



# AI 챗봇 콘텐츠 제작 과정

---

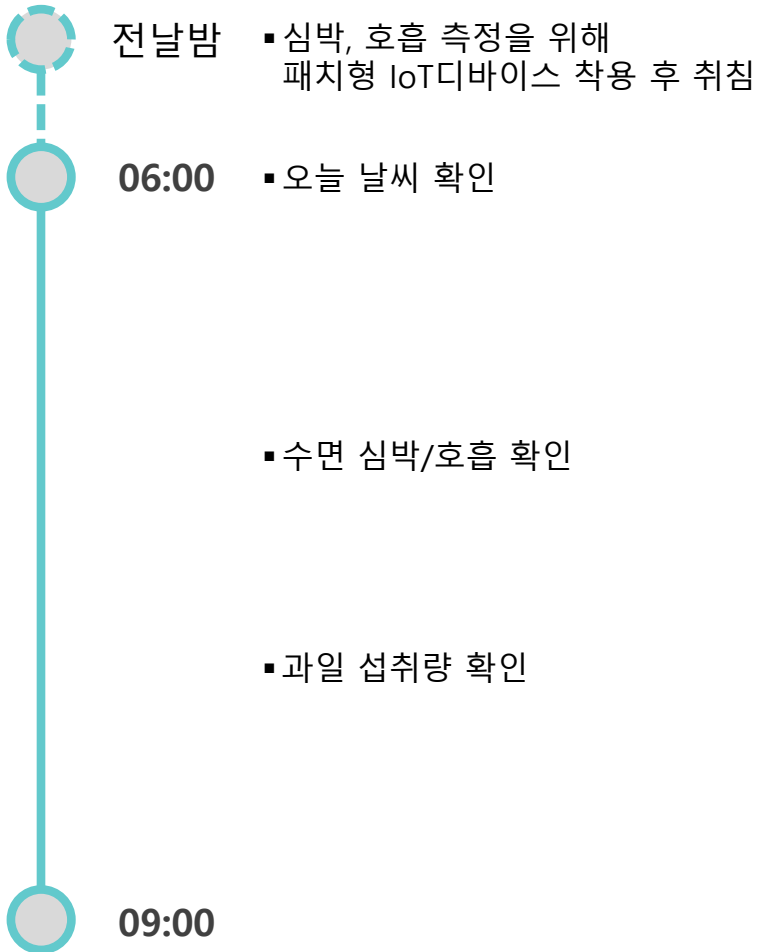
*Our Focus, AI Platform*  
*Our Next Step, AI Portal*



# SDS 설명

# 1. 챗봇 만들기

## 세부 시나리오 설계 - 대화 예시



오늘 날씨가 어때?

오늘 서울 날씨는 기온 7도, 습도는 35%입니다.

오늘 서울 미세먼지 농도는 주의 단계입니다. 가급적 외출을 자제하시고, 외출시에는 마스크를 꼭 챙기세요.

어제 수면패턴 괜찮았어?

무호흡 또는 부정맥은 측정되지 않았고 정상적인 수면패턴을 보였습니다.

과일을 먹어도 되나?

많이 드시면 혈당을 올려서 좋지 않습니다. 하지만, 적절한 양을 드시면 비타민, 무기질, 섬유소를 섭취할 수 있습니다. 사과 삼분의일개, 참외 반개, 오렌지 반개, 수박 한쪽, 배 사분의일개, 딸기 열개, 과일주스 반컵 정도가 적당합니다.

## 2. Context 유지 가능한 챗봇 만들기

위의 작업까지는 챗봇, FAQ로도 가능하다.

하지만..

- 복잡한 대화턴을 구현해야 하는 issue
- 키워드 context 유지 issue



## 2. Context 유지 가능한 챗봇 만들기

### Context?

- 맥락, 문맥, 전후사정

### 대화에 맥락이 없다면?



## 2. Context 유지 가능한 챗봇 만들기

### 해결 방법 : tagging을 통한 시나리오 전개

- Tagging이란?

콘텐츠의 내용을 대표할 수 있는 검색용 꼬리표인 키워드 또는 태그를 다는 것 .

글을 올린 사람이나 사이트 관리자가 글이나 이미지를 관련된 주제나 카테고리의 형태로 분류될 수 있도록 키워드 처리를 해 주는 것

- Tagging의 종류(SDS)

- 1) SLOT 태깅

- 2) DA 태깅



## 2. Context 유지 가능한 챗봇 만들기

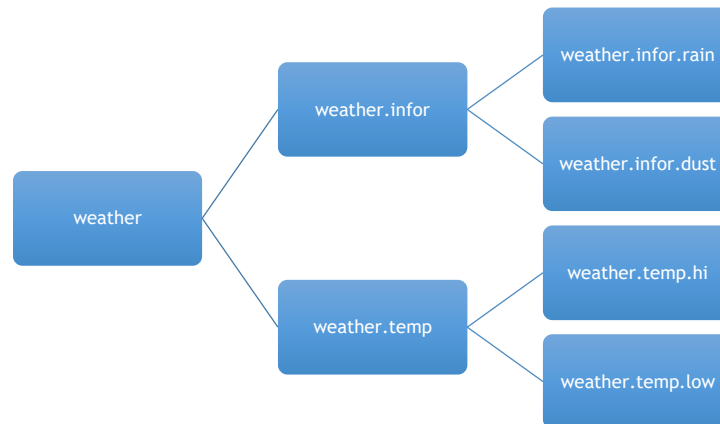
- SLOT의 정의

사용자와 시스템 간 대화를 진행하여 도메인에 주어진 목적을 해결하기 위해 공유해야 할 개념 정보

- CLASS 의 정의

슬롯의 그룹 - 관련 슬롯들의 집합

Ex) weather 라는 클래스 안에 다양한 슬롯들을 형성



## 2. Context 유지 가능한 챗봇 만들기

### 세부 시나리오 설계 - 세부 대화 설계(multi-turn, slot)

카테고리		Question Answer	내 용
계좌 변경	1	Q	안녕하세요? 마인즈 은행입니다. 무엇을 도와 드릴까요?
		A	계좌 변경 하고 싶어요
	2	Q	본 서비스에 앞서 고객님의 고객 정보가 필요 합니다. 고객님의 이름을 입력해 주시기 바랍니다.
		A	홍길동입니다
	3	Q	고객님 성함이 홍길동이 맞으신가요?
		A	네.
	4	Q	홍길동 고객님, 결제계좌 변경 서비스를 안내해 드리겠습니다. 고객님의 기존 계좌가 마인즈은행이신가요?
		A	아니요.
	5	Q	변경하려는 계좌의 은행이 어디신가요?
		A	신한은행입니다.
	6	Q	변경하시려는 계좌가 신한은행이 맞습니까?
		A	네

Multi-turn

Slots!!



## 2. Context 유지 가능한 챗봇 만들기

- DA(Dialog-Act) Type의 정의

사용자 혹은 시스템의 발화 의도의 뜻을 명확히 하는 행위(대화 행위의 종류)

Ex) SDS에 정의된 **기본 DA**

- hello : 만남 인사
- bye : 헤어짐 인사
- affirm : 긍정("예")
- negate : 부정("아니오")
- inform : 정보 제공 의도
- request : 정보에 대한 질문
- confirm : 확인 의도 예/아니오



### 3. SDS 대화 모델링

#### • 지식스크립트 함수 모음

#### 대화가 시나리오 대로 진행되기 위한 조건식

함수 이름	설명	예제
IsDAType('발화자','DA')	현재까지의 대화에서 "발화자"가 "DA type"으로 발화한 적이 있는가를 검사 IsDAType('user','negate')==true	S : 네, 결제계좌 변경 안내드리겠습니다. 고객님의 기존 계좌가 KEB하나은행이신가요? # inform() U : 아니오 # negate() S : 변경하려는 계좌의 은행은 어디신가요? U : 저는 우리은행을 사용합니다 결과) IsDAType('user','negate')==true → 현재까지의 대화에서 negate가 발화되었기 때문에 true 리턴
IsDATypeAtPreviousUtter('발화자','DA')	바로 전 대화에서 "발화자"가 "DA type"으로 대화의 도를 수행했는가를 검사하여 맞으면 true, 아니면 false를 리턴 IsDATypeAtPreviousUtter('user','affirm') == true → true or false	S : 네, 결제계좌 변경 안내드리겠습니다. 고객님의 기존 계좌가 KEB하나은행이신가요? U : 네 # affirm() 결과) IsDATypeAtPreviousUtter('user','affirm')==true → 바로 이전 대화에서 affirm이 발화되었기 때문에 true 리턴
IsDATypeAtUtterHistory("발화자","DA","발화시점")	IsDATypeAtPreviousUtter()함수와 유사하다. 단, "발화시점"이 변수로 추가됨. "발화시점"은 현재 검사 시점보다 얼마큼 앞의 대화를 검색할 것인가를 표시. IsDATypeAtUtterHistory("user","affirm","1")==true은, IsDATypeAtPreviousUtter('user','affirm') == true와 동일	S : 네, 결제계좌 변경 안내드리겠습니다. 고객님의 기존 계좌가 KEB하나은행이신가요? # inform() U : 네 # affirm() 결과) IsDATypeAtUtterHistory("system","inform","2")==true → 발화시점을 변수의 값 만큼 거슬러 올라가 조건식과 비교하였을 때 true이기 때문에 true 리턴

