원하시는 조건의 숙소로 힐 모텔과 모텔 킹이 있습니다. 모텔 킹의 경우 가격대비 서비스가 좋아 3.8점의 좋은 평점을 받고 있는데 예약 해 드릴까요?

**user** 네. 모텔 킹으로 예약 해 주세요. 금요일 5명으로 3일 예약해주세요.

**sys** 네. 모텔 킹으로 예약 도와드리겠습니다. 궁금한 점 있으세요?

**user** 영업시간과 예약 번호를 알고싶어요.

**sys** 네. 모텔 킹의 영업 시간은 15:00-12:00이고, 예약 번호는 YZWC5입니다. 더 궁금한 점 있으세요?

**user** 아뇨. 감사합니다.

**sys** 네. 이용해주셔서 감사합니다.

user

Messages

send

### Goal Instruction

당신은 서울을 친구들과 여행하고 있습니다.

잘 곳을 찾고 있는데요.

해당 숙소는 <서울 북쪽>에 위치한 <저렴>한 가격대이어야 합니다.

숙소의 종류는 <모텔>입니다.

또한 해당 숙소는 <주차 시설을 구비하고 있어야 합니다>.

만약 적당한 숙소를 찾았다면 <금요일>에 <5>명으로 <3>일간 예약하세요.

예약에 성공했다면 해당 숙소의 {영업 시간}과 {예약 번호}를 알아내세요.

FAQ

### Notice

Be careful..

FAQ

User Role을 맡은 Worker는 주어진 Goal Instruction에 따라 대화를 이어 나가도록 요청 받는다.

Goal Instruction은 유저가 달성해야 할 목적 (User Goal)이 유저의 Context, Informable Slot과 Requestable Slot을 포함하여 자연어로 기술 되어 있다.

# 1.3 Data Construction

## Annotation error types in the MultiWOZ

Type	Conversation	MultiWOZ 2.0	MultiWOZ 2.1
Delayed Markups	User: I'd also like to try a Turkish restaurant. Is that possible? Agent: I'm sorry but the only restaurants in that part of town serve either Asian food or African food. User: I don't mind changing the area. I just need moderate pricing and want something that serves Turkish food.	restaurant.food: None	restaurant.food: Turkish
Multi-annotations	User: Can you tell me more about Cambridge Belfry	hotel.name: The Cambridge Belfry attraction.name: belf	hotel.name: The Cambridge Belfry attraction.name: None
Mis-annotations	User: Yes, I need to leave on Thursday and am departing from London Liverpool Street.	train.leaveAt: Thursday train.day: Not Mentioned	train.leaveAt: None train.day: Thursday
Typos	Although, I could use some help finding an attraction in the centre of town.	attraction.area: cent	attraction.area: Centre
Forgotten values	User: No particular price range, but I do need a restaurant that is available to book 7 people on Friday at 19:15.	restaurant.pricerange: None	restaurant.pricerange: Dontcare
Value Canonicalization	User: I think you <small>train_db_210323.tsv</small> got it wrong. Cambridge to Bishop Stafford on Thursday.	train.destination: Bishop Stortford	train.destination: Bishops Stortford

Table 3: Examples of annotation errors between MultiWOZ 2.0 and 2.1

# 1.3 Data Construction

## Easy-to-Operate system interface

지하철	관광	숙소	식당	택시
숙소-가격대 지렴	TRACKING	숙소-종류 모텔	TRACKING	
숙소-지역 서울 북쪽	TRACKING	숙소-주차 가능	TRACKING	
숙소-인터넷 가능	TRACKING	숙소-헬스장 유무	TRACKING	
숙소-수영장 유무	TRACKING	숙소-조식 가능	TRACKING	
숙소-스파 유무	TRACKING	숙소-흡연 가능	TRACKING	
숙소-도보 가능	TRACKING	숙소-예약 요일 금요일	TRACKING	
숙소-예약 필수 5	TRACKING	숙소-예약 기간 3	TRACKING	
숙소-이름 모텔 킹	TRACKING			

LOOKUP

BOOKING

System Role을 맡은 Worker는  
답변을 만들기 전에 다음과 같은 Interface  
를 이용하여, **Dialogue State를 Tracking**

DB를 조회하거나 **Booking**을 생성한 후

그 결과를 바탕으로 답변을 만들어야 한다.

**Dropdown Interface** (객관식 형태의 후보  
제공)을 통해 Typos, Multi-annotations,  
Mis-annotations, Value Canonicalization  
등의 에러 타입 해소

# 1.3 Data Construction

## Data Example

user 서울 북쪽에 있는 저렴한 가격대의 모텔에 예약하고 싶어요

숙소-가격대-저렴 ✕    숙소-종류-모텔 ✕    숙소-지역-서울 북쪽 ✕    ADD

sys 안녕하세요. 더 원하시는 조건이 있으신가요?

user 네. 주차장이 있는 곳이면 좋겠어요.

숙소-가격대-저렴 ✕    숙소-종류-모텔 ✕    숙소-지역-서울 북쪽 ✕    ADD

sys 네. 원하시는 조건의 숙소로 힐 모텔과 모텔 킹이 있습니다. 모텔 킹의 경우 가격대비 서비스가 좋아 3.8점의 좋은 평점을 받고 있는데 예약 해 드릴까요?

user 네. 모텔 킹으로 예약 해 주세요. 금요일 5명으로 3일 예약 해주세요.

숙소-가격대-저렴 ✕    숙소-종류-모텔 ✕    숙소-지역-서울 북쪽 ✕    숙소-예약 요일-금요일 ✕    숙소-예약 명수-5 ✕    숙소-예약 기간-3 ✕    숙소-이름-모텔 킹 ✕    ADD

sys 네. 모텔 킹으로 예약 도와드리겠습니다. 궁금한 점 있으세요?

user 영업시간과 예약 번호를 알고싶어요.

숙소-가격대-저렴 ✕    숙소-종류-모텔 ✕    숙소-지역-서울 북쪽 ✕    숙소-예약 요일-금요일 ✕    숙소-예약 명수-5 ✕    숙소-예약 기간-3 ✕    숙소-이름-모텔 킹 ✕    ADD

sys 네. 모텔 킹의 영업 시간은 15:00-12:00이고, 예약 번호는 YZWC5입니다. 더 궁금한 점 있으세요?

user 아뇨. 감사합니다.

숙소-가격대-저렴 ✕    숙소-종류-모텔 ✕    숙소-지역-서울 북쪽 ✕    숙소-예약 요일-금요일 ✕    숙소-예약 명수-5 ✕    숙소-예약 기간-3 ✕    숙소-이름-모텔 킹 ✕    ADD

sys 네. 이용해주셔서 감사합니다.

