# Continual Learning Dialogue Systems

## – Learning during Conversation

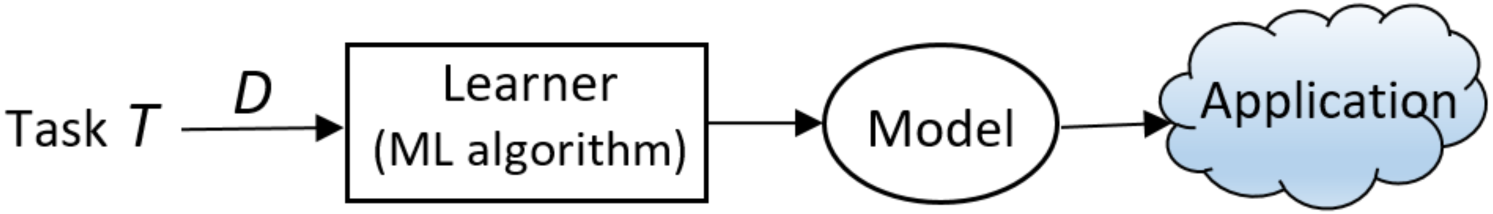

**Sahisnu Mazumder**
Multimodal Dialogue & Interactions
Intel Labs, USA

**Bing Liu**
Department of Computer Science
University of Illinois at Chicago

**Web:** https://www.cs.uic.edu/~liub/SIGIR-2022-CL-Chatbot.html

# Introduction

- Classical machine learning: Isolated single-task learning




  - Key weaknesses
    - **Closed-world assumption**: nothing new or unexpected in application
    - **No knowledge accumulation or transfer**: isolated learning
    - **Model is fixed after deployment**: no learning or adaptation
- **Focus of this tutorial**: Learning after/post model deployment on the job, particularly, in the dialogue domain.
    - In the **open-world** with unknowns and distribution changes.



- Chen and Liu. *Lifelong machine learning*. Morgan & Claypool. 2016, 2018
- Liu. *Learning on the Job: Online Lifelong and Continual Learning*. AAAI - 2020

# Learning on the job (while working)

(Liu, 2020, Chen and Liu, 2018)

- It is estimated that about 70% of our human knowledge comes from 'on-the-job' learning.
  - Only about 10% through formal training
  - The rest 20% through observation of others
- An AI agent should learn on the job too as
  - The world is too complex and constantly changing.
    - must learn and adapt continually to achieve autonomy.
  - Without this capability, an AI agent is not truly intelligent.


- Chen and Liu. *Lifelong machine learning*. Morgan & Claypool. 2018
- Liu. *Learning on the Job: Online Lifelong and Continual Learning*. AAAI - 2020

# Self-driving cars : A motivating example

- Self-driving cars cannot reach human-level of driving with only rules and off-line training.
  - Impossible to cover all corner cases
  - Real-world is full of unknowns or novelties.
- Has to learn & adapt continuously in its interaction with humans and the environment by itself

- Chen and Liu. Lifelong machine learning. Morgan & Claypool. 2018
- Liu. *Learning on the Job: Online Lifelong and Continual Learning.* AAAI - 2020

# A personal experience with a self-driving car

- I consulted for a self-driving car company for a year.
- Once we took a self-driving car for a field test on the road.
  - At a T-junction, the car suddenly stopped and refused to move.
    - Every direction was clear, and nothing was on the road.
- We had to take over manually and drove the car to the lab.
  - Debugging found that a sensor detected a pebble on the road.
- If the car could say ”*I detected an unknown object here. What should I do?*” we would have said “*It is safe. Go ahead*.”
  - The car can then learn the new object so that it will have no issue next time.
    - That is, **learning on the fly** or on the job.

# AI in open-world & Learning on the job
(Liu 2020, Liu & Mazumder 2021)

- AI has to learn continually and interactively on the job (after/post deployment) in the open-world
  - ✓ Needs to communicate with humans, e.g., take human instructions, ask when it has difficulty in the open world with unknowns, and learn from humans.
  - ✓ Communicating in natural language (NL) is a natural choice.

- Dialogue systems with on-the-job learning (i.e., *learning during conversation*) capability is essential for the next-generation dialogue systems


- Liu. *Learning on the Job: Online Lifelong and Continual Learning.* AAAI - 2020
- Liu and Mazumder. *Lifelong and Continual Learning Dialogue Systems: Learning during Conversation.* AAAI-2021

# Dialogue & Interactive Systems : Tasks

- Question-answering: provide **concise, direct answers to user queries** based on rich knowledge drawn from various data sources (QA Bots)
  - ✓ text collections such as Web documents
  - ✓ pre-compiled KBs such as sales and marketing database, factual KBs, etc.
- Task completion: help user **accomplish their tasks** (task completion bots)
  - ✓ restaurant reservation, meeting scheduling, trip planning
- Social chats: **converse seamlessly** and appropriately with users and provide useful recommendations (social chatbots)



- Gao, Galley and Li. *Neural approaches to conversational AI: Question answering, task-oriented dialogues and social chatbots*. Now Foundations and Trends, 2019.

# Example – Human-agent dialogues for business decision making

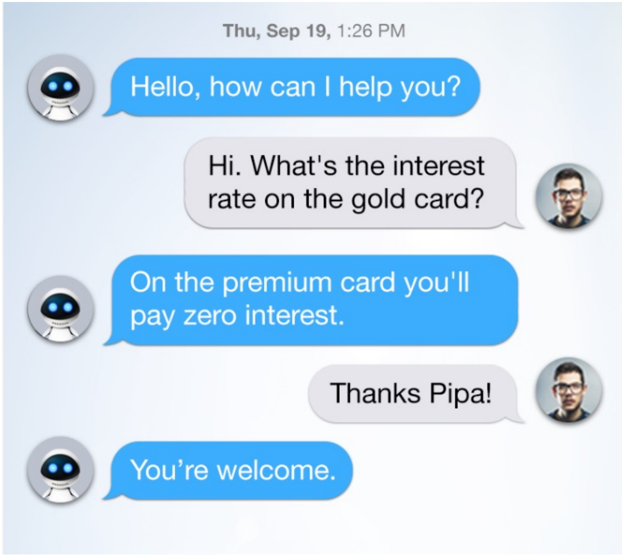
| | Turn | Dialogue |
|---|---|---|
| *Social chats* | 1 | *usr:* Good morning! |
| | | *agt:* Good morning, I hope you had a great weekend. |
| | 2 | *usr:* not really. |
| | | *agt:* You sound depressed. What can I do for you? |
| *Question-answering* | 3 | *usr:* I'd like to know where sales are lagging behind our forecast? |
| | | *agt:* The worst region is [country], where sales are 15% below projections. |
| | 4 | *usr:* Do you know why? |
| | | *agt:* The forecast for [product] growth was overly optimistic. |
| | 5 | *usr:* How can we turn this around? |
| | | *agt:* Here are the 10 customers in [country] with the most growth potential, per our CRM model. |
| *Task completion* | 6 | *usr:* Can you set up a meeting with the CTO of [company]? |
| | | *agt:* Yes, I've set up a meeting with [person name] for next month when you are in [location]. |
| | 7 | *usr:* Thanks! |

- Gao, Galley and Li. *Neural approaches to conversational AI: Question answering, task-oriented dialogues and social chatbots*. Now Foundations and Trends, 2019.

# Dialogue & Interactive Systems : Broad Categories

- **Task-oriented chatbots**
  - ✓ Complete tasks based on users' requests, e.g., providing the requested information and taking actions.
  - ✓ Personal assistants: Alexa, Siri, Google Home, etc.