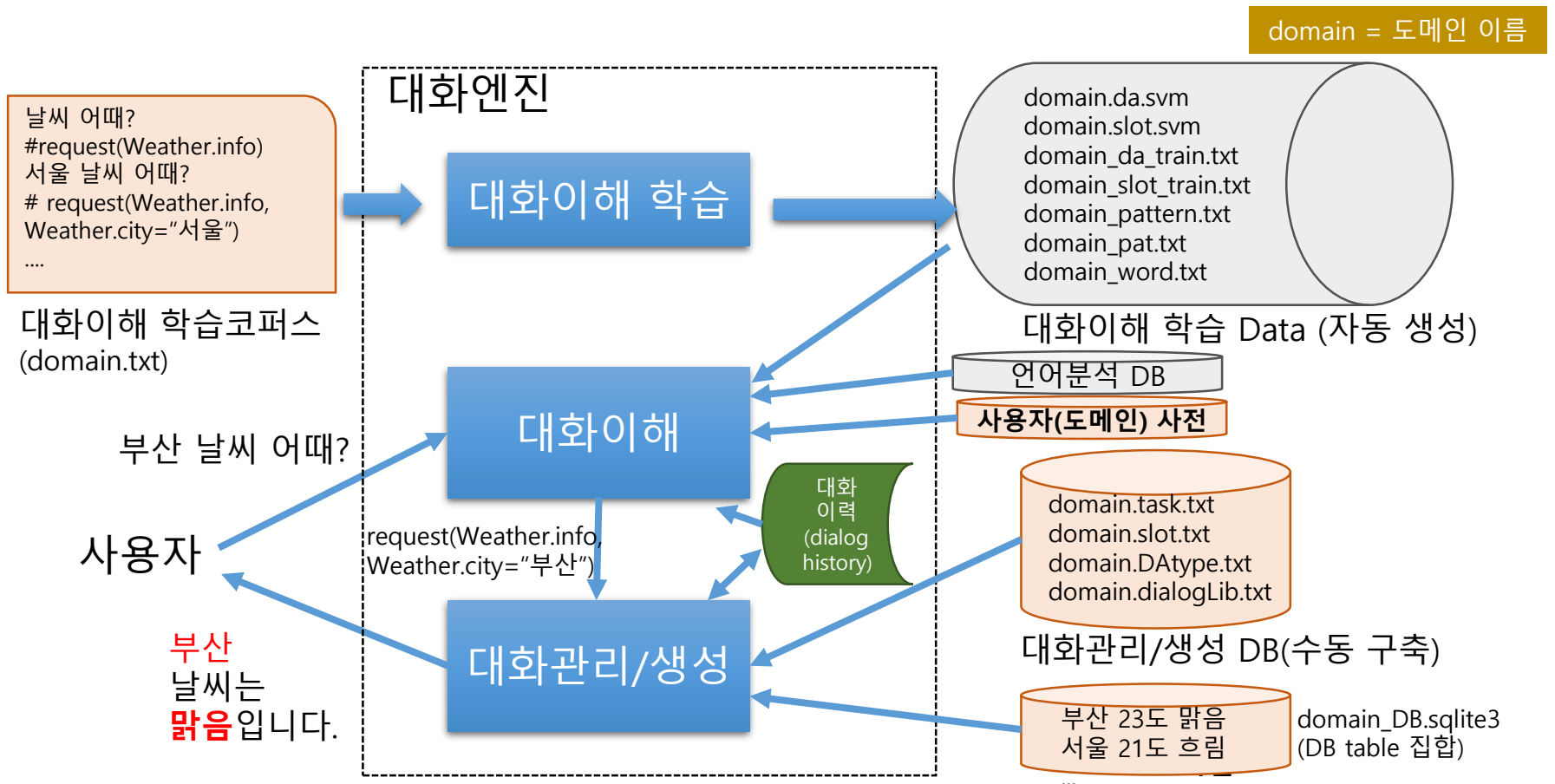
### 3. SDS 대화 모델링

#### • 지식스크립트 함수 모음

함수 이름	설명	예제
ExistValue('슬롯명')	"슬롯이름"에 해당하는 슬롯의 값이 현재 대화흐름에서 나타났으면 true를 아니면 false를 리턴 ExistValue('user_bank') == true → true or false	S : 변경하려는 계좌의 은행은 어디신가요? U : 저는 우리은행을 사용합니다. #inform(user_bank="우리은행") 결과) ExistValue('user_bank') == true → 대화에서 user_bank슬롯의 값이 나타났기 때문에 true
HasValue('슬롯명', '슬롯값')	문맥 상황(실제대화 + 도메인 DB정보)고려하여, "슬롯이름"의 slot이 현재 "슬롯값"을 가지는가를 검사하여 참이면 true 거짓이면 false를 리턴 HasValue('user_bank','우리은행') == true	S : 변경하려는 계좌의 은행은 어디신가요? U : 저는 우리은행을 사용합니다. #inform(user_bank="우리은행") 결과) HasValue('user_bank','신한은행') == true → 대화에서 user_bank슬롯의 값이 우리은행이기 때문에 false  HasValue('user_bank','우리은행') == true → 대화에서 user_bank슬롯의 값이 우리은행과 같기 때문에 true  HasValue('user_bank','신한은행') == false → 대화에서 user_bank슬롯의 값이 조건식과 일치하기 때문에 true

### 3. SDS 대화 모델링

#### • SDS(Spoken Dialog System) - 개념도



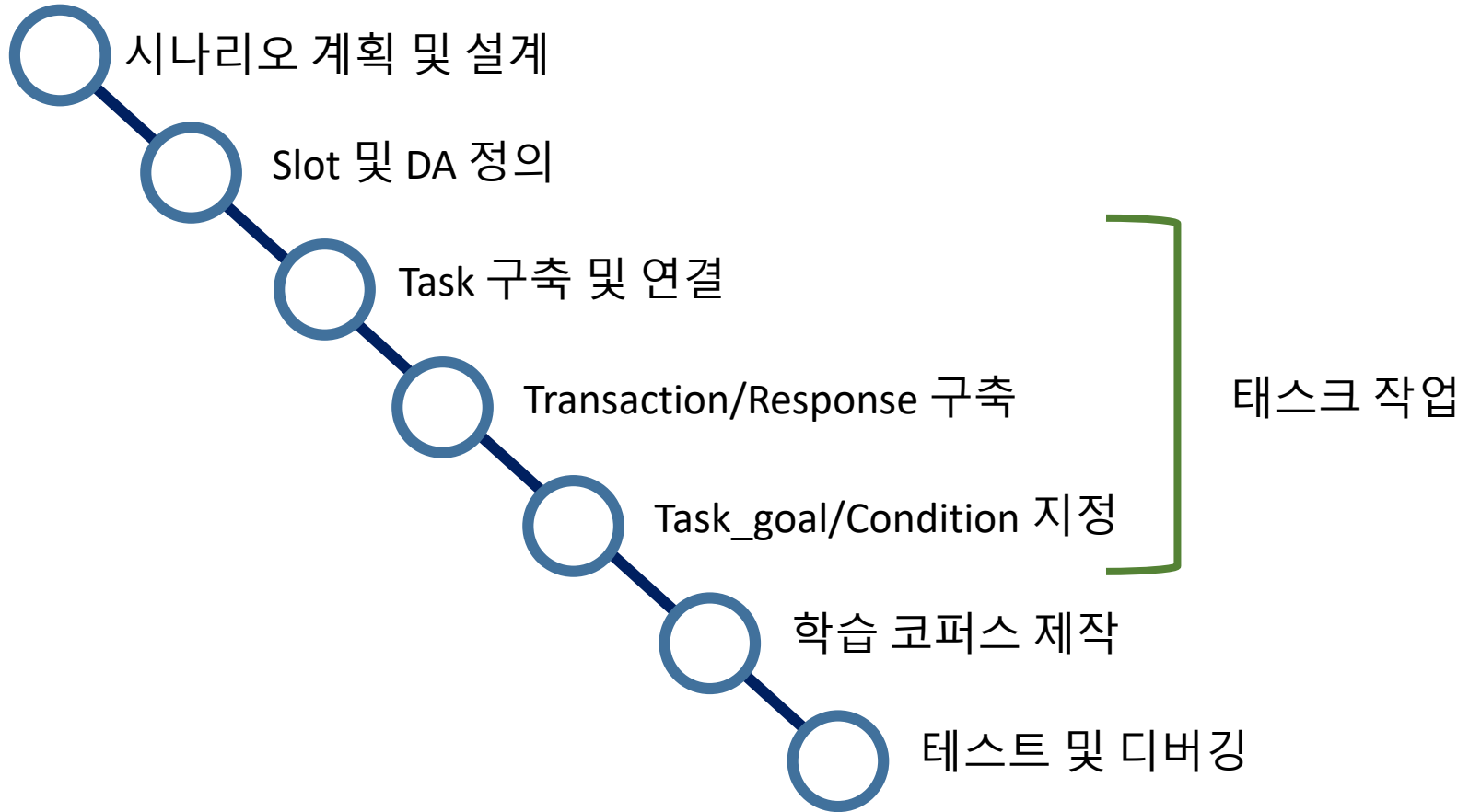
### 3. SDS 대화 모델링

#### • 학습 코퍼스 구축

학습을 위한 말주머니로, 컴퓨터가 사람의 대화를 이해하기 위해 학습.