# 1.4 Data Format

```
{
  "dialogue": [
    {
      "role": "user",
      "state": [
        "숙소-가격대-저렴",
        "숙소-종류-모텔",
        "숙소-지역-서울 북쪽"
      ],
      "text": "서울 북쪽에 있는 저렴한 가격대의 모텔에 예약하고 싶어요"
    },
    {
      "role": "sys",
      "text": "안녕하세요. 더 원하시는 조건이 있으신가요?"
    },
    {
      "role": "user",
      "state": [
        "숙소-가격대-저렴",
        "숙소-종류-모텔",
        "숙소-지역-서울 북쪽"
      ],
      "text": "네. 주차장이 있는 곳이면 좋겠어요."
    },
    {
      "role": "sys",
      "text": "네. 원하시는 조건의 숙소로 힐 모텔과 모텔 킹이 있습니다. 모텔 킹의 경우 가격대비 서비스가 좋아 3.8점의 좋은 평점을 받고 있는데 예약 해 드릴까요?"
    },
    {
      "role": "user",
      "state": [
        "숙소-가격대-저렴",
        "숙소-종류-모텔",
        "숙소-지역-서울 북쪽",
        "숙소-예약 요일-금요일",
        "숙소-예약 명수-5",
        "숙소-예약 기간-3",
        "숙소-이름-모텔 킹"
      ],
      "text": "네. 모텔 킹으로 예약 해 주세요. 금요일 5명으로 3일 예약 해주세요."
    },
  ],
}
```

- Json Format - Each examples consists of information below:
  - **dialogue\_idx** - **str** : unique id for a dialogue
  - **domains** - **List[str]** : single or multi domains for the dialogue
  - **dialogue** - **List[Dict]** : turn-level information that contains attributes below:
    - **role** - **str** : a speaker of the turn, user or sys
    - **text** - **str** : an utterance (transcription)
    - **state** - **List[str]** : dialogue-state formed in **DOMAIN-SLOT-VALUE**

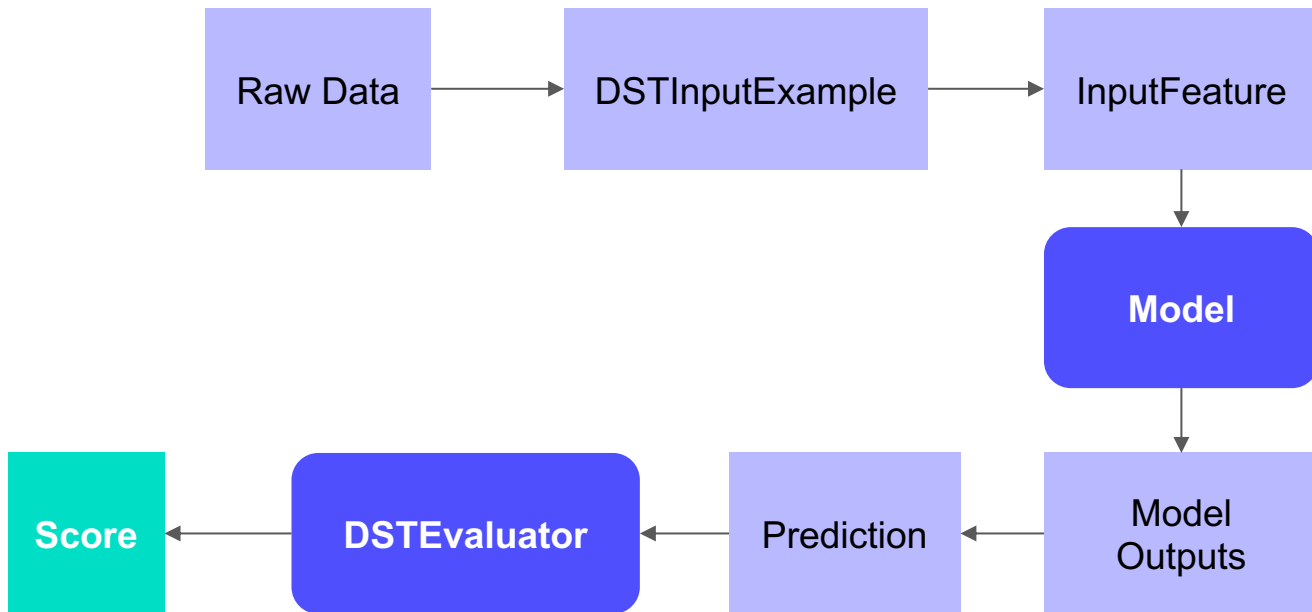
## 2.

# Processing WOS Dataset

모델 구현 흐름을 보며,

WOS 데이터셋을 어떻게 다루면 좋을지 생각해봅시다.