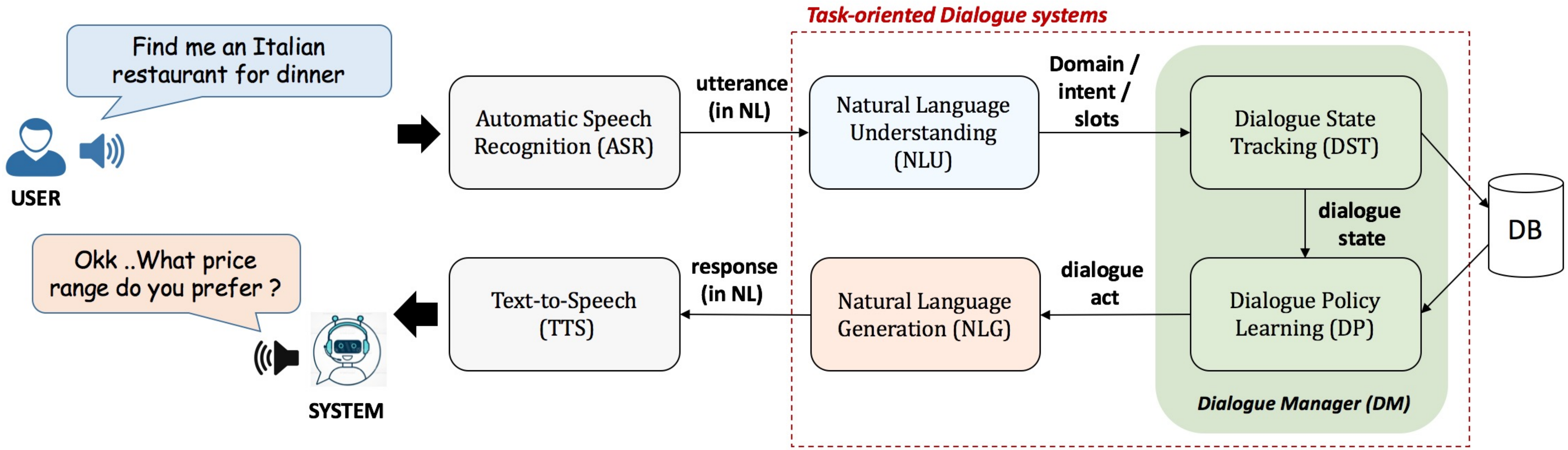

- **Chit-chat systems**
  - ✓ Conduct chit-chat type of conversation in wide range of topics without having a specific goal to complete.
  - ✓ Example – ELIZZA, Microsoft XiaoIce.





- Gao, Galley and Li. *Neural approaches to conversational AI: Question answering, task-oriented dialogues and social chatbots*. Now Foundations and Trends, 2019.

# Task-oriented Dialogue Systems




- Often designed as a Modular system –
  - ✓ Natural Language Understanding (NLU): Identify user intents and extract associated information
  - ✓ State Tracking: Track the dialogue state to capture all essential information in conversation so far
  - ✓ Dialogue Policy: Select the next action based on the current state
  - ✓ Natural Language Generation (NLG): Convert agent actions to natural language responses

# Natural Language Understanding (NLU)

- Domain Classification: Classifying the domain of the task.
  - ✓ E.g., is this user talking about airlines, programming an alarm clock, or dealing with their calendar?
- Intent Classification: What general task or goal is the user trying to accomplish?
  - ✓ E.g., Find a Movie, or Show a Flight, or Remove a Calendar Appointment.
- Slot filling: Extract the slots and fillers that the user intends the system to understand from their utterance with respect to their intent → a sequence labeling problem

- Jurafsky and Martin. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice Hall, 2020.

# Natural Language Understanding (NLU): Examples

"Find me an italian restaurant for dinner"

**Domain:** RESTAURANT
**Intent:** *SearchRestaurant*
**Slots:** {CUISINE: *Italian*, TIME: *dinner*}

"Wake me up tomorrow at 6"

**Domain:** ALARM-CLOCK
**Intent:** *Set-Alarm*
**Slots:** {DATE: *2022-07-11*, TIME: *06:00*}


- Jurafsky and Martin. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice Hall, 2020.

# State Tracking

User: Find me an italian restaurant for dinner.
    inform(cuisine=*Italian*; time=*dinner*)
System: Okk. What price range do you prefer?
User: may be in moderate
    inform(cuisine=*Italian*; time=*dinner*; price_range=$$)
System: Sure. There an nice italian resturant at moderate price range nearby.

User: Awesome! What's the name and address?
    inform(cuisine=*Italian*; time=*dinner*; price_range=$$);
    request(name; address)
System: The restaurant name is *Franco's Ristorante* and address is
    *300 W 31st St, Chicago, IL 60616.*

## dialogue-state:

the **entire state of the frame** at this point (the filler of each slot), as well as the **user's most recent dialogue act,** summarizing all of the user's constraints.



- Jurafsky and Martin. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice Hall, 2020.

# Dialogue Policy (DP) Learning

User: Find me an italian restaurant for dinner.
inform(cuisine=*Italian*; time=*dinner*)
System: Okk. What price range do you prefer?

inform(cuisine=*Italian*; time=*dinner*) [ *dialogue state* ]
⇓
request(price_range) [ *dialogue act* ]

- Decide what action the system should take next, that is, what dialogue act to generate.
  - ✓ At turn $i$ in the conversation, we want to predict which action $A_i$ to take, based on the **entire dialogue state** [entire sequence of dialogue acts from the system (A) and from the user (U)].

$$\hat{A}_i = \underset{A_i \in A}{\operatorname{argmax}} P(A_i | (A_1, U_1, \ldots, A_{i-1}, U_{i-1})$$

- Jurafsky and Martin. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice Hall, 2020.

# Natural Language Generation (NLG)

inform(name; address) [ *dialogue act* ]

$\Downarrow$

The restaurant name is NAME_SLOT and address is ADDRESS_SLOT

[ *delexicalized response* ]

- Once the policy has decided what speech act to generate, the NLG generates the text response
  - ✓ modeled in two stages, content planning (what to say) and sentence realization (how to say it).
- sentence realization is commonly achieved through delexicalization
  - ✓ Mapping from frames to delexicalized sentences using encoder decoder models



- Jurafsky and Martin. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice Hall, 2020.

# Social Chatbots

(Jurafsky et.al. 2020)

- Often Implemented using a unitary (non-modular) system
  - ✓ Rule-based systems
    - ✓ Works based on pattern/transform rules
    - ➢ Examples - ELIZA (Weizenbaum, 1966) and PARRY (Colby et al., 1971)
  - ✓ Corpus-based systems
    - ✓ Mimic human conversations by training on large amounts of human-human conversational data
    - ➢ Example - Microsoft XioIce

- Jurafsky and Martin. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice Hall, 2020.

# Social Chatbots : Rule-based systems

- ELIZA (Weizenbaum, 1966)
  worked by pattern/transform rules like the following one:

```
(0 YOU 0 ME) [pattern]
->
(WHAT MAKES YOU THINK I 3 YOU) [transform]
```

You hate me

into:

WHAT MAKES YOU THINK I HATE YOU

Men are all alike.
IN WHAT WAY
They're always bugging us about something or other.
CAN YOU THINK OF A SPECIFIC EXAMPLE
Well my boyfriend made me come here.
YOUR BOYFRIEND MADE YOU COME HERE
He says I'm depressed much of the time.
I AM SORRY TO HEAR YOU ARE DEPRESSED
..
..

*Example dialogue by ELIZA*

- Each ELIZA pattern/rule is linked to a keyword that might occur in a user sentence


- Jurafsky and Martin. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice Hall, 2020.

# Social Chatbots : Corpus-based systems

(Jurafsky et.al. 2020)

- Response by retrieval: considering user's turn as a query $\boldsymbol{q}$, the goal is to retrieve and repeat some appropriate turn $\boldsymbol{r}$ as the response from a corpus of conversations $\boldsymbol{C}$ (training set for the system)
- Score each turn in $\boldsymbol{C}$ as a potential response to the context $\boldsymbol{q}$ and select the highest-scoring one.