	A	B	C	D	E	F	G	H	I	J
165	human	123-486547-4890		<user.account=123-486547-489-0>			# inform(user.account=123-486547-489)	carvatar	OPEN	
166	human	123-486547-4892내		<user.account=123-486547-489-인대			# inform(user.account=123-486547-489)	carvatar	OPEN	
167	human	내 계좌는 123-486547-489		내 계좌는 <user.account=123-486547-489>			# inform(user.account=123-486547-489)	carvatar	OPEN	
168	human	계좌번호는 123-486547-4890		계좌번호는 <user.account=123-486547-489-0>			# inform(user.account=123-486547-489)	carvatar	OPEN	
169	human	아뇨		아뇨			# negate()	carvatar	OPEN	
170	human	아니		아니			# negate()	carvatar	OPEN	
171	human	결제 계좌 변경 해줘		결제 계좌 변경 해줘			# change_payment_account()	carvatar	OPEN	
172	human	결제 계좌 변경 해줄래		결제 계좌 변경 해줄래			# change_payment_account()	carvatar	OPEN	
173	human	결제 계좌 변경 부탁할게		결제 계좌 변경 부탁할게			# change_payment_account()	carvatar	OPEN	
174	human	결제 계좌 변경 하고 싶어		결제 계좌 변경 하고 싶어			# change_payment_account()	carvatar	OPEN	
175	human	결제 계좌를 변경 할래		결제 계좌를 변경 할래			# change_payment_account()	carvatar	OPEN	
176	human	결제계좌를 바꾸고 싶어		결제계좌를 바꾸고 싶어			# change_payment_account()	carvatar	OPEN	
177	human	결제계좌 바꿔줄래		결제계좌 바꿔줄래			# change_payment_account()	carvatar	OPEN	
178	human	결제계좌 바꿔줘		결제계좌 바꿔줘			# change_payment_account()	carvatar	OPEN	
179	human	결제 계좌를 바꿀게		결제 계좌를 바꿀게			# change_payment_account()	carvatar	OPEN	
180	human	결제 계좌 변경 어떻게 하지		결제 계좌 변경 어떻게 하지			# change_payment_account()	carvatar	OPEN	
181	human	결제 계좌 어떻게 바꾸지		결제 계좌 어떻게 바꾸지			# change_payment_account()	carvatar	OPEN	
182	human	결제 계좌 어떻게 변경하니		결제 계좌 어떻게 변경하니			# change_payment_account()	carvatar	OPEN	
183	human	결제 계좌를 어떻게 변경하는지 방법 알려줘		결제 계좌를 어떻게 변경하는지 방법 알려줘			# change_payment_account()	carvatar	OPEN	
184	human	결제 계좌를 어떻게 하면 변경할 수 있는지 알려줘		결제 계좌를 어떻게 하면 변경할 수 있는지 알려줘			# change_payment_account()	carvatar	OPEN	
185	human	결제하는 계좌를 어떻게 변경하는지 방법 알려줘		결제하는 계좌를 어떻게 변경하는지 방법 알려줘			# change_payment_account()	carvatar	OPEN	
186	human	결제 계좌 변경 방법이 궁금해		결제 계좌 변경 방법이 궁금해			# change_payment_account()	carvatar	OPEN	
187	human	결제 계좌 변경 방법 좀 알려줄래		결제 계좌 변경 방법 좀 알려줄래			# change_payment_account()	carvatar	OPEN	
188	human	결제 계좌 변경 방법을 알려줘		결제 계좌 변경 방법을 알려줘			# change_payment_account()	carvatar	OPEN	
189	human	결제 계좌 변경이 하고 싶네		결제 계좌 변경이 하고 싶네			# change_payment_account()	carvatar	OPEN	
190	human	결제하는 계좌를 어떻게 하면 변경할 수 있는지 알려줘		결제하는 계좌를 어떻게 하면 변경할 수 있는지 알려줘			# change_payment_account()	carvatar	OPEN	
191	human	결제하는 계좌를 어떻게 하면 변경할 수 있는지 알려줘		결제하는 계좌를 어떻게 하면 변경할 수 있는지 알려줘			# change_payment_account()	carvatar	OPEN	
192	human	결제하는 계좌 변경 방법이 뭐야		결제하는 계좌 변경 방법이 뭐야			# change_payment_account()	carvatar	OPEN	
193	human	결제하는 계좌 변경 방법이 궁금해		결제하는 계좌 변경 방법이 궁금해			# change_payment_account()	carvatar	OPEN	
194	human	결제되는 계좌 변경 방법 좀 알려줄래		결제되는 계좌 변경 방법 좀 알려줄래			# change_payment_account()	carvatar	OPEN	
195	human	결제되는 계좌 변경 방법을 알려줘		결제되는 계좌 변경 방법을 알려줘			# change_payment_account()	carvatar	OPEN	
196	human	결제되는 계좌 변경이 하고 싶네		결제되는 계좌 변경이 하고 싶네			# change_payment_account()	carvatar	OPEN	
197	human	결제하고 있는 통장을 바꾸고 싶은데		결제하고 있는 통장을 바꾸고 싶은데			# change_payment_account()	carvatar	OPEN	
198	human	결제하고 있는 통장을 바꿔줄래		결제하고 있는 통장을 바꿔줄래			# change_payment_account()	carvatar	OPEN	
199	human	내 결제 계좌 변경 방법이 뭐야		내 결제 계좌 변경 방법이 뭐야			# change_payment_account()	carvatar	OPEN	
200	human	내 결제 계좌 변경 방법이 궁금해		내 결제 계좌 변경 방법이 궁금해			# change_payment_account()	carvatar	OPEN	
201	human	내 결제 계좌 변경 방법 좀 알려줄래		내 결제 계좌 변경 방법 좀 알려줄래			# change_payment_account()	carvatar	OPEN	
202	human	내 결제 계좌 변경 방법을 알려줘		내 결제 계좌 변경 방법을 알려줘			# change_payment_account()	carvatar	OPEN	
203	human	내 결제 계좌 변경이 하고 싶네		내 결제 계좌 변경이 하고 싶네			# change_payment_account()	carvatar	OPEN	
204	human	나의 결제하는 계좌를 어떻게 변경하는지 방법 알려줘		나의 결제하는 계좌를 어떻게 변경하는지 방법 알려줘			# change_payment_account()	carvatar	OPEN	
205	human	나의 결제하는 계좌를 어떻게 하면 변경할 수 있는지 알려줘		나의 결제하는 계좌를 어떻게 하면 변경할 수 있는지 알려줘			# change_payment_account()	carvatar	OPEN	
206	human	나의 결제하는 계좌 변경 방법이 뭐야		나의 결제하는 계좌 변경 방법이 뭐야			# change_payment_account()	carvatar	OPEN	
207	human	나의 결제하는 계좌 변경 방법이 궁금해		나의 결제하는 계좌 변경 방법이 궁금해			# change_payment_account()	carvatar	OPEN	

## 4. SDS 제작 방법



## 4. SDS 제작 방법

### 1. 시나리오 계획 및 설계

- 어떤 시나리오를 가진 챗봇을 만들지 구상하는 단계.
- 질문과 답변 사항에 대해서 구체적으로 시나리오 작업 필요.
- 예외 상황들에 대해서도 시나리오 구상 필요.
- 슬롯화 할 내용과 사용할 대화 의도 들에 대해서 구상하기

### 2. Slot 정의

- 대화를 이어가는데 있어서 꼭 필요한 내용
- 슬롯 만들기

### 3. DA(Dialog-Act) 정의

- 유저 혹은 시스템의 발화 의도
- 대화 의도를 만들기

## 4. SDS 제작 방법

### 4. Task 구축

- Task : 상황에 대해서 대화를 관리해 주는 역할
- 여러 개의 테스트를 구조화 하여서 대화의 흐름을 생성.
- 테스트 간의 관계를 설정이 중요.
- 테스트 이름 설정 후에 테스트 연결 해주기.

### 5. Transaction/Response 설정

- transaction : 테스트 진입시 맨 처음에 작동하게 되는 시스템 발화  
ex) 대화 시작시 설정된 greeting 메시지
- Response : 대화 상황 내에서 유저 발화에 따라 출력되는 시스템 발화를 설정 해주는 메뉴  
ex) System : 어서오세요. 마인즈랩 도우미 봇입니다.  
무엇을 도와드릴까요?  
User : 마인즈랩 위치는 어디야?  
System : 마인즈랩은 경기도 성남시 분당구 대왕판교로 644번길 49,  
다산타워 601호에 위치하고 있습니다.



## 4. SDS 제작 방법

### 6. Task\_goal 및 condition 지정

- 태스크의 종료 시점과 연결 조건을 조정하기 위한 과정.
- task\_goal : 태스크의 대화를 끝마치기 위한 조건
- condition : 다른 태스크와의 연결 조건

### 7. 학습코퍼스 제작

- 예상되는 사용자 발화들을 컴퓨터에 학습 시켜 주기 위한 과정.
- 규칙에 맞도록 코퍼스를 제작.