## 2.1 Implementation Flow

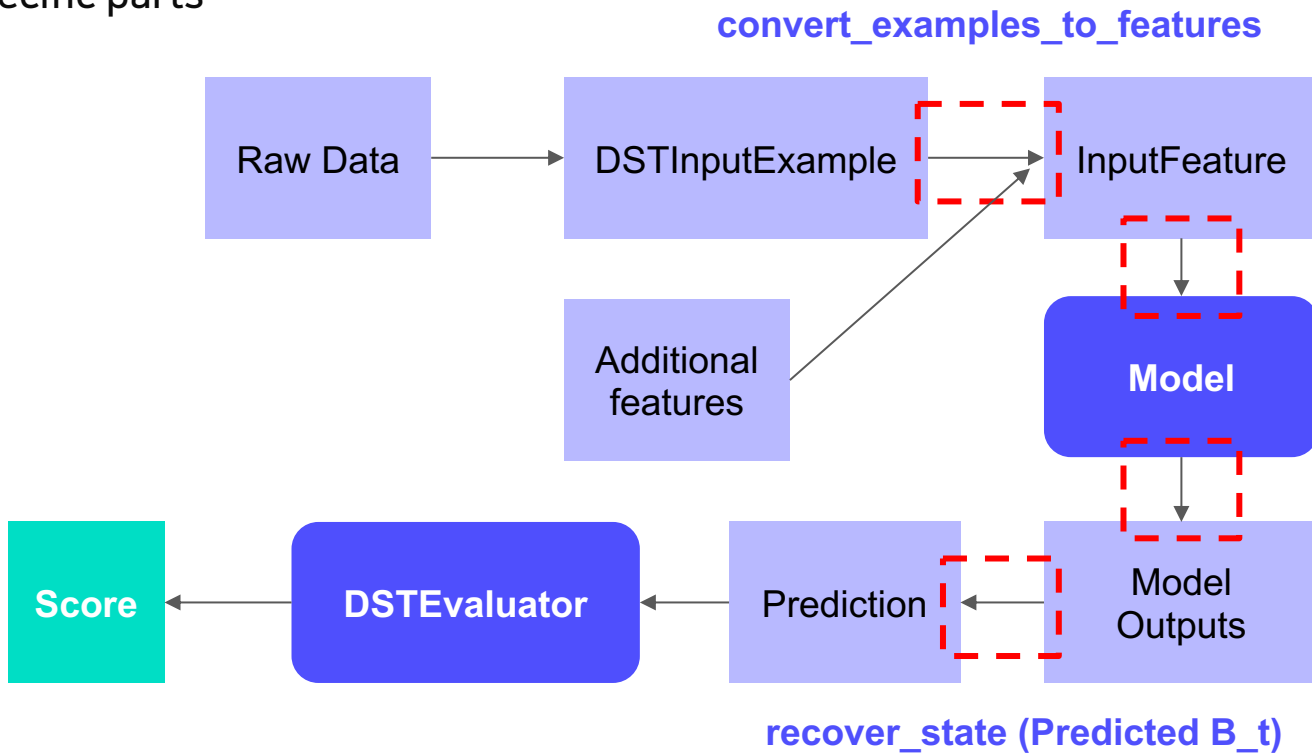




## 2.1 Implementation Flow

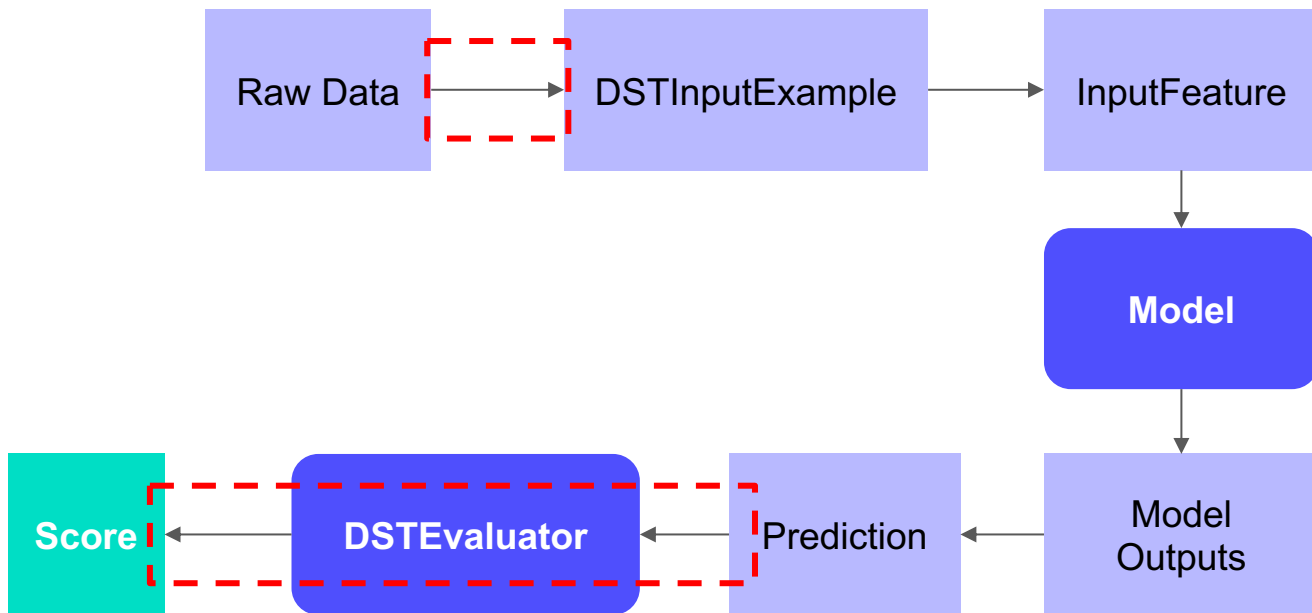
Processing WOS Dataset

Model-Specific parts



## 2.1 Implementation Flow

Global parts: Today's Scope



## 2.2 DSTInputExample

```
@dataclass
class DSTInputExample:
    guid: str
    context_turns: List[str]
    current_turn: List[str]
    label: Optional[List[str]] = None
```

**guid**: dialogue\_idx + turn\_idx

**context\_turns**: turn t(현재 target turn) 이전의  
dialogue context  $D_{\{t-1\}}$

**current\_turn**: 현재 turn에서 시스템, 유저 발화  
( $r_t, u_t$ ) or ( $u_t, r_t$ )

**label**: Dialogue State at turn t ( $B_t$ )

## 2.2 DSTInputExample

```
def get_examples_from_dialogue(dialogue, user_first=False):
    guid = dialogue['dialogue_idx']
    examples = []
    history = []
    d_idx = 0
    for idx, turn in enumerate(dialogue['dialogue']):
        if turn['role'] != 'user':
            continue

        if idx:
            sys_utter = dialogue['dialogue'][idx - 1]['text']
        else:
            sys_utter = ''

        user_utter = turn['text']
        state = turn['state']
        context = deepcopy(history)
        if user_first:
            current_turn = [user_utter, sys_utter]
        else:
            current_turn = [sys_utter, user_utter]
        examples.append(DSTInputExample(guid=f'{guid}-{d_idx}',
                                         context_turns=context,
                                         current_turn=current_turn,
                                         label=state))

        history.append(sys_utter)
        history.append(user_utter)
        d_idx += 1
    return examples
```