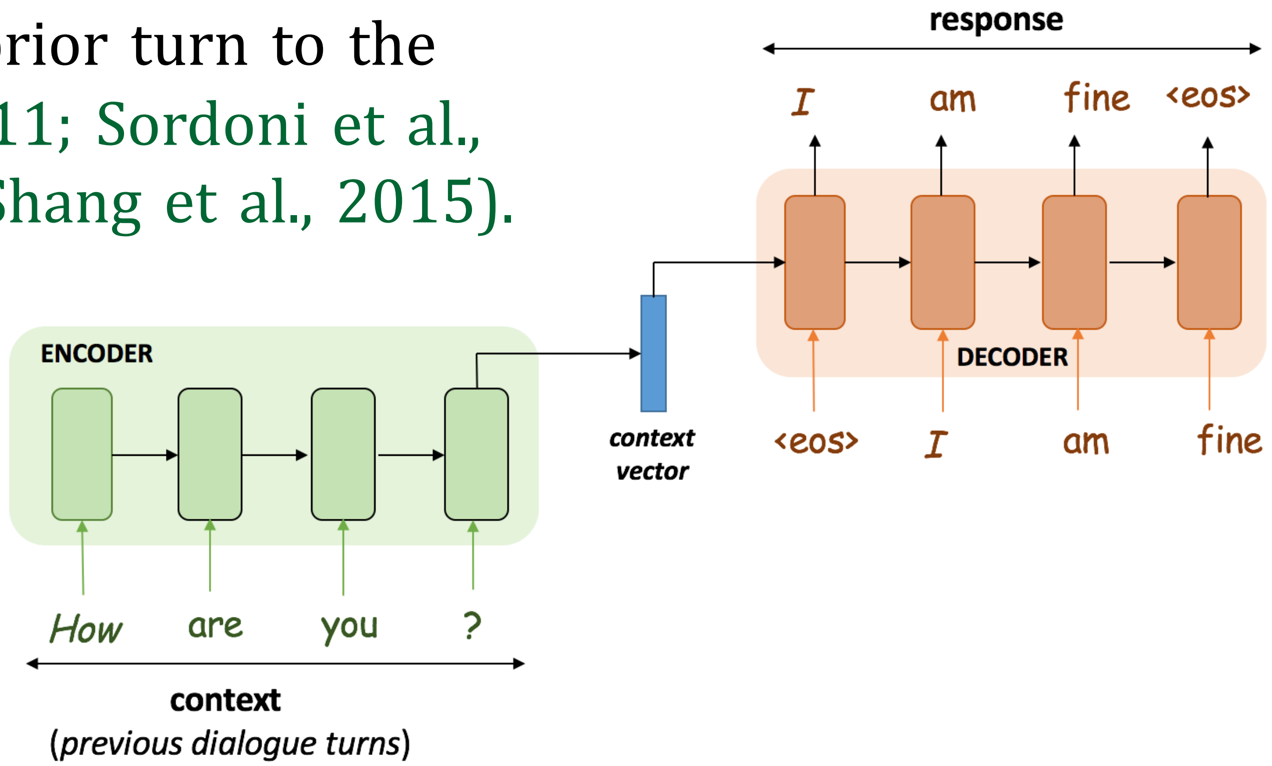
$$\text{response}(q, C) = \underset{r \in C}{\text{argmax}} \frac{q \cdot r}{|q||r|}$$

- Jurafsky and Martin. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice Hall, 2020.

# Social Chatbots : Corpus-based systems

- Response by Generation: Models response production as an encoder-decoder task
  - transducing from the user's prior turn to the system's turn (Ritter et al., 2011; Sordoni et al., 2015b; Vinyals and Le, 2015; Shang et al., 2015).

$$\hat{r}_t = \text{argmax}_{w \in V} P(w|q, r_1 \ldots r_{t-1})$$




- Jurafsky and Martin. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice Hall, 2020.

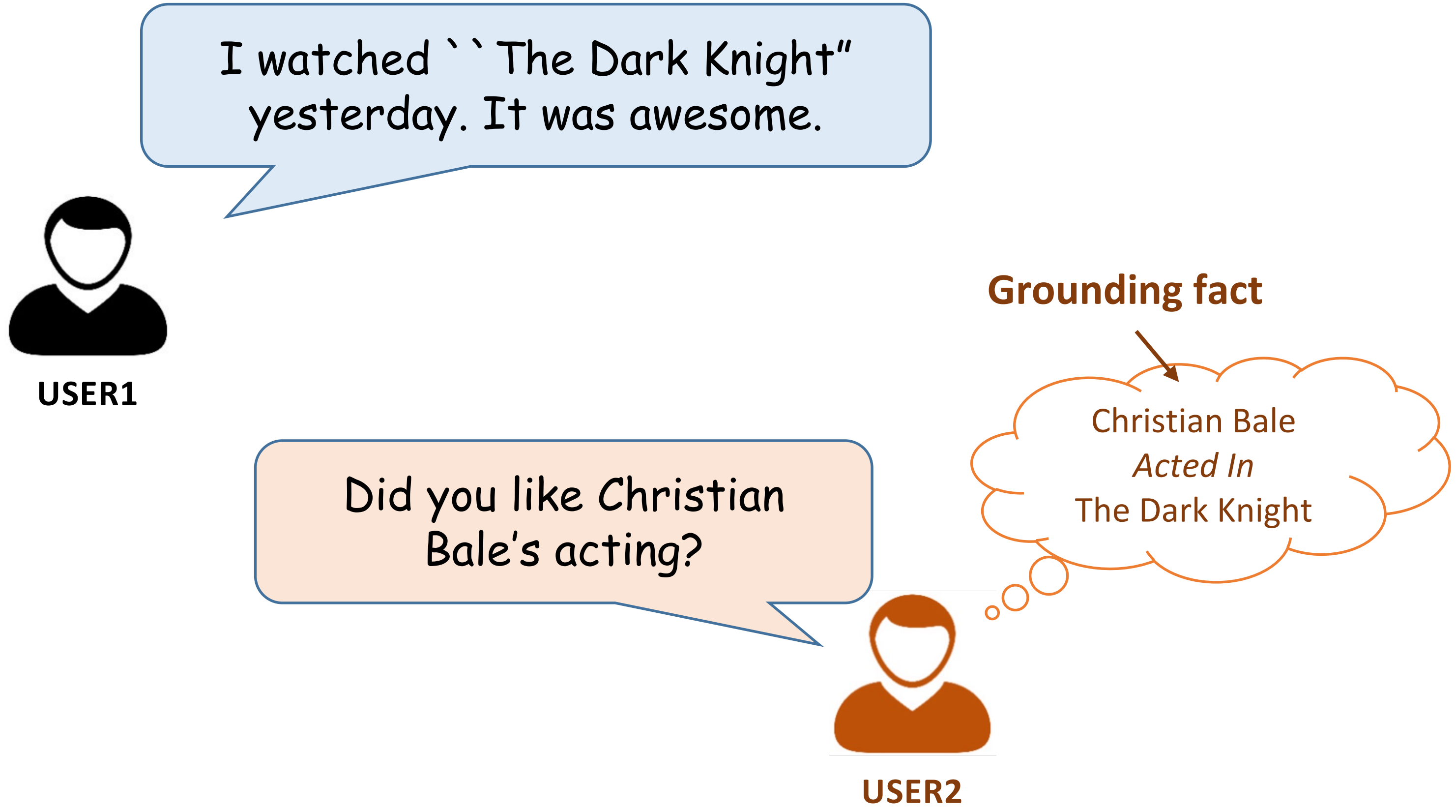
# Response Generation in Dialogue Systems: weaknesses

- Dull responses
  - e.g., *``I don't know"* , *``I don't have a clue"*.
- Out-of-context responses
- Semantic Conflicts
- Inconsistent response
- Lack of context-awareness
- so on ….