#### - 코퍼스 제작 규칙

- 1) 핵심어 파생어 위주로 코퍼스를 형성.
- 2) 문장의 여러 패턴을 적용해서 코퍼스 생성(패턴 중시)
- 3) 슬롯은 명사를 태깅.
- 4) 문장 내 슬롯의 위치 변화의 경우를 포함하여 생성
- 5) 어미를 최대한 다양한 형태로 넣어주기
- 6) 한 SDS 내에서 여러 의도로 갈라질 경우, 코퍼스 비중 맞춰주기.

## 4. SDS 제작 방법

### 8. 테스트 및 디버깅 작업

- 대화 프로젝트가 작동 되는지 여부 파악
- 테스트 이상 시, 로그 창을 열어서 원인 파악하기.
- 많이 일어나는 실수 :  
조건문 오타, 저장 시 오류(DA 저장, 테스트 조건 저장)

# SDS 실습

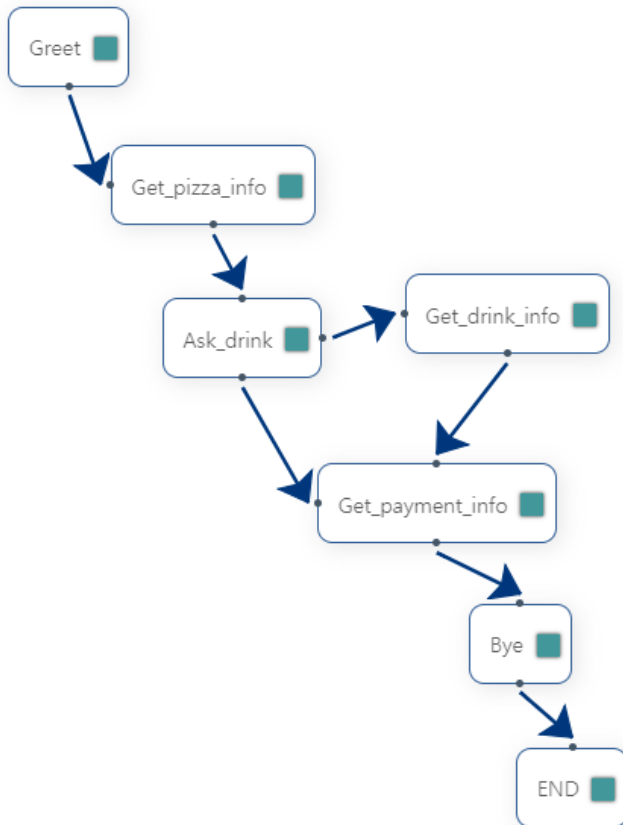
## SDS 실습 – 피자 주문 챗봇

고객으로부터 주문 받을 피자 정보를 입력 받고, 음료 주문 여부를 물어본 다음 그에 따라 주문 받을 음료 정보를 입력 받고, 결제 수단을 물어본 다음 주문을 접수함으로써 대화가 종료

### 대표 대화 시나리오

발화 주체	발화 내용	
System	안녕하세요, 함께 즐겨요 피자훗입니다. 주문하실 피자 종류, 크기, 도우를 말씀해주세요.	1. 피자 정보 입력 받기 (종류, 가격 등)
User	페퍼로니 피자 라지 사이즈로 주세요	
System	어떤 도우의 피자를 주문하시겠습니까? (오리지널, 나폴리, 썬)	
User	보통 도우로요	2. 음료 주문 받기 (종류, 가격 등)
System	페퍼로니 피자 라지 사이즈 오리지널 도우의 가격은 22900원 입니다.음료도 주문하시겠습니까?	
User	네	
System	주문하실 음료를 말씀해주세요. (콜라, 사이다)	3. 총 결제 금액 계산 및 결제 수단 입력 받기
User	사이다 얼마예요	
System	사이다의 가격은 1800원 입니다. 주문하실 음료를 말씀해주세요. (콜라, 사이다)	
User	사이다로 주세요	4. 결제 수단 입력 받고 대화 종료
System	음료 가격 1800원을 포함하여 총 가격은 24700원 입니다. 결제는 현금으로 하시겠어요, 카드로 하시겠어요?	
User	카드로 할게요	
System	30분 안에 배달됩니다. 카드를 미리 준비해주세요. 이용해 주셔서 감사합니다.	

### Task Graph



Task 이름	수행 내용	다음 Task	이동 조건
Greet	첫인사를 수행함.	Get_pizza_info	항상
Get_pizza_info	주문 받을 피자 정보(피자 종류, 크기, 도우)를 입력 받음. 추가적으로, 사용자의 발화에 따라 피자 종류와 가격을 알려줌.	Ask_drink	사용자가 DB에 있는 피자 종류, 크기, 가격을 모두 말했을 때
Ask_drink	음료 주문 여부를 물어봄	Get_drink_info	사용자가 음료 주문 물음에 대해 긍정적으로 대답했을 때
		Get_payment_info	사용자가 음료 주문 물음에 대해 부정적으로 대답했을 때
Get_drink_info	주문 받을 음료 정보(음료 종류)를 입력 받음. 추가적으로, 사용자의 발화에 따라 음료 종류와 가격을 알려줌.	Get_payment_info	사용자가 DB에 있는 음료 종류를 말하거나, 음료 주문을 취소할 때
Get_payment_info	총 주문 금액을 알려주고 결제 수단을 물어봄.	Bye	사용자가 결제 수단을 말했다고 했을 때
Bye	결제 수단을 확인하고 대화 종료.	END	시스템이 끝인사를 수행했을 때

# SDS 실습 – 피자 주문 챗봇

## 신규 도메인 생성

로그인 후 ①을 클릭하여 신규 도메인을 생성한 후, 도메인 이름과 설명을 입력하여 저장한다.

GENIEDIALOG Overview

선택된 도메인이 없습니다.

① Domain List

+ Create Domain

Task Graph

Entities

Intents

Manual

Users

Samples

Logs

한글 대화로 가능한 인터페이스를 만들어 보세요.

Quick Start 동영상 Quick Guide #1 : 간단한 대화 만들기

1. 먼저 Domain을 생성하세요.

2. Domain에서 하는 일을 Tasks로 나누어 보세요.

4. 내가 말한

5. 학습하시.

Domain Name

Pizza\_bot

③ Save

Description

피자 주문

대화의도 필터링 낮은 신뢰도로 인식되는 대화의도를 인식하지 못한 경우(unknown 의도)로 설정하세요.

0.35 일반적으로 0.3 ~ 0.4 이하는 인식된 의도의 가능성이 아닐 가능성이 높고, 0.8 이상은 매우 높아서 정의한 문장에 매우 유사한 경우입니다.

Create from other saved domains

선택하세요

Sample Domain으로 시작하기

LANGUAGE

한국어 ☒

English ☐

# SDS 실습 – 피자 주문 챗봇

## Greet Task

Create Task를 클릭한 후, Task 이름과 설명을 입력한다. Task 시작 시스템 발화를 추가하여 아래와 같이 입력해 준다. 그 후, Save를 눌러 저장해준다.

The screenshot shows the GENIEDIALOG interface for creating a task. The sidebar on the left lists various components: Pizza\_bot, Task Graph, Entities, Intents, Manual, Users, Samples, and Logs. The main area displays a 'Create Task' dialog with the following fields and actions:

- 1** Create Task button (top right)
- 2** Task Name: Greet
- 3** Description: 첫인사
- 4** Initial: ☒
- 5** Save button (top right)
- Task 시작 시, 시스템 발화 정의** section:
  - 3** + 추가 button
  - 4** Text input: 안녕하세요 함께 즐겨요, 피자훗입니다.

## SDS 실습 - 피자 주문 챗봇

### Get\_pizza\_info Task 생성

동일한 방법으로 Get\_pizza\_info 태스크를 생성한다.


새로 만들기

Task Name **1** Get\_pizza\_info

Description 주문 받을 피자 정보(피자 종류, 크기, 도우)를 입력 받음. 추가적으로, 사용자의 발화에 따라 피자 종류와 가격을 알려줌.


Initial ☐


Task 시작 시, 시스템 발화 정의 **+ 추가**

**2** 

Task 시작 시, 시스템 발화 정의 **+ 추가**

**시스템 발화** **시스템 발화 후 행동**

시스템 발화 조건 true 

**3** 시스템 intent inform() 

주문하실 피자 종류, 크기, 도우를 말씀해주세요.