dialogue 하나를 받아서 **T**(총 turn 갯수)만큼의  
**List of DSTInputExamples**를 반환하는 함수

## 2.2 DSTInputExample

```
get_examples_from_dialogue(dialogue)
```

Last executed at 2021-03-22 17:16:58 in 22ms

```

DSTInputExample(guid='snowy-hat-8324:관광_식당_11-0', context_turns=[], current_turn=['', '서울 중랑에 있는 박물관을 찾아주세요'], label=['관광-종류-박물관', '관광-지역-서울 중랑']),
DSTInputExample(guid='snowy-hat-8324:관광_식당_11-1', context_turns=['', '서울 중랑에 있는 박물관을 찾아주세요'], current_turn=['안녕하세요. 문화역서를 284은 어디서나요?'], label=['관광-종류-박물관', '관광-지역-서울 중랑', '관광-이름-문화역서를 284']),
기 평점은 말해주셨구 전화번호가 어떻게되나요?', label=['관광-종류-박물관', '관광-지역-서울 중랑', '관광-이름-문화역서를 284']),
DSTInputExample(guid='snowy-hat-8324:관광_식당_11-2', context_turns=['', '서울 중랑에 있는 박물관을 찾아주세요'], '안녕하세요. 문화역서를 284은 어디서나요? 평점도 4점으로 방문객들에게 좋은 평가를 받고 있습니다.', '좋네요 거기 평점은 말해주셨구 전화번호가 어떻게되나요?', current_turn=['전화번호는 983880764입니다. 더 필요하신 게 있으실까요?'], '네 관광지와 같은 지역의 한식당을 가고싶는데 야외석이 있어어되요'], label=['관광-종류-박물관', '관광-지역-서울 중랑', '관광-이름-문화역서를 284', '식당-지역-서울 중랑', '식당-종류-한식당', '식당-야외석 유무=yes']),
DSTInputExample(guid='snowy-hat-8324:관광_식당_11-3', context_turns=['', '서울 중랑에 있는 박물관을 찾아주세요'], '안녕하세요. 문화역서를 284은 어디서나요? 평점도 4점으로 방문객들에게 좋은 평가를 받고 있습니다.', '좋네요 거기 평점은 말해주셨구 전화번호가 어떻게되나요?', '전화번호는 983880764입니다. 더 필요하신 게 있으실까요?', '네 관광지와 같은 지역의 한식당을 가고싶는데 야외석이 있어어되요'], current_turn=['생각하고 계산 가격대가 있으신가요?', '음.. 저렴한 가격대에 있네요?'], label=['관광-종류-박물관', '관광-지역-서울 중랑', '관광-이름-문화역서를 284', '식당-가격대-저렴', '식당-지역-서울 중랑', '식당-종류-한식당', '식당-야외석 유무=yes']),
DSTInputExample(guid='snowy-hat-8324:관광_식당_11-4', context_turns=['', '서울 중랑에 있는 박물관을 찾아주세요'], '안녕하세요. 문화역서를 284은 어디서나요? 평점도 4점으로 방문객들에게 좋은 평가를 받고 있습니다.', '좋네요 거기 평점은 말해주셨구 전화번호가 어떻게되나요?', '전화번호는 983880764입니다. 더 필요하신 게 있으실까요?', '네 관광지와 같은 지역의 한식당을 가고싶는데 야외석이 있어어되요'], '생각하고 계산 가격대가 있으신가요?', '음.. 저렴한 가격대에 있네요?', current_turn=['최종하지만 저렴한 가격대에는 없으시네요.', '그럼 비싼 가격대로 다시 찾아주세요'], label=['관광-종류-박물관', '관광-지역-서울 중랑', '관광-이름-문화역서를 284', '식당-가격대-비싼', '식당-지역-서울 중랑', '식당-종류-한식당', '식당-야외석 유무=yes']),
DSTInputExample(guid='snowy-hat-8324:관광_식당_11-5', context_turns=['', '서울 중랑에 있는 박물관을 찾아주세요'], '안녕하세요. 문화역서를 284은 어디서나요? 평점도 4점으로 방문객들에게 좋은 평가를 받고 있습니다.', '좋네요 거기 평점은 말해주셨구 전화번호가 어떻게되나요?', '전화번호는 983880764입니다. 더 필요하신 게 있으실까요?', '네 관광지와 같은 지역의 한식당을 가고싶는데 야외석이 있어어되요'], '생각하고 계산 가격대가 있으신가요?', '음.. 저렴한 가격대에 있네요?', '최종하지만 저렴한 가격대에는 없으시네요.', '그럼 비싼 가격대로 다시 찾아주세요'], current_turn=['외계인의맛집은 어디서나요? 대표 메뉴는 한정식입니다.', '좋습니다 토요일 18:00에 1명 예약가능하나요?'], label=['관광-종류-박물관', '관광-지역-서울 중랑', '관광-이름-문화역서를 284', '식당-가격대-비싼', '식당-지역-서울 중랑', '식당-종류-한식당', '식당-야외석 유무=yes', '식당-예약 요일-토요일', '식당-예약 명수=1', '식당-이름-외계인의맛집']),
DSTInputExample(guid='snowy-hat-8324:관광_식당_11-6', context_turns=['', '서울 중랑에 있는 박물관을 찾아주세요'], '안녕하세요. 문화역서를 284은 어디서나요? 평점도 4점으로 방문객들에게 좋은 평가를 받고 있습니다.', '좋네요 거기 평점은 말해주셨구 전화번호가 어떻게되나요?', '전화번호는 983880764입니다. 더 필요하신 게 있으실까요?', '네 관광지와 같은 지역의 한식당을 가고싶는데 야외석이 있어어되요'], '생각하고 계산 가격대가 있으신가요?', '음.. 저렴한 가격대에 있네요?', '최종하지만 저렴한 가격대에는 없으시네요.', '그럼 비싼 가격대로 다시 찾아주세요', '외계인의맛집은 어디서가요? 대표 메뉴는 한정식입니다.', '좋습니다 토요일 18:00에 1명 예약가능하나요?', '네 예약가능하나요?', '네 거기 주류는 판매하나요?주자는 가능한가요?', current_turn=['주류는 판매하고 있고 주차도 가능합니다. 더 궁금하신 것 있으신가요?'], '안녕요'], label=['관광-종류-박물관', '관광-지역-서울 중랑', '관광-이름-문화역서를 284', '식당-가격대-비싼', '식당-지역-서울 중랑', '식당-종류-한식당', '식당-야외석 유무=yes', '식당-예약 요일-토요일', '식당-예약 명수=1', '식당-이름-외계인의맛집']),
DSTInputExample(guid='snowy-hat-8324:관광_식당_11-7', context_turns=['', '서울 중랑에 있는 박물관을 찾아주세요'], '안녕하세요. 문화역서를 284은 어디서나요? 평점도 4점으로 방문객들에게 좋은 평가를 받고 있습니다.', '좋네요 거기 평점은 말해주셨구 전화번호가 어떻게되나요?', '전화번호는 983880764입니다. 더 필요하신 게 있으실까요?', '네 관광지와 같은 지역의 한식당을 가고싶는데 야외석이 있어어되요'], '생각하고 계산 가격대가 있으신가요?', '음.. 저렴한 가격대에 있네요?', '최종하지만 저렴한 가격대에는 없으시네요.', '그럼 비싼 가격대로 다시 찾아주세요', '외계인의맛집은 어디서가요? 대표 메뉴는 한정식입니다.', '좋습니다 토요일 18:00에 1명 예약가능하나요?', '네 예약가능하나요?', '네 거기 주류는 판매하나요?주자는 가능한가요?', current_turn=['주류는 판매하고 있고 주차도 가능합니다. 더 궁금하신 것 있으신가요?'], '안녕요'], label=['관광-종류-박물관', '관광-지역-서울 중랑', '관광-이름-문화역서를 284', '식당-가격대-비싼', '식당-지역-서울 중랑', '식당-종류-한식당', '식당-야외석 유무=yes', '식당-예약 요일-토요일', '식당-예약 명수=1', '식당-이름-외계인의맛집'])

```

## 2.3 Slot Meta

```
import json

data = json.load(open('data/train_data.json'))
slot_meta = build_slot_meta(data)

print(len(slot_meta)) # 45

for slot in slot_meta:
    print(slot)
```

관광-경치 좋은  
관광-교육적  
관광-도보 가능  
관광-문화 예술  
관광-역사적  
관광-이름  
관광-종류  
관광-주차 가능  
관광-지역  
숙소-가격대  
숙소-도보 가능  
숙소-수영장 유무  
숙소-스파 유무  
숙소-예약 기간  
숙소-예약 명수  
숙소-예약 요일  
숙소-이름  
숙소-인터넷 가능

...

$$B_t = \left\{ \left( S^j, V_t^j \right) \mid 1 \leq j \leq J \right\}$$

**Slot Meta:**  $J$ 개의 미리 시나리오에 의해 정의된 **List of trackable slots**

## 2.4 DSTEvaluator

```
class DSTEvaluator:
    def __init__(self, slot_meta):
        self.slot_meta = slot_meta
        self.init()

    def init(self):
        self.joint_goal_hit = 0
        self.all_hit = 0
        self.slot_turn_acc = 0
        self.slot_F1_pred = 0
        self.slot_F1_count = 0

    def update(self, gold, pred):
        self.all_hit += 1
        if set(pred) == set(gold):
            self.joint_goal_hit += 1

        temp_acc = compute_acc(gold, pred, self.slot_meta)
        self.slot_turn_acc += temp_acc

        temp_f1, _, _, count = compute_prf(gold, pred)
        self.slot_F1_pred += temp_f1
        self.slot_F1_count += count

    def compute(self):
        turn_acc_score = self.slot_turn_acc / self.all_hit
        slot_F1_score = self.slot_F1_pred / self.slot_F1_count
        joint_goal_accuracy = self.joint_goal_hit / self.all_hit
        eval_result = {
            'joint_goal_accuracy': joint_goal_accuracy,
            'turn_slot_accuracy': turn_acc_score,
            'turn_slot_f1': slot_F1_score
        }
    return eval_result
```

### Usage

```
from eval_utils import DSTEvaluator

evaluator = DSTEvaluator(slot_meta)
```

```
for (pred, gold) in zip(preds, golds):
    evaluator.update(gold, pred)
evaluator.compute()
```

## 2.4 DSTEvaluator

J의 갯수가 4라고 가정할 때 (Slot A, B, C, D)