- Huang, Zhu, and Gao. Challenges in building intelligent open-domain dialog systems. ACM Transactions on Information Systems (TOIS), 2020

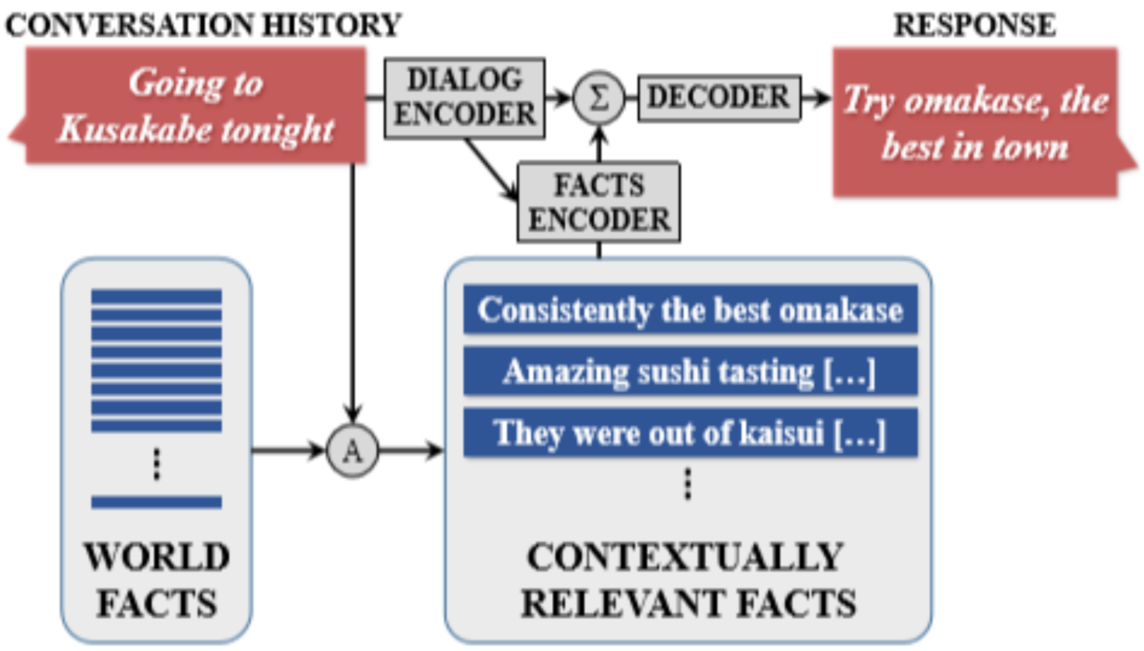
# Knowledge grounding in Conversation: Example



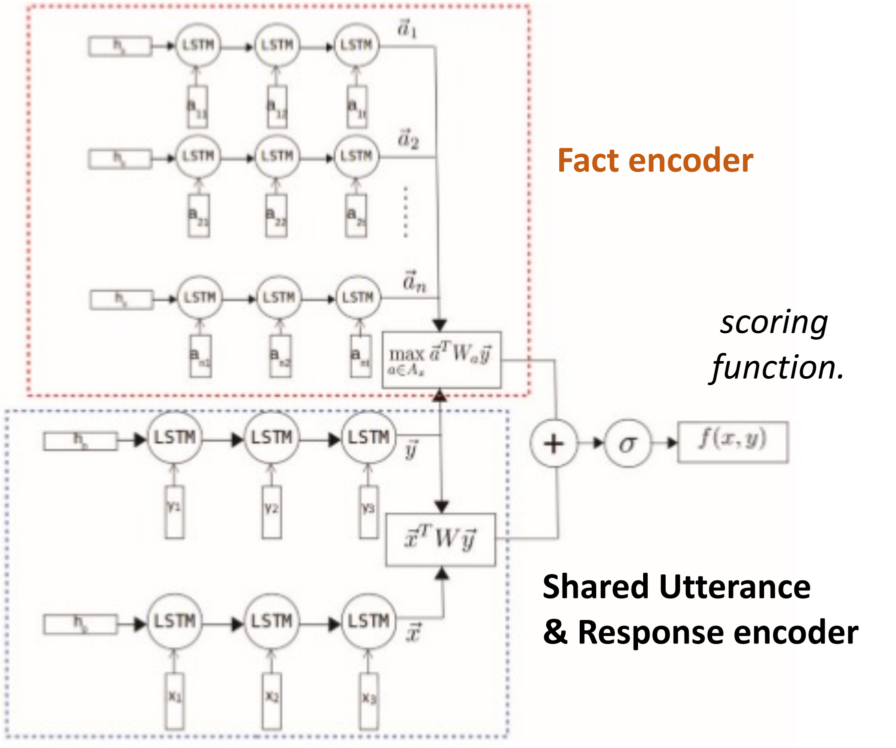

Knowledge grounding makes conversation **interesting** and **intelligent!**

# Knowledge-grounded Conversation Modeling

Recently researchers have begun to explore how to ground the chitchat in world knowledge to make the conversation more contentful and interesting




Ghazvininejad et. al., AAAI-2018




Young et.al., AAAI-2018

- Ghazvininejad et. al. A knowledge-grounded neural conversation model. *AAAI* 2018.
- Young et. al. Augmenting end-to-end dialogue systems with commonsense knowledge. AAAI 2018.

# Dialogue systems in the open-world : Challenges

- Built with **pre-collected** training **data, fixed rules** and **pre-compiled knowledge bases** (KBs)
  - ✓ Great deal of manual effort is needed
  - ✓ No matter how much data is collected, can’t cover all possible variations of natural language.

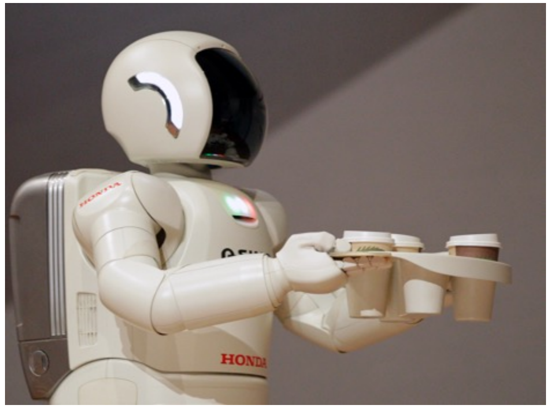
- Pre-compiled KB **can’t cover all rich knowledge** needed in practice
  - ✓ Knowledge bases are incomplete (West et. al. 2014)
  - ✓ KB of existing systems does not grow over time!

- West et. al. Knowledge base completion via search-based question answering. WWW-2014

# Chatbots should learn continually after deployment

(Chen & Liu 2018, Liu 2020, Liu & Mazumder 2021)

- Chatbot: human users may say things a chatbot does not understand.
  - It must learn new knowledge and new language expressions **during chatting**.
    - E.g., asking the current or other users.
  - Humans learn a great deal in our daily conversations
- Chatbots **should not** solely rely on offline training initiated by engineers.