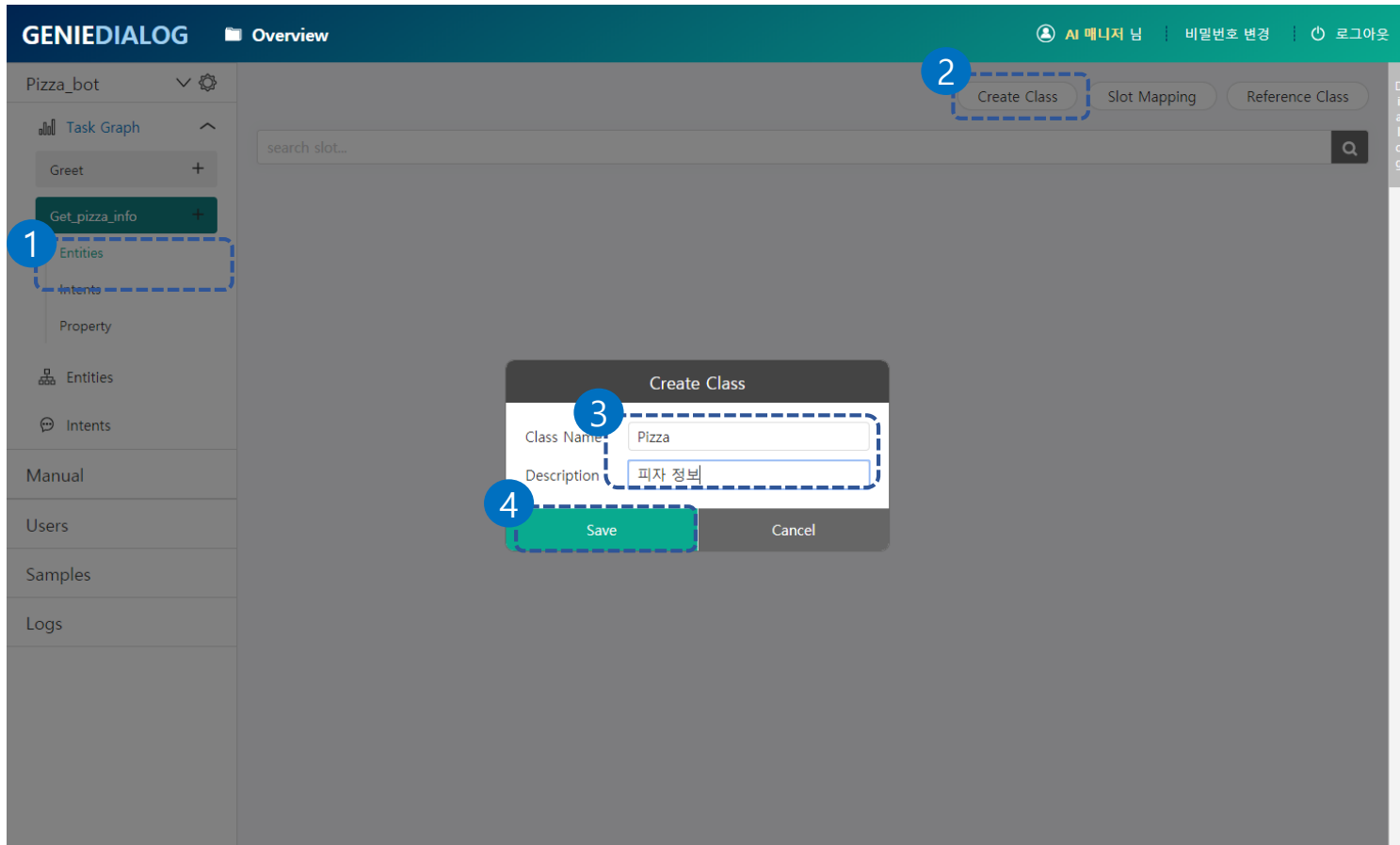
**+ Add**



## SDS 실습 – 피자 주문 챗봇

### Get\_pizza\_info Task – Entities 설정

Task 하위 메뉴의 “Entites” 메뉴로 이동 후 Create Class를 클릭하여 Class를 생성해준다.



# SDS 실습 – 피자 주문 챗봇

## Get\_pizza\_info Task – Entities 설정

생성한 Class 하위에 Create Slot을 통해 Entity를 생성해준다.

The screenshot shows the GENIEDIALOG interface for configuring a chatbot. On the left, a sidebar lists 'Pizza\_bot' with sub-items like 'Task Graph', 'Greet', 'Get\_pizza\_info', 'Entities', 'Intents', 'Property', 'Manual', 'Users', 'Samples', and 'Logs'. The main area displays the 'Get\_pizza\_info' task with a 'Pizza' entity. A 'Create Slot' dialog is open, allowing the user to define a slot for the 'Pizza' class. The dialog fields are: Class (Pizza), Slot Name (type), Slot Type (Sys.string), Default Value, and Slot description (피자 타입). A blue arrow points from the 'Slot Type' field to a detailed view of the 'Sys' type options on the right. The 'Sys' dropdown is open, showing options: person, place, date, time, None, Sys.string (checked), Sys.number, and New defined type (Sys.string, Sys.number). Numbered callouts 1, 2, and 3 highlight the 'Create Slot' button, the 'Slot Name' field, and the 'Save' button respectively.

## SDS 실습 – 피자 주문 챗봇

### Get\_pizza\_info Task – Entities 설정

동일한 방법으로 아래와 같이 나머지 Entity도 만들어 준다.

Class	Slot	Slot 설명
Pizza	type	피자 종류, Sys.string 대표값: 리얼바비큐, 포테이토, 슈퍼디럭스, 슈퍼슈프림, 베이컨체더치즈, 불고기, 페퍼로니, 치즈
	size	피자 크기, Sys.string 대표값: 미디엄, 라지
	dough	피자 도우, Sys.string 대표값: 오리지널, 나폴리, 썬
	price	피자 가격, Sys.number

Pizza ▲	피자 클래스
▪ type	@ Sys.string
▪ size	@ Sys.string
▪ dough	@ Sys.string
▪ price	@ Sys.number

## SDS 실습 – 피자 주문 챗봇

### Get\_pizza\_info Task – Instance 설정

해당 Class의 Instances를 클릭하여, 값을 저장해준다. (상세 데이터는 다음 페이지 참조)

The image shows two overlapping screenshots of the SDS (Software Development Studio) interface. The top screenshot shows the 'Get\_pizza\_info' task configuration. The bottom screenshot shows the 'Instances' table for the 'Pizza' class.

**Top Screenshot (Configuration View):**

- Search bar: search slot...
- Task Name: Get\_pizza\_info
- Class: Pizza ▲
- Attributes:
  - type: @ Sys.string
  - price: @ Sys.number
  - size
  - dough
- Buttons: Create Slot, **Instances** (highlighted with a blue circle and arrow), Edit, Delete.

**Bottom Screenshot (Instances View):**

- Search bar: search slot...
- Buttons: 고급검색, 원래대로
- Class: Pizza
- Buttons: Delete All, Download Instance, Upload Instance
- Table (highlighted with a blue dashed box and a blue circle '2'):
 

type	price	size	dough	
치즈	미디엄	썸	15000	🗑️
치즈	미디엄	나폴리	15000	🗑️
				🗑️
- Button: SAVE (highlighted with a blue circle and arrow '3')

## SDS 실습 – 피자 주문 챗봇

### Get\_pizza\_info Task – Instance 설정

#### Pizza Class Instance

Pizza

Download Instance

Upload Instance

type	size	dough	price	
치즈	미디엄	썬	15000	🗑
치즈	미디엄	나폴리	15000	🗑
치즈	미디엄	오리지널	15000	🗑
치즈	라지	썬	21900	🗑
치즈	라지	나폴리	21900	🗑
치즈	라지	오리지널	21900	🗑
페퍼로니	미디엄	썬	16000	🗑
페퍼로니	미디엄	나폴리	16000	🗑
페퍼로니	미디엄	오리지널	16000	🗑
페퍼로니	라지	썬	22900	🗑
				🗑

# SDS 실습 – 피자 주문 챗봇

## Get\_pizza\_info Task – Property 설정

### Task Property에서 Slot 값 질문 설정 (1/2)

**GENIEDIALOG** Overview

AI 매니저 님 | 비밀번호 변경 | 로그아웃

Pizza\_bot

Task Graph

Greet

Get\_pizza\_info

Entities

Intents

Property

Entities

Intents

Manual

Users

Samples

Logs

저장 된 내용

Task Name: Get\_pizza\_info

Description: 주문 받을 피자 정보(피자 종류, 크기, 도우)를 입력 받음. 추가적으로, 사용자의 발화에 따라 피자 종류와 가격을 알려줌.

Initial: ☐

Task Goal:

Related Slots:

Reset Slots:

Task 시작 시, 시스템 발화 정의

+ 추가

주문하실 피자 종류, 크기, 도우를 말씀해주세요.

사용자에게 Slot 값 질문하기

+ 추가

Task 다시 시작할 경우, 시스템 발화 정의

+ 추가

Task action 및 모니터링 정보 정의

보기

## SDS 실습 – 피자 주문 챗봇

### Get\_pizza\_info Task – Property 설정

#### Task Property에서 Slot 값 질문 설정 (2/2)

\* 작성 완료 후 상단에 Save 눌러주기

사용자에게 Slot 값 질문하기

+ 추가

대상슬롯	Pizza.type	▲
Pizza.type	어떤 종류의 피자를 주문하시겠습니까? (e.g. 페퍼로니, 치즈, 불고기 등)	▼
대상슬롯	Pizza.size	▲
Pizza.size	어떤 크기의 피자를 주문하시겠습니까? (미디엄, 라지)	▼
대상슬롯	Pizza.dough	▲
Pizza.dough	어떤 도우의 피자를 주문하시겠습니까? (오리지널, 나폴리, 썬)	▼

# SDS 실습 – 피자 주문 챗봇

## Get\_pizza\_info Task – Intent 설정

### Intent 설정 – inform\_pizza (사용자가 주문할 피자 정보를 말하는 Intent) (1/6)

The screenshot shows the GENIEDIALOG interface for setting up an intent. The interface is divided into a left sidebar and a main workspace.