	Slot A	Slot B	Slot C	Slot D
Pred	a	b	c	d
Ground Truth	a	b	c	d

```
from eval_utils import DSTEvaluator

slot_meta = ['Slot-A', 'Slot-B', 'Slot-C', 'Slot-D']
evaluator = DSTEvaluator(slot_meta)

pred = ['Slot-A-a', 'Slot-B-b', 'Slot-C-c', 'Slot-D-d']
gold = ['Slot-A-a', 'Slot-B-b', 'Slot-C-c', 'Slot-D-d'] # (DOMAIN-SLOT-VALUE)

evaluator.update(pred, gold)
evaluator.compute()

{'joint_goal_accuracy': 1.0, 'turn_slot_accuracy': 1.0, 'turn_slot_f1': 1.0}
```



## 2.4 DSTEvaluator

J의 갯수가 4라고 가정할 때 (Slot A, B, C, D)

	Slot A	Slot B	Slot C	Slot D
Pred	b	b	c	d
Ground Truth	a	b	c	d

```
from eval_utils import DSTEvaluator

slot_meta = ['Slot-A', 'Slot-B', 'Slot-C', 'Slot-D']
evaluator = DSTEvaluator(slot_meta)

pred = ['Slot-A-b', 'Slot-B-b', 'Slot-C-c', 'Slot-D-d']
gold = ['Slot-A-a', 'Slot-B-b', 'Slot-C-c', 'Slot-D-d'] # (DOMAIN-SLOT-VALUE)

evaluator.update(pred, gold)
evaluator.compute()

{'joint_goal_accuracy': 0.0, 'turn_slot_accuracy': 0.75, 'turn_slot_f1': 0.75}
```

## 2.4 DSTEvaluator

J의 갯수가 4라고 가정할 때 (Slot A, B, C, D)

	Slot A	Slot B	Slot C	Slot D
Pred	b	b	d	d
Ground Truth	a	b	c	d

```
from eval_utils import DSTEvaluator

slot_meta = ['Slot-A', 'Slot-B', 'Slot-C', 'Slot-D']
evaluator = DSTEvaluator(slot_meta)

pred = ['Slot-A-b', 'Slot-B-b', 'Slot-C-d', 'Slot-D-d']
gold = ['Slot-A-a', 'Slot-B-b', 'Slot-C-c', 'Slot-D-d'] # (DOMAIN-SLOT-VALUE)

evaluator.update(pred, gold)
evaluator.compute()

{'joint_goal_accuracy': 0.0, 'turn_slot_accuracy': 0.5, 'turn_slot_f1': 0.5}
```

## 2.4 DSTEvaluator

### Joint Goal Accuracy

pred =

['식당-가격대-적당',  
'식당-지역-서울 북쪽',  
'식당-종류-dontcare',  
'식당-예약 요일-목요일',  
'식당-예약 시간-19:30',  
'식당-예약 명수-1',  
'식당-이름-치키웨이']

gold =

['식당-가격대-적당',  
'식당-지역-서울 북쪽',  
'식당-종류-dontcare',  
'식당-예약 요일-목요일',  
'식당-예약 시간-19:30',  
'식당-예약 명수-1',  
'식당-이름-치키웨이']

State set이 정확히 일치해야 1점,  
하나라도 틀리면 0점

```
def update(self, gold, pred):  
    self.all_hit += 1  
    if set(pred) == set(gold):  
        self.joint_goal_hit += 1
```

```
evaluator.init()  
evaluator.update(gold, pred)  
result = evaluator.compute()  
print(result)
```

```
{'joint_goal_accuracy': 1.0, 'turn_slot_accuracy': 1.0, 'turn_slot_f1': 1.0}
```

## 2.4 DSTEvaluator

### Slot Accuracy

pred =

['식당-지역-서울 북쪽',  
'식당-종류-dontcare',  
'식당-예약 요일-목요일',  
'식당-예약 시간-19:30',  
'식당-예약 명수-1',  
'식당-이름-치키웨이']

gold =

['식당-가격대-적당',  
'식당-지역-서울 북쪽',  
'식당-종류-dontcare',  
'식당-예약 요일-목요일',  
'식당-예약 시간-19:30',  
'식당-예약 명수-1',  
'식당-이름-치키웨이']

이 경우 "식당-가격대"의 value는 "none"으로 취급  
(즉 value의 예측값이 "none"인 경우,  
리스트에 생략되어 있지만 **실제론 J개의 Slot들의  
value가 모두 평가되고 있는 상태**)

```
def compute_acc(gold, pred, slot_meta):  
    miss_gold = 0  
    miss_slot = []  
    for g in gold:  
        if g not in pred:  
            miss_gold += 1  
            miss_slot.append(g.rsplit("-", 1)[0])  
    wrong_pred = 0  
    for p in pred:  
        if p not in gold and p.rsplit("-", 1)[0] not in miss_slot:  
            wrong_pred += 1  
    ACC_TOTAL = len(slot_meta)  
    ACC = len(slot_meta) - miss_gold - wrong_pred  
    ACC = ACC / float(ACC_TOTAL)  
    return ACC
```

```
evaluator.init()  
evaluator.update(gold, pred)  
result = evaluator.compute()  
print(result)
```

```
{'joint_goal_accuracy': 0.0, 'turn_slot_accuracy': 0.9777777777777777, 'turn_slot_f1': 0.923076923076923}
```

## 2.4 DSTEvaluator

### Slot Accuracy

pred =

['식당-주차 가능-yes',  
'식당-가격대-적당',  
'식당-지역-서울 북쪽',  
'식당-종류-dontcare',  
'식당-예약 요일-목요일',  
'식당-예약 시간-19:30',  
'식당-예약 명수-1',  
'식당-이름-치키웨이']

gold =

['식당-가격대-적당',  
'식당-지역-서울 북쪽',  
'식당-종류-dontcare',  
'식당-예약 요일-목요일',  
'식당-예약 시간-19:30',  
'식당-예약 명수-1',  
'식당-이름-치키웨이']

```
def compute_acc(gold, pred, slot_meta):  
    miss_gold = 0  
    miss_slot = []  
    for g in gold:  
        if g not in pred:  
            miss_gold += 1  
            miss_slot.append(g.rsplit("-", 1)[0])  
    wrong_pred = 0  
    for p in pred:  
        if p not in gold and p.rsplit("-", 1)[0] not in miss_slot:  
            wrong_pred += 1  
    ACC_TOTAL = len(slot_meta)  
    ACC = len(slot_meta) - miss_gold - wrong_pred  
    ACC = ACC / float(ACC_TOTAL)  
    return ACC
```

```
evaluator.init()  
evaluator.update(gold, pred)  
result = evaluator.compute()  
print(result)
```

```
{'joint_goal_accuracy': 0.0, 'turn_slot_accuracy': 0.9777777777777777, 'turn_slot_f1': 0.923076923076923}
```

# 3.

## Taxonomy of DST models

DST model을 구현하는 두 가지 방법에 대해 소개합니다.

## 3.1 DST Recap

$$\prod_{t=1}^T \prod_{i=1}^J p\left(V_t^j \mid D_t, S^j\right)$$

$$D_t = (r_1, u_1, r_2, u_2, \dots, r_t, u_t)$$

**Input: t-번째 턴까지의 Dialogue Context**

(한 턴은 각각 system response r과 user utterance u로 이루어져 있다.)