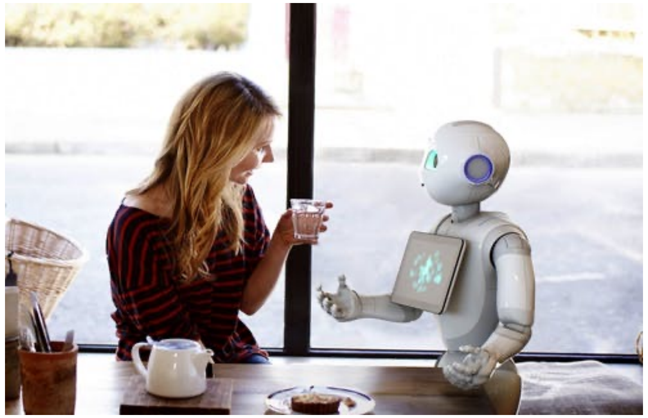

- Chen and Liu. *Lifelong machine learning*. Morgan & Claypool. 2016, 2018
- Liu. *Learning on the Job: Online Lifelong and Continual Learning*. AAAI - 2020
- Liu and Mazumder. *Lifelong and Continual Learning Dialogue Systems: Learning during Conversation.* AAAI-2021

# Learning during Conversation : Scopes

(Liu 2020, Liu & Mazumder 2021)

- **Passive learning**
  - ✓ Learning by reading web corpus, web tables or past conversation [information extraction]

- **Interactive learning**
  - ✓ Learning through interactive multi-turn dialogue [**our focus**]



- Liu. *Learning on the Job: Online Lifelong and Continual Learning.* AAAI - 2020
- Liu and Mazumder. *Lifelong and Continual Learning Dialogue Systems: Learning during Conversation.* AAAI-2021

# Goals of this Tutorial

- Introducing the paradigm of lifelong or continual learning and discuss various related problems and challenges in the context of conversational AI applications.
- Recent advancements in continual learning in Chatbots after model deployment via interactions with end-users.
- A discussion on the future scope for continual conversational learning and open challenges.

# Outline

# Classic definition of lifelong/continual learning

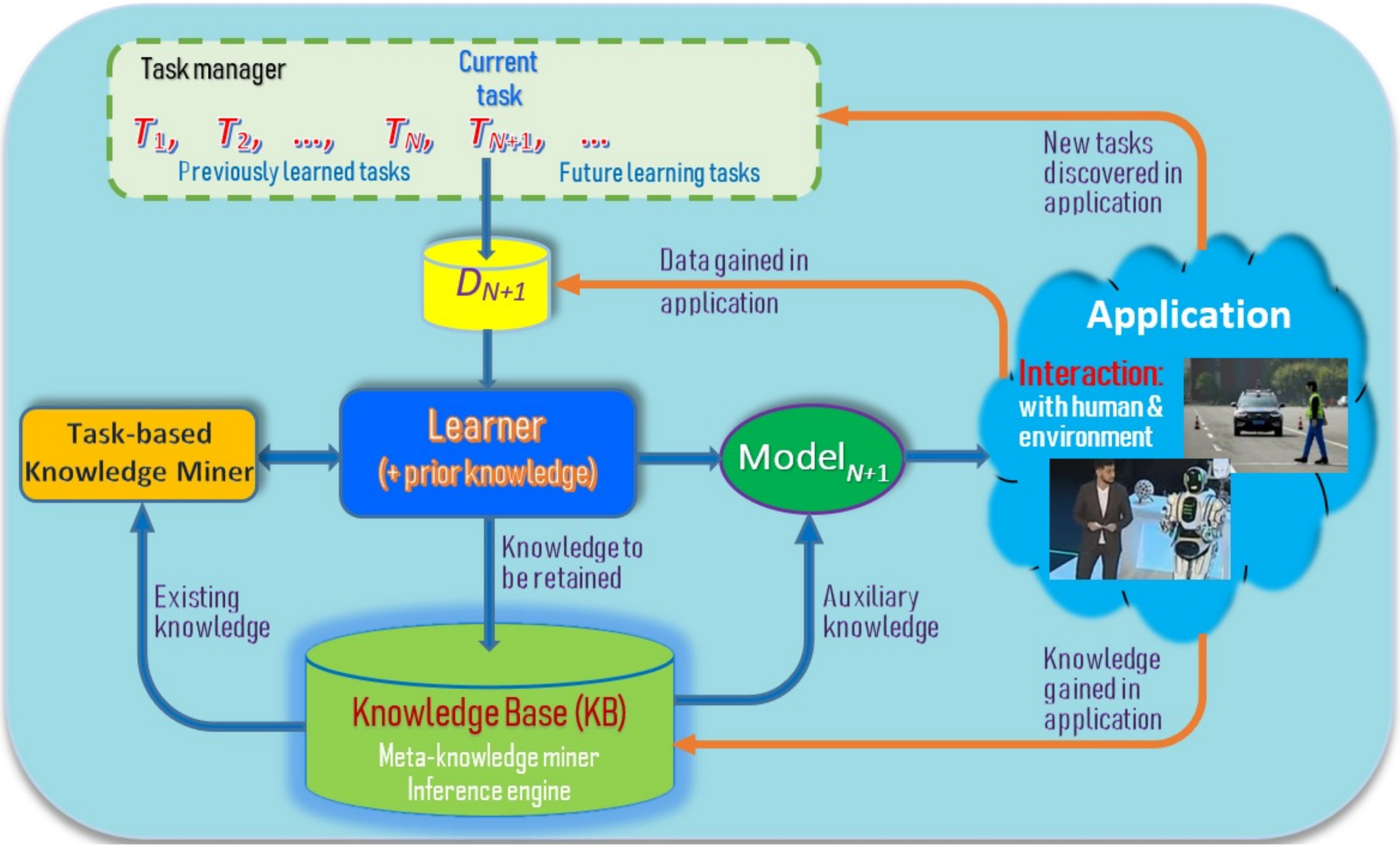
(Thrun 1996, Silver et al 2013; Ruvolo and Eaton, 2013; Chen and Liu, 2016, 2018)

- Learn a sequence of tasks, $T_1$, $T_2$, …, $T_N$, … incrementally. Each task $t$ has a training dataset $\mathcal{D}_t = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ .
- Goal: learn each new task $T_{N+1}$ incrementally
  1. with no catastrophic forgetting: Learning of the new task $T_{N+1}$ should not result in degradation of accuracy for previous $N$ tasks.
  2. with knowledge transfer: leveraging the knowledge learned from previous $N$ tasks to learn the new task $T_{N+1}$ better.
- Question: Where do the task $T_{N+1}$ and its training data $D_{N+1}$ come from?
  - Currently, they are given by the user.

- Thrun. Is learning the *n*-th thing any easier than learning the first? NIPS, 1996.
- Chen and Liu. *Lifelong machine learning*. Morgan & Claypool. 2016, 2018

# Continual learning with learning after deployment

(Chen & Liu, 2018, Liu, 2020, Liu & Mazumder 2021)




**Orange lines:**

Learning after model deployment

(Learning on the job)

(more details)

Liu, Robertson, Grigsby, and Mazumder. ***Self-Initiated Open World Learning for Autonomous AI Agents***. AAAI Spring Symposium, 2022.