- Left Sidebar:** Contains a menu with 'Entities', 'Intents', and 'Property'. A blue dashed box labeled '1' highlights the 'Intents' option.
- Top Bar:** Contains buttons for 'Download Intent', 'Upload Intent', 'Copy Intent', '의도 Slot 등록', and 'Create Intent'. A blue dashed box labeled '2' highlights the 'Create Intent' button, with a blue arrow pointing down to the main workspace.
- Main Workspace:**
  - Intent Type:** A text input field containing 'inform\_pizza', highlighted by a blue dashed box labeled '3'.
  - 사용자 발화 (User Utterances):** A list of example sentences: '페퍼로니 피자요', '치즈 피자 라지 사이즈요', and '페퍼로니 피자 라지 사이즈 오리지널 도우요'. This section is highlighted by a blue dashed box labeled '4'.
  - 시스템 응답 정의 (System Response Definition):** A text input field at the bottom.
  - Bottom Right:** A blue dashed box labeled '5' highlights a '+ 추가' (Add) button, and a blue dashed box labeled '6' highlights a '+ 추가' button below it.



## SDS 실습 – 피자 주문 챗봇

### Get\_pizza\_info Task – Intent 설정 (Entity 태깅)

#### Entity 태깅하기

The screenshot shows the Maum Academy interface for setting up the Get\_pizza\_info task. The 'Intent Type' is set to 'inform\_pizza'. The '사용자 발화' (User Utterance) section contains three examples: '페퍼로니 피자요', '치즈 피자 라지 사이즈요', and '페퍼로니 피자 라지 사이즈 **오리지널** 도우요'. The third example is highlighted with a blue dashed box and a blue circle with the number 1. A tooltip with a blue circle with the number 2 points to the '오리지널' entity, stating '선택하려는 slot이 없습니다. <직접선택>'. The '시스템 응답 정의' (System Response Definition) section shows two responses: '<Pizza.type> 피자는 판매하고 있지 않습니다.' and '<Pizza.size> 사이즈는 판매하고 있지 않습니다.'. A modal window titled 'Entity 검색' (Entity Search) is open, showing a list of entities for 'Pizza' with 'Pizza.dough' selected. A blue circle with the number 3 points to the 'Pizza.dough' entity in the modal.

Intent Type

inform\_pizza

사용자 발화

페퍼로니 피자요

치즈 피자 라지 사이즈요

페퍼로니 피자 라지 사이즈 **오리지널** 도우요

선택하려는 slot이 없습니다.  
<직접선택>

시스템 응답 정의

<Pizza.type> 피자는 판매하고 있지 않습니다.

<Pizza.size> 사이즈는 판매하고 있지 않습니다.

Entity 검색

Pizza

☐ Pizza.type

☐ Pizza.size

☒ Pizza.dough

☐ Pizza.price

Cancel

# SDS 실습 – 피자 주문 챗봇

## Get\_pizza\_info Task – Intent 설정

### Intent 설정 – inform\_pizza (사용자가 주문할 피자 정보를 말하는 Intent) (2/6)

시스템 응답 정의

시스템 발화 | 시스템 발화 후 행동

시스템 발화 조건: true

시스템 Intent: inform()

시스템 발화 조건

and or ( ) space

Intent 검사 | Entity 검사 | Task 검사

Entity 검사

사용자 발화 내용

슬롯 값 존재 확인

슬롯 값 개수 확인

슬롯 값 정보 확인

사용자 발화 슬롯 값 Instance 존재 확인

Slot 검색

Pizza

Pizza.type

Pizza.size

Pizza.dough

Pizza.price

Cancel

# SDS 실습 – 피자 주문 챗봇

## Get\_pizza\_info Task – Intent 설정

### Intent 설정 – inform\_pizza (사용자가 주문할 피자 정보를 말하는 Intent) (3/6)

시스템 발화 조건

Entity("Pizza.type", "user\_history", "true") and Instance("Pizza.type", "false") Clear

1 and or ( ) space

Intent 검사 Entity 검사 Task 검사

Entity 검사 5 Add Clear

사용자 발화 내용 >

- 슬롯 값 존재 확인  
 true false
- 슬롯 값 개수 확인  
 ==
- 슬롯 값 정보 확인  
 ==

사용자 발화 슬롯 값 Instance 존재 확인

2  클릭 4 true false

Slot 검색

Pizza

3 ☐ Pizza.type

☐ Pizza.size

☐ Pizza.dough

☐ Pizza.price

Cancel

# SDS 실습 – 피자 주문 챗봇

## Get\_pizza\_info Task – Intent 설정

### Intent 설정 – inform\_pizza (사용자가 주문할 피자 정보를 말하는 Intent) (4/6)

2
+ 추가

시스템 응답 정의

시스템 발화
시스템 발화 후 행동

시스템 발화 조건 </>

Entity("Pizza.type", "user\_history", "true") and Instance("Pizza.type", "false")

시스템 intent inform()

1 <Pizza.type> 피자는 판매하고 있지 않습니다.

3
+ Add

시스템 발화
시스템 발화 후 행동

시스템 발화 조건 </>

Entity("Pizza.size", "user\_history", "true") and Instance("Pizza.size", "false")

시스템 intent inform()

<Pizza.size> 사이즈는 판매하고 있지 않습니다.

+ Add

시스템 발화
시스템 발화 후 행동

시스템 발화 조건 </>

Entity("Pizza.dough", "user\_history", "true") and Instance("Pizza.dough", "false")

시스템 intent inform()

<Pizza.dough> 도우는 판매하고 있지 않습니다.

+ Add

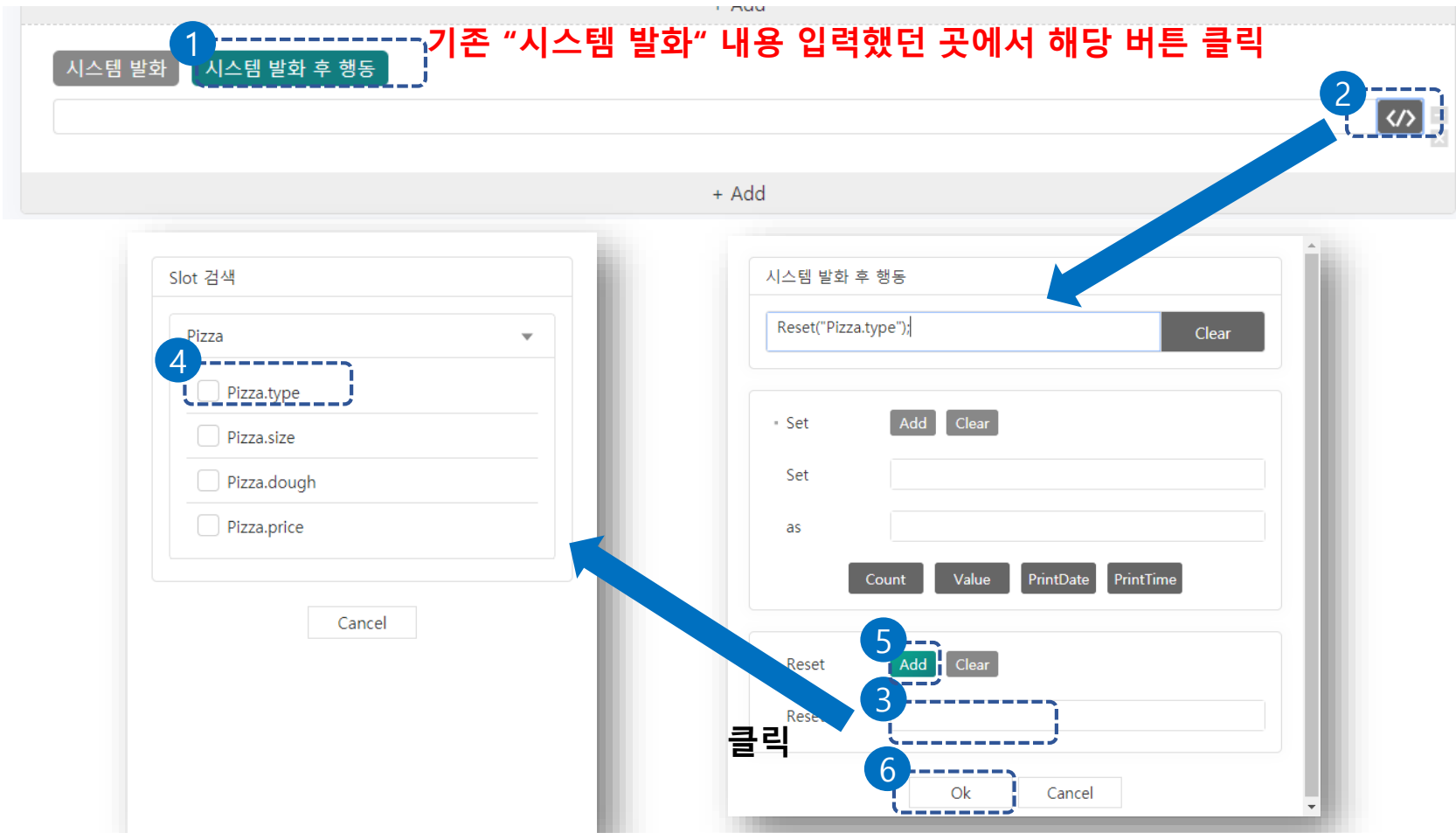
© 2019, MindsLab. All Rights Reserved

MINDs Lab

## SDS 실습 – 피자 주문 챗봇

### Get\_pizza\_info Task – Intent 설정

#### Intent 설정 – inform\_pizza (사용자가 주문할 피자 정보를 말하는 Intent) (5/6)



1 기존 "시스템 발화" 내용 입력했던 곳에서 해당 버튼 클릭

2

+ Add

Slot 검색

Pizza

4

☐ Pizza.type

☐ Pizza.size

☐ Pizza.dough

☐ Pizza.price

Cancel

시스템 발화 후 행동

Reset("Pizza.type") Clear

Set Add Clear

Set

as

Count Value PrintDate PrintTime

Reset Add Clear

3

클릭


6

Ok Cancel

## SDS 실습 – 피자 주문 챗봇

### Get\_pizza\_info Task – Intent 설정

#### Intent 설정 – inform\_pizza (사용자가 주문할 피자 정보를 말하는 Intent) (6/6)

시스템 발화	시스템 발화 후 행동
	<input type="text" value="Reset('Pizza.type');"/> 
	<input type="text" value="Reset('Pizza.size');"/> 
	<input type="text" value="Reset('Pizza.dough');"/> 