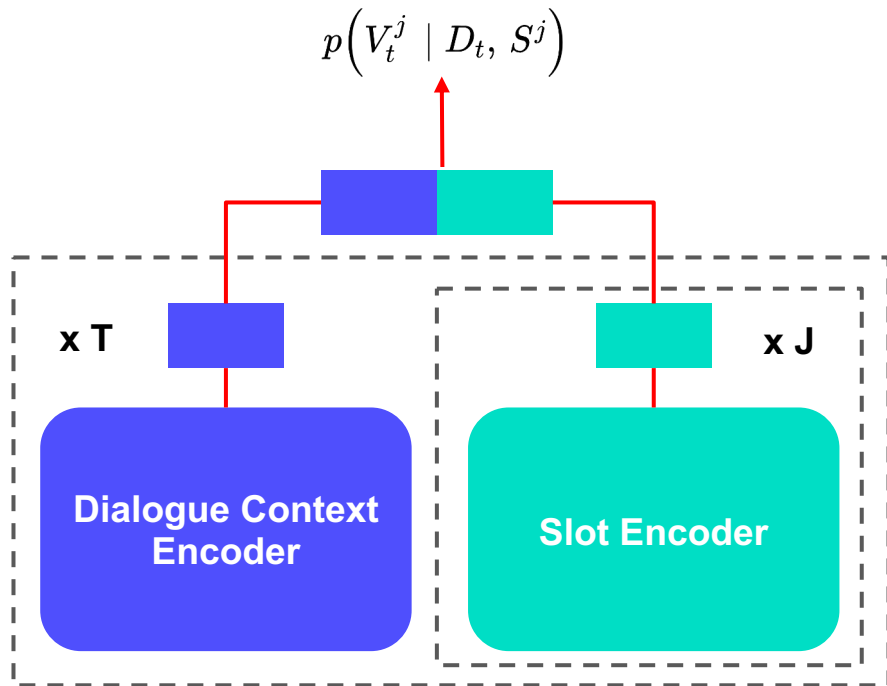
+ a

$$B_t = \left\{ \left( S^j, V_t^j \right) \mid 1 \leq j \leq J \right\}$$

**Output: t-번째 턴까지 추정되는 Dialogue State**

## 3.2 Model Concept

$$\prod_{t=1}^T \prod_{i=1}^J p(V_t^j | D_t, S^j)$$





## 3.2 Model Concept

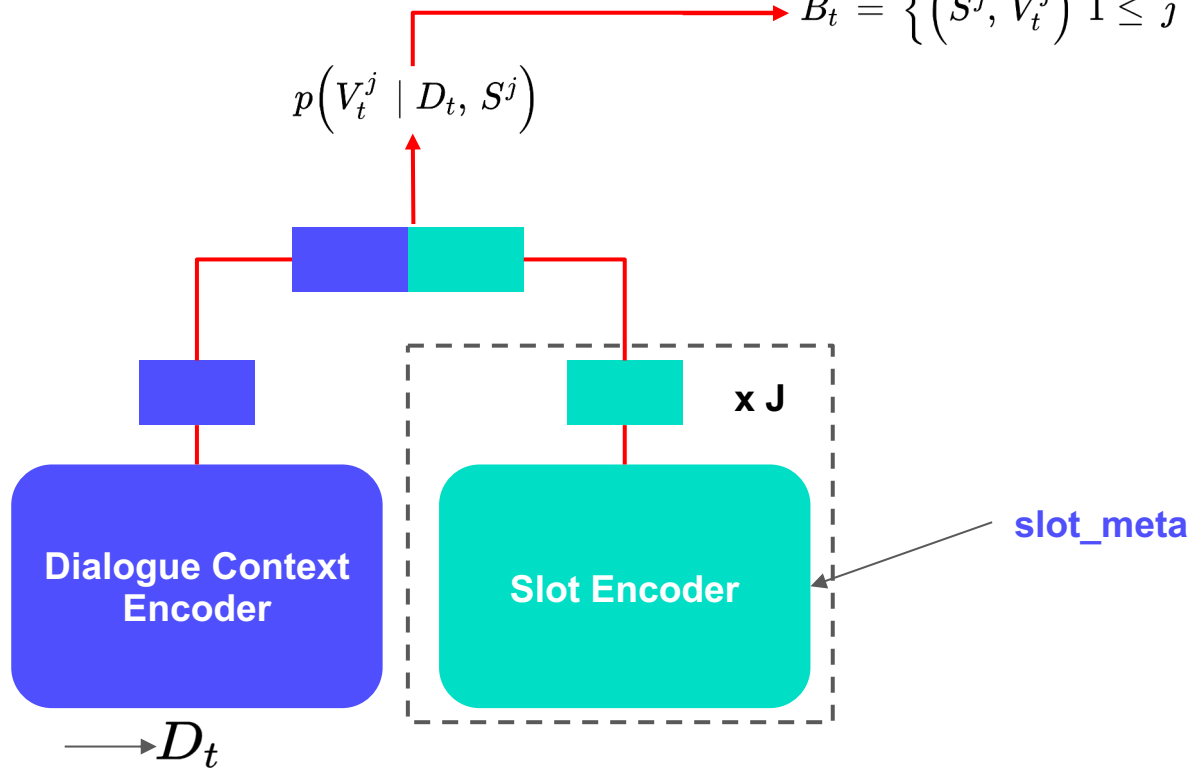
$$\prod_{i=1}^J p(V_t^j \mid D_t, S^j)$$

recover\_state (Predicted B\_t)