- Liu. *Learning on the Job: Online Lifelong and Continual Learning.* AAAI – 2020
- Liu and Mazumder. *Lifelong and Continual Learning Dialogue Systems: Learning during Conversation.* AAAI-2021

# Characteristics of continual learning

(Chen and Liu, 2018, Liu, 2020)

- **Continuous incremental learning process** (no forgetting)
  - ✓ *Without forgetting*: Learning a new task should not forget the past.
- **Knowledge accumulation in KB** (long-term memory)
- **Knowledge transfer/adaptation** (across tasks) (Ke, Liu, Huang, 2020)
  - ✓ Using/adapting past knowledge to help learn new tasks
- **Learning after deployment** **(on the job)**. *Self-supervision* learning using the *accumulated knowledge* and *interaction* with **humans** & **environment.**

---

- Chen and Liu. *Lifelong machine learning*. Morgan & Claypool. 2016, 2018
- Liu. *Learning on the Job: Online Lifelong and Continual Learning*. AAAI – 2020
- Ke, Liu, and Huang. Continual Learning of a Mixed Sequence of Similar and Dissimilar Tasks. *NeurIPS 2020.*

# Closed-world assumption and open-world

(Fei et al, 2016; Shu et al., 2017)

- Supervised learning:
  - Training data: $D^{train} = \{D_1, D_2, \ldots, D_t\}$ of class labels $Y^{train} = \{l_1, l_2, \ldots, l_t\}$.
  - Test data: $D^{test}, Y^{test} \in \{l_1, l_2, \ldots, l_t\}$
- Classic paradigm: closed-world assumption: $Y^{test} \subseteq Y^{train}$
  - Classes appeared in testing must have appeared in training, **nothing new**
- Open-world (with out-of-distribution data, **OOD**) $Y^{test} - Y^{train} \neq \phi$
  - Test data: $D^{test}, Y^{test} \in \{l_1, l_2, \ldots, l_t, L_0\}$
  - $L_0$: novel or unseen classes
  - A system **unable to identify anything new/novel** cannot learn by itself
- Novelty is a key motivation for lifelong or continual learning

- Fei, and Liu. Breaking the Closed World Assumption in Text Classification. *NAACL-HLT 2016*
- Fei, Wang, and Liu. Learning Cumulatively to Become More Knowledgeable. KDD-2016
- *Shu, Hu and Liu. DOC: Deep Open Classification of Text Documents. EMNLP 2017*

# On the job continual learning (CL): Main steps

(Chen and Liu, 2018, Liu, 2020)

- Identify a new task to learn (tasks not given)
  - **Discover novel instances (OOD):** e.g., classify instances in $D^{test}$ to $Y^{train}$ and **detect** *novel instances* $D^{novel} \subseteq D^{test}$ belonging to unknown classes $\boldsymbol{L_0}$
  - **Identify the unseen/new classes** in $D^{novel}$, $\boldsymbol{L_0} = \{l_{t+1}, l_{t+2}, \ldots\}$
  - **Create a task with classes in** $\boldsymbol{L_0} = \{l_{t+1}, l_{t+2}, \ldots\}$ to be learned.
- Acquire ground-truth training data (training data not given)
  - Gather additional ground-truth data if needed
- Learn the task incrementally (one-shot or few-shot CL)

# Lifelong Interactive Learning in Conversation (LINC)

(Liu and Mazumder 2021)

- The tasks have to be **self-discovered on-the-fly** and the **training data has to be found by the agent**.
  - ✓ A **new learning task $T_{N+1}$ is formed** when the agent **needs to** or **can learn a piece of knowledge** or **encounters a problem** in conversation.
    - ✓ Could not understand a user utterance or could not answer a user question
- In order to learn the new task $T_{N+1}$, it needs to **formulate a plan to obtain the ground truth training** data $D_{N+1}$ **on the job.**
  - ✓ That is, to **interact with the user and ask the user questions** and learn from it.
  - ✓ This learning process is **like human on-the-job learning**.

- Liu and Mazumder. *Lifelong and Continual Learning Dialogue Systems: Learning during Conversation.* AAAI-2021

# Example - a learning greeting bot in a hotel
(Chen and Liu 2018)

- See an existing/known guest.
  - Bot: “Hello John, how are you today?”
- See a new guest. Bot recognizes the guest as new
  - Bot: “Welcome to our hotel! What is your name, sir?”
  - Guest: “David”
  - Bot learns to recognize David automatically
    - take pictures of David, and
    - learn to recognize David
- See David next time.
  - Bot: “Hello David, how are you today?”

- Chen and Liu. Lifelong machine learning. Morgan & Claypool. 2018

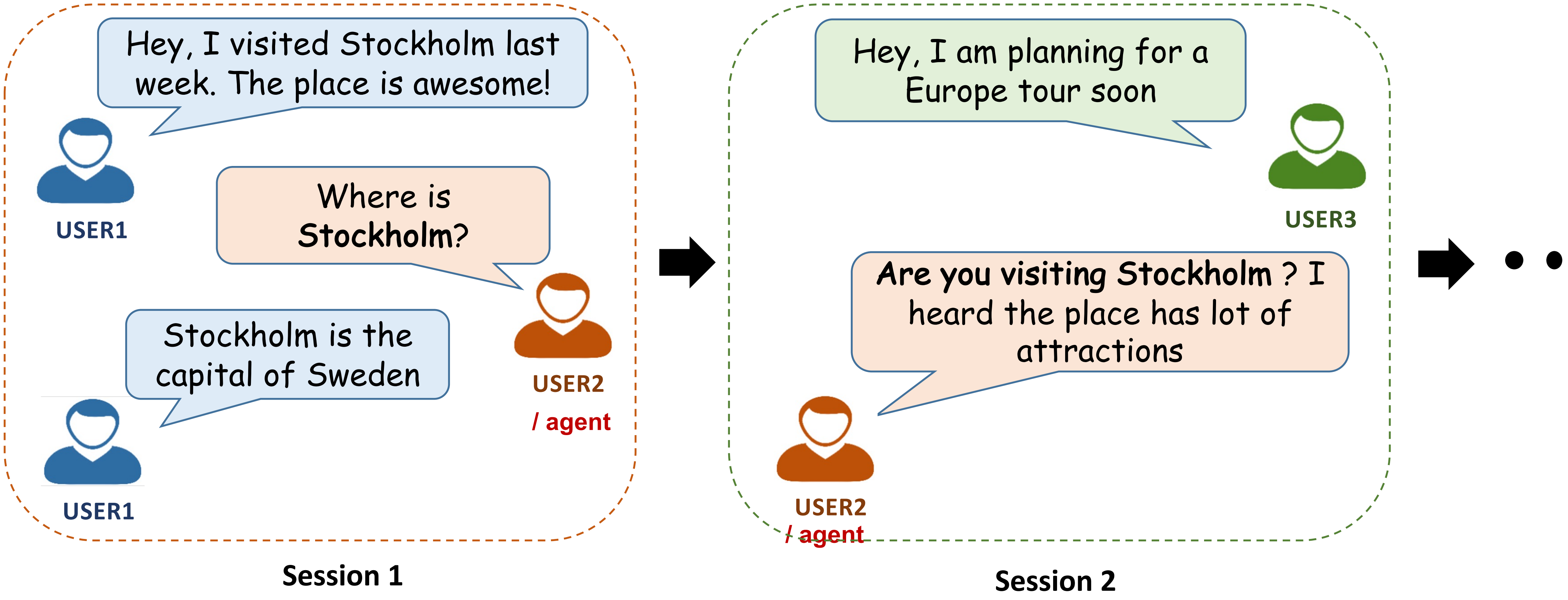
# Example - a learning greeting bot in a hotel
(Chen and Liu 2018)

- See an existing/known guest.
  - Bot: “Hello John, how are you today?”
- See a new guest. Bot recognizes the guest as new.(**create a new task**)
  - Bot: “Welcome to our hotel! What is your name, sir?” (**ask for class label**)
  - Guest: “David” (**got class label: David**)
  - Bot learns to recognize David automatically
    - take pictures of David, and (**get training data**)
    - learn to recognize David (**learn incrementally**)
- See David next time.
  - Bot: “Hello David, how are you today?” (**use the new knowledge**)

- Chen and Liu. Lifelong machine learning. Morgan & Claypool. 2018

# Example - knowledge learning in conversation

Humans Learn and Leverage Knowledge in Lifelong Manner!




Knowledge learning happens in a multi-user environment

- Liu and Mazumder. *Lifelong and Continual Learning Dialogue Systems: Learning during Conversation.* AAAI-2021

# Outline

# Knowledge learning in chatbots: Opportunities

(Liu and Mazumder 2021)

Did you watch anything yesterday?

Watched Forest Gump. The movie was awesome. Liked Tom Hanks' performance a lot!