# SDS 실습 – 피자 주문 챗봇

## Get\_pizza\_info Task – Intent 설정

### Intent 설정 – ask\_pizza\_type (피자 종류 문의 Intent)

Intent Type

ask\_pizza\_type

사용자 발화

검색

피자 뭐뭐 있어요

어떤 피자 있어요

시스템 응답 정의

+ 추가

시스템 발화

시스템 발화 후 행동

시스템 발화 조건

true

</>

시스템 intent

inform()

피자 종류는 다음과 같습니다.  
(<Pizza.type>)

+ Add

# SDS 실습 – 피자 주문 챗봇

## Get\_pizza\_info Task – Intent 설정

### Intent 설정 – ask\_pizza\_price (피자 가격 문의 Intent)

Intent Type

사용자 발화

시스템 응답 정의

시스템 발화

시스템 발화 후 행동

시스템 발화 조건

Entity("Pizza.type", "user\_history", "true") and Instance("Pizza.type", "false")

시스템 intent

inform()

<Pizza.type> 피자는 판매하고 있지 않습니다.

+ Add

시스템 발화

시스템 발화 후 행동

시스템 발화 조건

Entity("Pizza.type", "user\_history", "true") and Instance("Pizza.type", "true")

시스템 intent

inform()

<Pizza.type> 피자의 가격은 다음과 같습니다.

<Pizza.type> 피자 {<Pizza.size> 사이즈 <Pizza.dough> 도우: <Pizza.price> 원}

시스템 발화

시스템 발화 후 행동

Reset("Pizza.type");

시스템 발화

시스템 발화 후 행동

Reset("Pizza.type");



## SDS 실습 – 피자 주문 챗봇

### Ask\_drink Task – Entity 설정

#### Entity 설정

Class	Slot	Slot 설명
Ask_drink	answer	음료 주문 여부, Sys.string, Instance와 연동 체크 해제

## SDS 실습 – 피자 주문 챗봇

### Ask\_drink Task – Property 설정

#### Property 설정

시스템 발화

시스템 발화 후 행동

시스템 발화 조건

true



시스템 intent

request\_ask\_drink()

Ask\_drink.answer

<Pizza.type> 피자 <Pizza.size> 사이즈 <Pizza.dough> 도우의 가격은 <Pizza.price> 원 입니다.음료도 주문하시겠습니까?

다음 사용자 발화 DA 제한

사용자에게 Slot 값 질문하기

+ 추가

대상슬롯

Ask\_drink.answer

Ask\_drink.answer

음료도 주문하시겠습니까?

## SDS 실습 - 피자 주문 챗봇

### Ask\_drink Task - Intent 설정

**Intent 설정 - affirm** (음료 주문 여부 물음에 대해 사용자가 긍정적으로 대답한 Intent)

시스템 응답 정의

+ 추가

시스템 발화

시스템 발화 후 행동

시스템 발화 조건

Intent("system", "request\_ask\_drink", "2", "true")

</>

시스템 intent

inform()

시스템 발화

시스템 발화 후 행동

Set("Ask\_drink.answer", "yes");

</>

이전 시스템 발화가 음료 주문 여부를 물어보는 것이고 (-> Intent("system", "request\_ask\_drink", "2", "true")), 사용자가 긍정적으로 대답했다면 (-> affirm), Ask\_drink.answer의 Slot 값을 "yes"로 설정합니다. 여기서 시스템 응답이 비어있는 이유는, 이 응답 대화에서는 별다른 발화는 하지 않고 Ask\_drink.answer의 Slot 값만 변화시키기 위함입니다.

## SDS 실습 - 피자 주문 챗봇

### Ask\_drink Task - Intent 설정

#### Intent 설정 - negate (음료 주문 여부 물음에 대해 사용자가 부정적으로 대답한 Intent)

시스템 응답 정의

+ 추가

시스템 발화

시스템 발화 후 행동

시스템 발화 조건

Intent("system", "request\_ask\_drink", "2", "true")

</>

시스템 intent

inform()

시스템 발화

시스템 발화 후 행동

Set("Ask\_drink.answer", "no");

</>

이전 시스템 발화가 음료 주문 여부를 물어보는 것이고 (-> Intent("system", "request\_ask\_drink", "2", "true")), 사용자가 부정적으로 대답했다면 (-> negate), Ask\_drink.answer의 Slot 값을 "no"로 설정합니다. 여기서 시스템 응답이 비어있는 이유는, 위에서와 마찬가지로, 이 응답 대화에서는 별다른 발화는 하지 않고 Ask\_drink.answer의 슬롯 값만 변화시키기 위함입니다.

## SDS 실습 – 피자 주문 챗봇

### Get\_drink\_info Task – Entity 설정




#### Entity 및 Instance 설정

Class	Slot	Slot 설명
Drink	type	음료 종류, Sys.string 대표값: 콜라, 사이다
	price	음료 가격, Sys.number

Drink

Download Instance

Upload Instance

type	price	
사이다	1800	
콜라	1800	
		

# SDS 실습 – 피자 주문 챗봇

## Get\_drink\_info Task – Property 설정

### Property 설정

시스템 발화

시스템 발화 후 행동

시스템 발화 조건

</>

시스템 intent

주문하실 음료를 말씀해주세요.  
(콜라, 사이다)

사용자에게 Slot 값 질문하기

+ 추가

대상슬롯

▲

Drink.type

주문하실 음료를 말씀해주세요.  
(콜라, 사이다)

▼

# SDS 실습 - 피자 주문 챗봇

## Get\_drink\_info Task - Intent 설정

### Intent 설정 - inform\_drink (사용자가 주문할 음료 정보를 말하는 Intent)

Intent Type

inform\_drink

사용자 발화

검색

콜라요

사이다 있어요

시스템 발화

시스템 발화 후 행동

시스템 발화 조건

Entity("Drink.type", "user\_history", "true") and Instance("Drink.type", "false")



시스템 intent

inform()

<Drink.type>은/는 판매하고 있지 않습니다.

## SDS 실습 - 피자 주문 챗봇

### Get\_drink\_info Task - Intent 설정

#### Intent 설정 - ask\_drink\_type (판매 중인 음료 종류를 물어보는 Intent)

Intent Type

ask\_drink\_type

사용자 발화

음료 뭐뭐 있어요

음료수 뭐뭐 팔아요

검색

시스템 발화

시스템 발화 후 행동

시스템 발화 조건

true



시스템 intent

inform()

판매 중인 음료는 다음과 같습니다.  
{<Drink.type>}



# SDS 실습 - 피자 주문 챗봇

## Get\_drink\_info Task - Intent 설정

### Intent 설정 - ask\_drink\_price (음료 가격을 물어보는 Intent) (1/2)

Intent Type

사용자 발화

검색

콜라 얼마야

음료수 가격이 어떻게 돼요

시스템 발화

시스템 발화 후 행동

시스템 발화 조건

Entity("Drink.type", "user\_history", "true") and Instance("Drink.type", "false")

</>

시스템 intent

inform()

<Drink.type>는 j 판매하고 있지 않습니다.

---

시스템 발화

시스템 발화 후 행동

시스템 발화 조건

Entity("Drink.type", "user\_history", "true") and Instance("Drink.type", "true")

시스템 intent

inform()

<Drink.type>의 가격은 <Drink.price>원 입니다.

시스템 발화

시스템 발화 후 행동

Reset("Drink.type");

---

시스템 발화

시스템 발화 후 행동

Reset("Drink.type");

사용자가 물어본 음료가 instance에 없다면 판매하고 있지 않다고 알립니다. Instance에 있다면 음료 가격을 알려줍니다. 그리고 관련 Slot 값을 초기화합니다.

## SDS 실습 – 피자 주문 챗봇

### Get\_drink\_info Task – Intent 설정

#### Intent 설정 - ask\_drink\_price (음료 가격을 물어보는 Intent) (2/2)

시스템 발화

시스템 발화 후 행동

시스템 발화 조건

Entity("Drink.type", "user\_history", "false")



시스템 intent

inform()

판매 중인 음료의 가격은 다음과 같습니다.  
{<Drink.type> <Drink.price> 원}

사용자가 특정 음료를 말하지 않았다면, 판매 중인 모든 음료의 정보를 알려줍니다.