Evaluation level

$$B_t = \left\{ (S^j, V_t^j) \mid 1 \leq j \leq J \right\}$$

$$p(V_t^j \mid D_t, S^j)$$



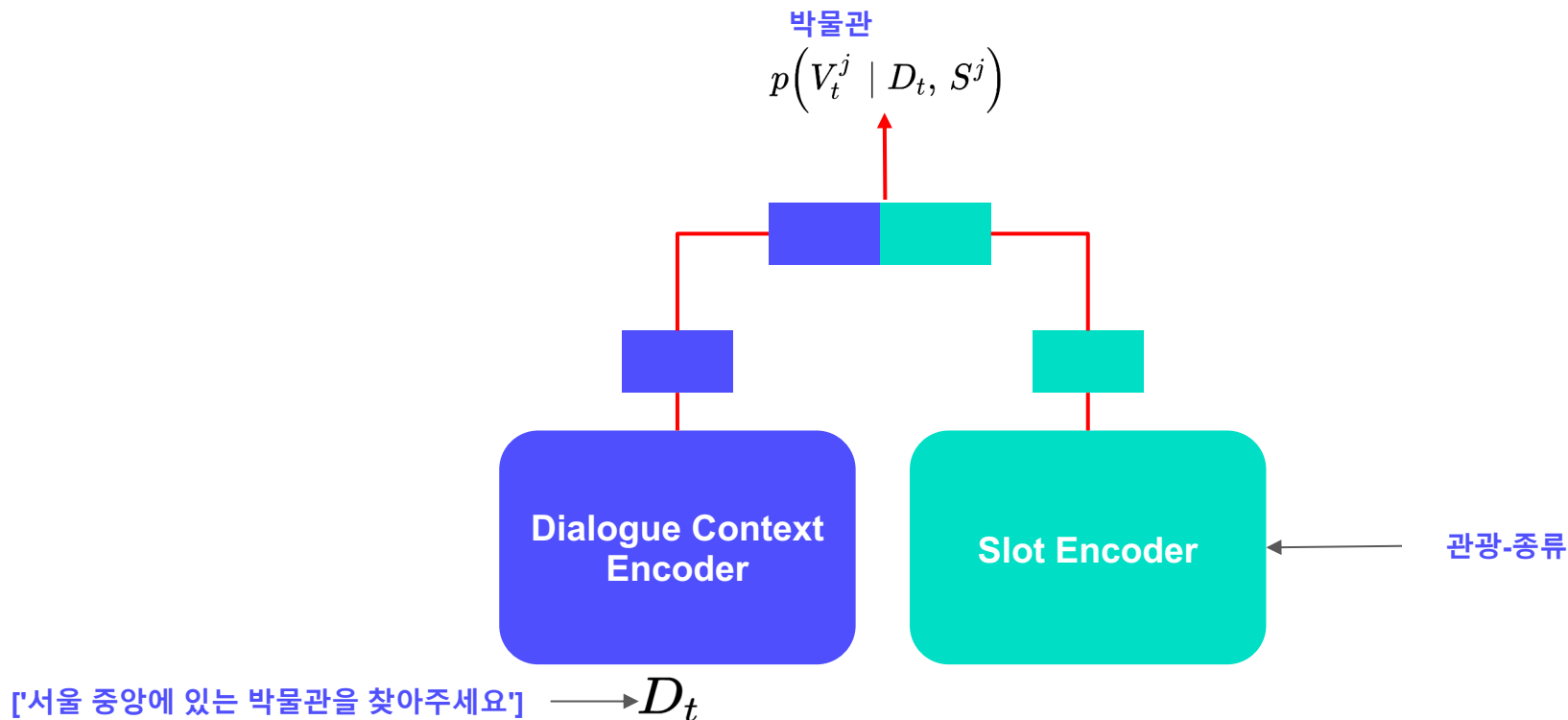
example.context\_turns +  
example.current\_turn

$\longrightarrow D_t$

## 3.2 Model Concept

$$p(V_t^j | D_t, S^j)$$

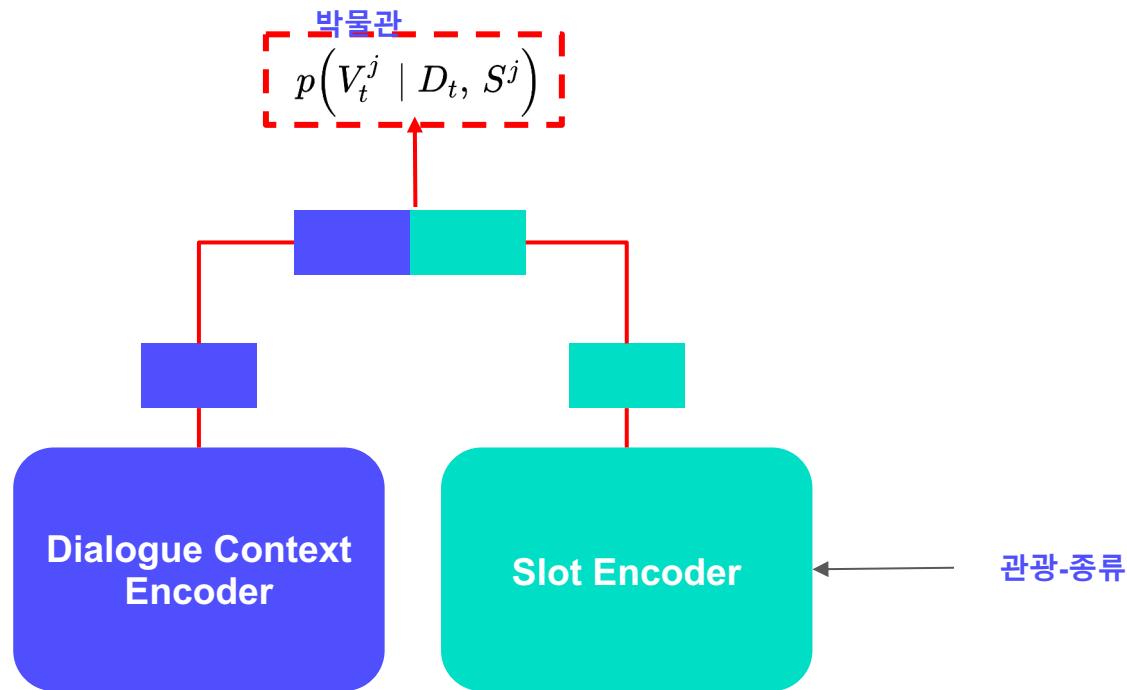
DSTInputExample(guid='snowy-hat-8324:관광\_식당\_11-0', context\_turns=[], current\_turn=[', '서울 중앙에 있는 박물관을 찾아주세요'], label=['관광-종류-박물관', '관광-지역-서울 중앙'])



## 3.2 Model Concept

$$p(V_t^j | D_t, S^j)$$

DSTInputExample(guid='snowy-hat-8324:관광\_식당\_11-0', context\_turns=[], current\_turn=[', '서울 중앙에 있는 박물관을 찾아주세요'], label=['관광-종류-박물관', '관광-지역-서울 중앙'])