**Extracted Facts:**
(*Forest Gump, is a, movie*)
(*Tom Hanks, acted in, Forest Gump*).

Hey, is there any good place around for having sushi ?

What is sushi ?

Japanese dish.

**Learned new concepts/entities:**
(*Sushi, is , food*)
(*Sushi, has cuisine, Japanese* ).

1 **Extracting facts from user utterances**

2 **Ask questions to learn about unknown entities and concepts.**

3 **Ask and infer new facts in conversation**

✓ When the chat-bot cannot answer an user query, it can **ask for some related supporting facts and then infer the answer.**

- Liu and Mazumder. *Lifelong and Continual Learning Dialogue Systems: Learning during Conversation.* AAAI-2021

# Continuous Factual knowledge learning in dialogues

(Mazumder et. al. 2019; 2020)

- Dialogue systems are increasingly using knowledge bases (KBs) storing factual knowledge to help generate responses.
  - ✓ KBs are inherently incomplete and remain fixed,
  - ✓ which limit dialogue systems' conversation capability
- CILK: ***Continuous and Interactive Learning of Knowledge*** (Mazumder et. al. 2019); IKAI : **Interactive Knowledge Acquisition and Inference** (Mazumder et. al. 2020)
  - ✓ to continuously and interactively learn and infer new knowledge during conversations


- Mazumder, Liu, Wang, and Ma. Lifelong and Interactive Learning of Factual Knowledge in Dialogues. SIGDIAL-2019
- Mazumder, Liu, Ma and Wang, Continuous and Interactive Factual Knowledge Learning in Verification Dialogues. HAMLETS - NeurIPS-2020 Workshop, 2020.

# Two types of queries or questions
(Mazumder et. al. 2019; 2020)

- Wh-question
  - ✓ E.g., Where was Obama born?
  - ✓ (Obama, bornIn, s?)

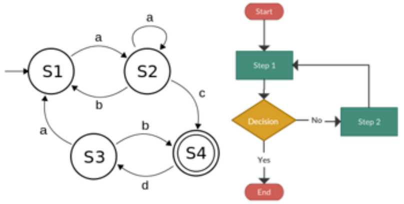
- Fact verification question
  - ✓ Was Obama born in Hawaii?
  - ✓ (Obama, bornIn? Hawaii)


- Mazumder, Liu, Wang, and Ma. Lifelong and Interactive Learning of Factual Knowledge in Dialogues. SIGDIAL-2019
- Mazumder, Liu, Ma and Wang, Continuous and Interactive Factual Knowledge Learning in Verification Dialogues. HAMLETS - NeurIPS-2020 Workshop, 2020.

# Components for knowledge learning

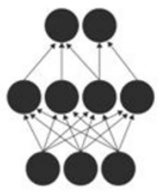
***Knowledge Base***

$\mathcal{K}$




***Interaction Module***

$\mathcal{I}$



***Inference Module***

$\mathcal{M}$

| Stores acquired Facts (Triples) | Interacts with user to acquire Facts | Infers new Knowledge to answer user's query |
|---|---|---|

**KB: Collection of Triples**

$$\mathcal{T}=\{\ (h, r, t) \mid h, t \in E, r \in R\ \}$$

***Triple*** ***Entity Set*** ***Relation Set***

**Triple Store**

**(Boston, *LocatedInCountry*, USA)**

*head* *relation* *tail*

**or**

Boston $\xrightarrow{\textit{LocatedInCountry}}$ USA

**Knowledge Graph**

- decides whether to ask or not, and formulates questions to ask the user for supporting facts

- Performs inference over the acquired Facts and existing KB

- Mazumder, Liu, Wang, and Ma. Lifelong and Interactive Learning of Factual Knowledge in Dialogues. SIGDIAL-2019
- Mazumder, Liu, Ma and Wang, Continuous and Interactive Factual Knowledge Learning in Verification Dialogues. HAMLETS - NeurIPS-2020 Workshop, 2020.

# Assumptions – Knowledge learning in dialogues
(Mazumder et. al. 2019; 2020)

- Focus on the core interactive knowledge learning framework
  - **Do not build all peripheral components** (like fact or relation extraction, entity linking, etc.) which are assumed to be available for use.
- We also assume that the user has good intentions
  - **User answers questions with 100% conformity** about the veracity of his/her facts (more discussion later)
- User is NOT omniscient
  - **We do not assume that the user can answer all questions**
    - Opposed to the teacher-student setup - the teacher is assumed to know everything.


- Mazumder, Liu, Wang, and Ma. Lifelong and Interactive Learning of Factual Knowledge in Dialogues. SIGDIAL-2019
- Mazumder, Liu, Ma and Wang, Continuous and Interactive Factual Knowledge Learning in Verification Dialogues. HAMLETS - NeurIPS-2020 Workshop, 2020.

# (1) When the user asks a Wh-question
(Mazumder et. al. 2019)

- Given a user query **(h, r, ?) [or (?, r, t)]**, the goal is two-fold:
  1. **Answering** the user query or **rejecting** the query to remain unanswered if the correct answer is believed to not exist in the KB
  2. **learning / acquiring** some knowledge (supporting facts) from the user to help the answering task.

- We further distinguish two types of queries:
  (1) **Closed-world Queries**: h (or t) and r are **known** to the KB
  (2) **Open-world Queries**: Either one or both h (or t) and r are **unknown**

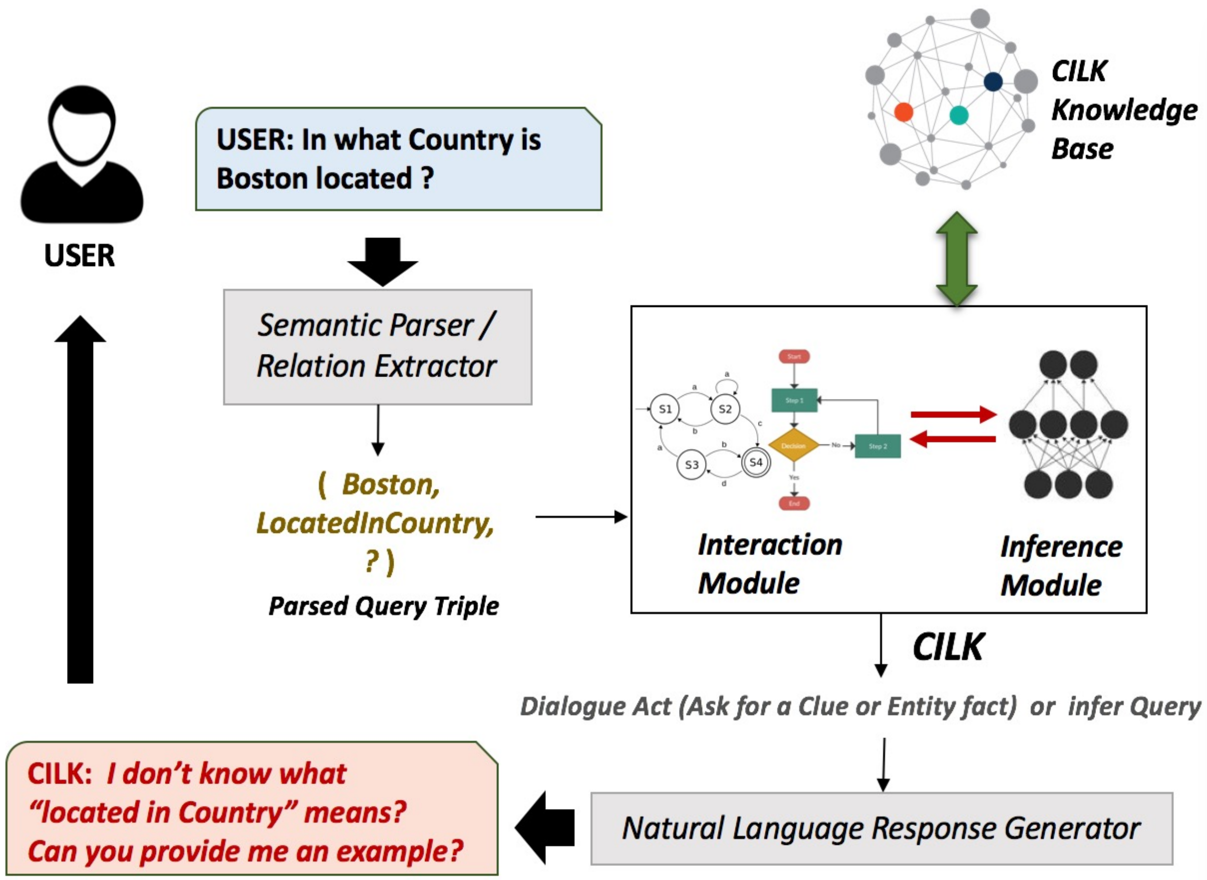
⬇ **Proposed Soln.**

an engine for Continuous and Interactive Learning of Knowledge (CILK)