## SDS 실습 - 피자 주문 챗봇

### Get\_drink\_info Task - Intent 설정

#### Intent 설정 - inform\_no\_drink (사용자가 음료 주문을 취소하는 Intent)

Intent Type

inform\_no\_drink

사용자 발화

음료 그냥 안 살래요

음료 주문 취소

검색

시스템 응답 정의

+ 추가

별다른 시스템 응답을 하지 않고 바로 다음 Task로 넘어갑니다.

## SDS 실습 – 피자 주문 챗봇

### Get\_payment\_info Task – Entity 설정

#### Entity 설정

Class	Slot	Slot 설명
Order	payment_type	결제 수단, Sys.string, Instance와 연동 체크 해제

# SDS 실습 – 피자 주문 챗봇

## Get\_payment\_info Task – Property 설정

### Property 설정

Task 시작 시, 시스템 발화 정의 + 추가

시스템 발화

시스템 발화 후 행동

시스템 발화 조건

Entity("Drink.type", "user\_history", "false") or Intent("user", "inform\_no\_drink", "0", "true")

시스템 intent

inform()

총 가격은 <Pizza.price>원 입니다. 결제는 현금으로 하시겠어요, 카드로 하시겠어요?

---

시스템 발화

시스템 발화 후 행동

시스템 발화 조건

Entity("Drink.type", "user\_history", "true")

시스템 intent

inform()

음료 가격 <Drink.price>원을 포함하여 총 가격은 <script=Value("Pizza.price")+Value("Drink.price")>원 입니다. 결제는 현금으로 하시겠어요, 카드로 하시겠어요?

Task 진입 시, 사용자가 음료를 주문했는지 여부에 따라 서로 다른 시스템 응답이 나옵니다. 우선 사용자가 주문할 음료를 말하지 않거나, 음료를 주문하지 않겠다고 하면 피자 금액만 안내가 됩니다. 반면 사용자가 주문할 음료를 말한 경우 피자 가격과 음료 가격이 더해진 총 금액이 안내가 됩니다. 그런 후에 결제 수단을 물어봅니다.

사용자에게 Slot 값 질문하기 + 추가

대상슬롯

Order.payment\_type

Order.payment\_type

결제는 현금으로 하시겠어요, 카드로 하시겠어요?

## SDS 실습 - 피자 주문 챗봇

### Get\_payment\_info Task - Intent 설정

#### Intent 설정 - inform\_payment\_type (결제 수단을 말하는 Intent)

Intent Type

inform\_payment\_type

사용자 발화

현금이요

신용카드로 할게요

검색

시스템 응답 정의

+ 추가

## SDS 실습 – 피자 주문 챗봇

### Bye Task – Property 설정

#### Property 설정

Task 시작 시, 시스템 발화 정의 + 추가

시스템 발화

시스템 발화 후 행동

시스템 발화 조건

true

</>

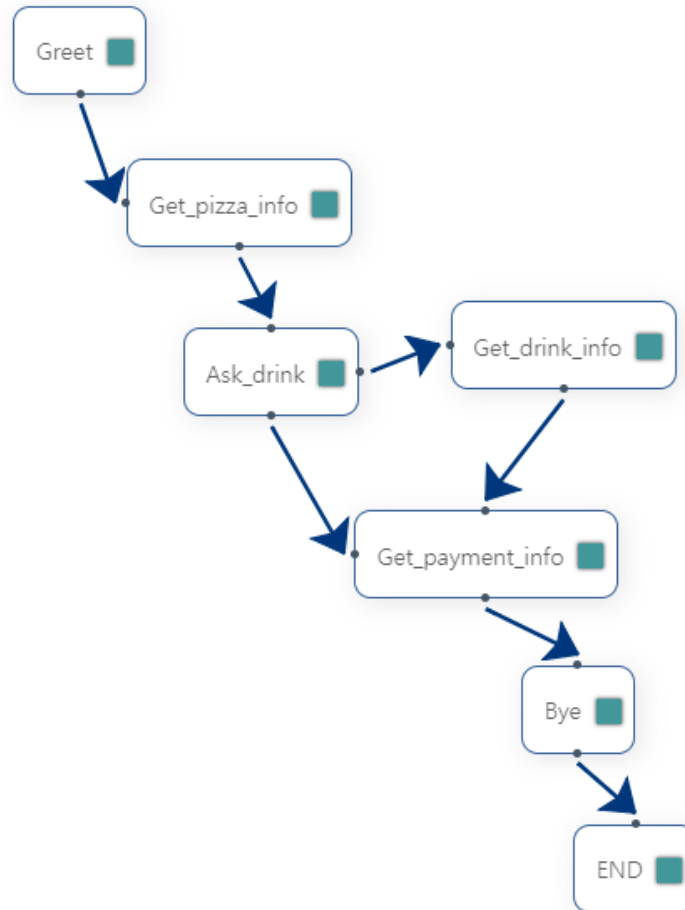
시스템 intent

bye()

30분 안에 배달됩니다. <Order.payment\_type>을:j 미리 준비해주세요. 이용해 주셔서 감사합니다.

## SDS 실습 – 피자 주문 챗봇

### Task 연결





# SDS 실습 - 피자 주문 챗봇

## 대화시스템 구동

**GENIEDIALOG** Task Graph

PizzaHoot

Task Graph

Greet +

Get\_pizza\_info +

Ask\_drink +

Get\_drink\_info +

Get\_payment\_info +

Bye +

Entities

Intents

Manual

```
graph TD; Greet --> Get_pizza_info; Get_pizza_info --> Ask_drink; Get_pizza_info --> Get_drink_info; Ask_drink --> Get_payment_info; Get_drink_info --> Get_payment_info; Get_payment_info --> Bye; Bye --> END;
```

Create Task Save

Dialog

권오욱 님 | 로그아웃

공지사항

DIALOG

지식저장 학습 구동

현재 도메인 PizzaHoot

Start Stop Clear DevLog

DevTest용

여기에 입력해주세요.

# SDS 실습 – 피자 주문 챗봇

## 대화시스템 구동 – 학습과 지식저장

❗ 공지사항

### DIALOG

지식저장 **1** 학습 구동

현재 도메인

PizzaHoot

```
20 0.00% 0.0 1 2.8 1.47e+02
3.91e+00 1 3938 681
```

```
10 0.00% 0.0 0 4.6 5.27e+02
1.04e+01 14 2160 706
```

```
20 0.00% 0.0 0 5.0 7.99e+01
1.29e+01 468 3726 2049
```

```
30 0.00% 0.0 0 5.0 4.77e+01
1.32e+01 2 4719 2981
```

```
[/phase 3:SVM training]
```

```
FINISHED
```

대화시스템을 동작하기 전에 먼저 "학습"을 해 주어야 합니다.

"학습"은 지금까지 도구에서 작업한 내용을 대화시스템이 사용하는 지식으로 저장하고, 대화이해 학습을 수행하여 대화시스템이 동작할 수 있는 지식들을 준비합니다.

학습이 완료되면 아래와 같은 메세지 "FINISHED"가 출력됩니다.

# SDS 실습 - 피자 주문 챗봇

## 대화시스템 구동 - 구동

① 공지사항

### DIALOG

지식저장 학습 **1** 구동

현재 도메인 PizzaHoot

Start Stop Clear DevLog DevTest용

system: 안녕하세요, 함께 즐겨요 피자훗입니다. 주문하실 피자 종류, 크기, 도우를 말씀해주세요.

여기에 입력해주세요.

Turn 1 ⏪ ⏩

Task

Previous task : Greet(1)  
Current task : Get\_pizza\_info(1)

Entities

Intents

User  
└ utterance :

System  
└ utterance : 안녕하세요, 함께 즐겨요 피자훗입니다. 주문하실 피자 종류, 크기, 도우를 말씀해주세요.  
└ intent : inform inform  
└ pattern : 안녕하세요, 함께 즐겨요 피자훗입니다. 주문하실 피자 종류, 크기, 도우를 말씀해주세요.

### DIALOG

지식저장 학습 **구동**

현재 도메인 PizzaHoot

Start Stop Clear DevLog DevTest용

system: 안녕하세요, 함께 즐겨요 피자훗입니다. 주문하실 피자 종류, 크기, 도우를 말씀해주세요.  
user: 페페로니 피자요  
system: 어떤 도우의 피자를 주문하시겠습니까?  
(오리지널, 나폴리, 썬)

여기에 입력해주세요.

Turn 2, 3 ⏪ ⏩

Task

Previous task : Greet(1)  
Current task : Get\_pizza\_info(1)

Entities

Pizza.type = "페페로니"(2)

Intents

User  
└ utterance : 페페로니 피자요  
└ intent (1) : inform\_pizza(Pizza.type="페페로니") (1.0)

System  
└ utterance : 어떤 도우의 피자를 주문하시겠습니까?  
(오리지널, 나폴리, 썬)  
└ intent : request  
└ pattern : 어떤 도우의 피자를 주문하시겠습니까?<br>(오리지널, 나폴리, 썬)

AI 전문가로 향하는  
가장 빠르고 정확한 길,  
마음아카데미

# 감사합니다.

**MINDs** Lab. AI Platform Company