**['서울 중앙에 있는 박물관을 찾아주세요']**  $\longrightarrow D_t$

### 3.3 How to represent $p(V)$ ?

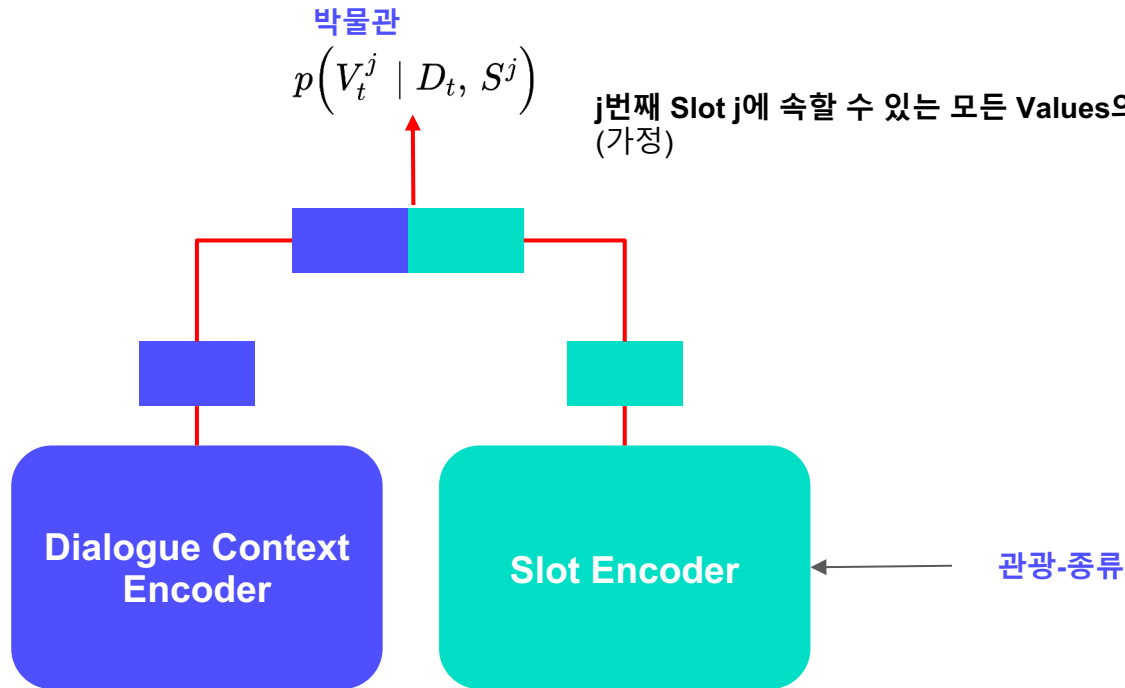
#### Classification

$$V^j \in \text{Set}(S^j) = \{V^{ji} \mid 1 \leq i \leq |\text{Set}(S^j)|\}$$

박물관

$$p(V_t^j \mid D_t, S^j)$$

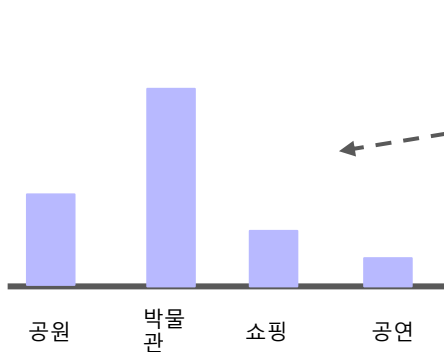
j번째 Slot j에 속할 수 있는 모든 Values의 집합을 정의  
(가정)



[서울 중앙에 있는 박물관을 찾아주세요]  $\longrightarrow D_t$

### 3.3 How to represent p(V)?

#### Classification



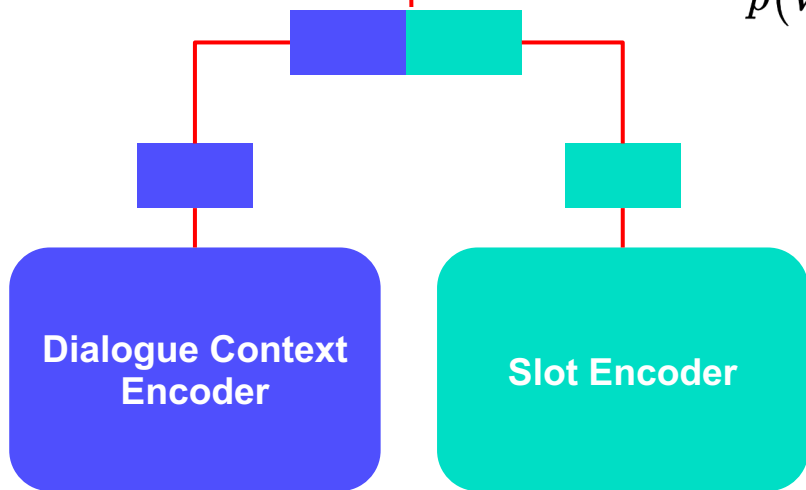
$$V^j \in \text{Set}(S^j) = \{V^{ji} \mid 1 \leq i \leq |\text{Set}(S^j)|\}$$

$$C_j = \text{Set}(S^j)$$

$$p(V^{ji}) = \frac{\exp(\text{score}(V^{ji}))}{\sum_{k=1}^{|C_j|} \exp(\text{score}(V^{jk}))}$$

박물관

$$p(V_t^j \mid D_t, S^j)$$

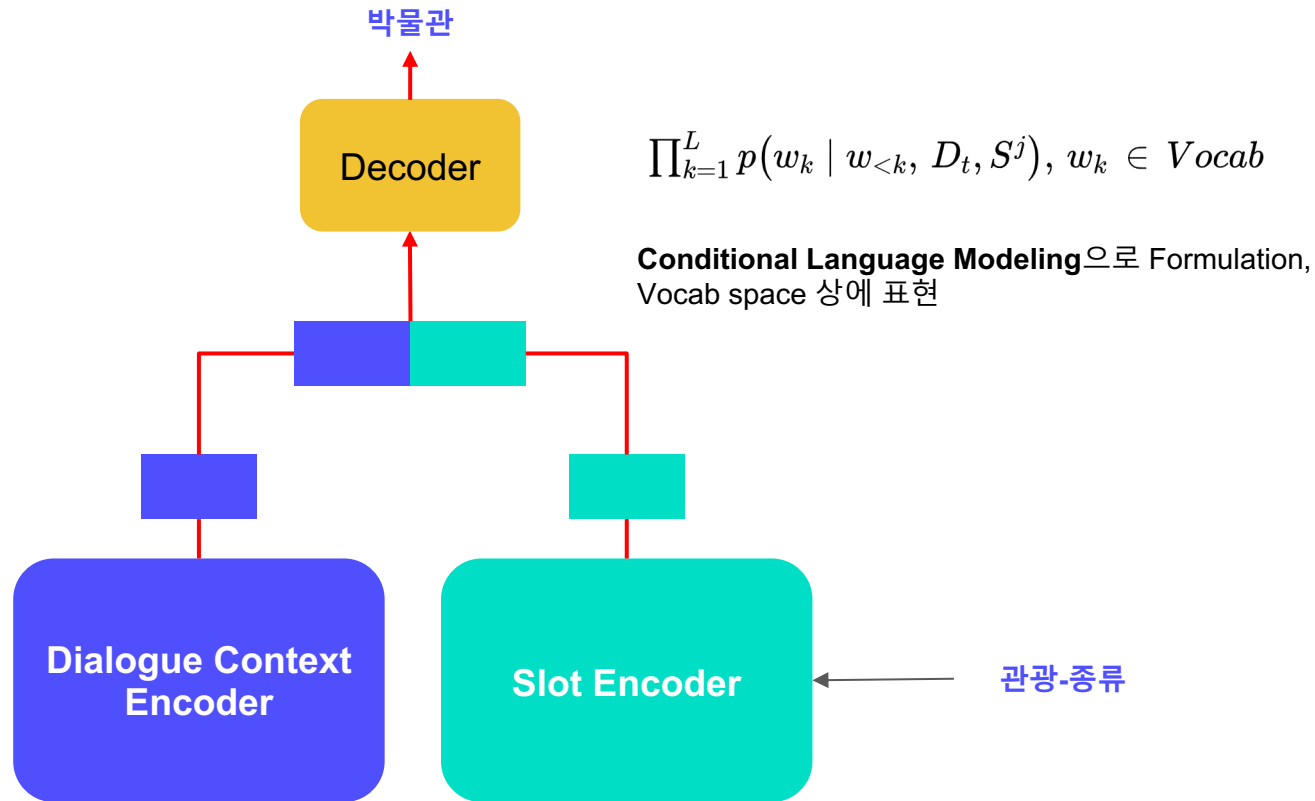


관광-종류

['서울 중앙에 있는 박물관을 찾아주세요']  $\longrightarrow D_t$

### 3.3 How to represent p(V)?

Generation



[서울 중앙에 있는 박물관을 찾아주세요]  $\longrightarrow D_t$

## 3.4 Taxonomy of DST models

---

Taxonomy of DST models

**Ontology-based DST  
(Classification)**

**Open-vocab based DST  
(Generation / Extraction)**

**End of Document**  
**Thank You.**