- Mazumder, Liu, Wang, and Ma. Lifelong and Interactive Learning of Factual Knowledge in Dialogues. SIGDIAL-2019

# Interactive knowledge learning in dialogue: example

## (Mazumder et al. 2019)




**USER:** (*Boston*, *LocatedInCountry*, *?*) "In what Country is Boston located?" [**Query**]

**CILK:** I do not know what "**located in Country**" means? Can you provide me an example? [*Ask for Clue*]

**USER:** (*London*, *LocatedInCountry*, *UK*). "London is located in UK." [**SF1**]

**CILK:** Got it. Can you tell me a fact about "**Boston**"? [*Ask for Entity Fact*]

**USER:** (*Harvard University*, *UniversityLocatedIn*, *Boston*). "Harvard university is located in Boston." [**SF2**]

**CILK:** (*Boston*, *LocatedInCountry*, *USA*) "Boston is located in USA." [**Answer**]

- Mazumder, Liu, Wang, and Ma. Lifelong and Interactive Learning of Factual Knowledge in Dialogues. SIGDIAL-2019

# CILK: The Inference Module

- We use the neural **knowledge base embedding (KBE)** [Bordes et al., 2011,2013; Yang et al., 2014] for learning $\mathcal{M}$

- For evaluation, we adopt DistMult [Yang et al., 2014]

Triple: $(h,\ r,\ t)$

$\mathbf{x}_h$, $\mathbf{x}_t$ and $\mathbf{x}_r$

*One-hot encoding Layer*

Entity Embed.

$$\begin{cases} \mathbf{v}_h = W_E\, \mathbf{x}_h \\ \mathbf{v}_t = W_E\, \mathbf{x}_t \end{cases}$$

Relation Embed.

$$\mathbf{v}_r = W_R\, \mathbf{x}_r$$

*Embedding Layer*

DisMult Scoring Function

$$S(h,r,t) = \mathbf{v}_h^T diag(\mathbf{v}_r)\mathbf{v}_t = \sum_{i=1}^{N} \mathbf{v_h}[i]\mathbf{v}_r[i]\mathbf{v}_t[i]$$

*Scoring Layer*

Max-margin Raking Loss

$$\mathcal{L} = \sum_{d \in D^+} \sum_{d' \in D^-} max\{S(d') - S(d) + 1, 0\}$$




- Bordes, Weston, Collobert, Bengio. **Learning structured embeddings of knowledge bases**. AAAI 2011
- Bordes, Usunier, Garcia-Duran, Weston, Yakhnenko. **Translating embeddings for modeling multi-relational data**. NIPS, 2013
- Yang, Yih, He, Gao, Deng. ***Embedding entities and relations for learning and inference in knowledge bases***. ICLR 2014

# Rejection in KB Inference

CILK maintains a **threshold buffer** $\mathcal{T}$ that stores entity and relation specific prediction thresholds and updates it continuously over time.

**mean scores of triples involving +ve entities**

$$\mathcal{T}[z] = \frac{1}{2\left|\boxed{D^z_{vd}}\right|} \sum_{(q, E^+, E^-) \in D^z_{vd}} \mu^+_E + \mu^-_E$$

**entity e or relation r**

**validation query tuple set involving z**

**mean scores of triples involving -ve entities**

$$\mu^+_E = \frac{1}{|E^+|} \sum_{e^+_i \in E^+} S(q, e^+_i) \qquad \mu^-_E = \frac{1}{|E^-|} \sum_{e^-_i \in E^-} S(q, e^-_i)$$

$S(e, r, e^+_i)$ **For tail query triple**

$S(e^+_i, r, e)$ **For head query triple**

- Mazumder, Liu, Wang, and Ma. Lifelong and Interactive Learning of Factual Knowledge in Dialogues. SIGDIAL-2019

# Designing the Interaction Module : CILK's Interaction Strategy

- CILK has to acquire supporting facts **to learn embeddings of e and r**

  - user **can only provide very few supporting facts per session** → **may not be sufficient** for learning good embeddings of *e* and *r*

  - Asking for too many SFs can be **annoying** and also, is **unnecessary** for entity and/or relation with good emmbeddings.

  - Need a **sufficiently good validation set for learning** $\mathcal{T}[e]$ **and** $\mathcal{T}[r]$

⬇

Ask for SFs for the **known entities and/or relations for which CILK is not confident enough**, besides the unknown ones.

▪ Mazumder, Liu, Wang, and Ma. Lifelong and Interactive Learning of Factual Knowledge in Dialogues. SIGDIAL-2019

# Acquiring Knowledge with limited Interaction: Improving Skillset over time

- A **performance buffer** $\mathcal{P}$ is used to store the performance statistics of $\mathcal{M}$

  $\mathcal{P}[e]$ and $\mathcal{P}[r]$ denote the **MRR achieved by** $\mathcal{M}$ while answering queries involving $e$ and $r$ respectively, *evaluated on the validation dataset* $D_{vd}$

- At the end of each dialogue session, CILK **detects bottom** $\rho\%$ **of query relations and entities** in $\mathcal{P}$ based on MRR scores.
  - ***diffident relation and entity sets*** *for the next dialogue session*.

⬇ **Strategy**

Ask for supporting facts for **diffident** and/or **unknown** query relation or entity

- Mazumder, Liu, Wang, and Ma. Lifelong and Interactive Learning of Factual Knowledge in Dialogues. SIGDIAL-2019

# CILK : Performance

(# clues, # entity facts) acquired per session

Performance Buffer disabled

| (#C, #EF) | WordNet | | | Nell | | |
|---|---|---|---|---|---|---|
| | MRR | H@1 | H@10 | MRR | H@1 | H@10 |
| (1, 1) | 0.30 | 22.09 | 37.83 | 0.23 | 16.89 | 31.14 |
| (1, 2) | 0.32 | 23.00 | 39.25 | **0.25** | **18.11** | **31.30** |
| (1, 3) | **0.33** | **25.27** | **40.95** | 0.23 | 17.16 | 30.03 |
| (1, 3)-U | 0.31 | 23.52 | 38.15 | 0.21 | 15.77 | 28.64 |
| (2, 2) | 0.32 | 23.43 | 39.05 | 0.23 | 16.82 | 30.33 |

**Performance improves with the increase in (acquired) entity fact triples (specially, for WordNet).**

- Mazumder, Liu, Wang, and Ma. Lifelong and Interactive Learning of Factual Knowledge in Dialogues. SIGDIAL-2019

# (2) When the user asks a fact-verification question
(Mazumder et. al. 2020)

Learn new knowledge interactively from user when the system is unable to answer a user's factual verification (yes/no) question.

| | | |
|---|---|---|
| **USER:** | (*Obama, CitizenOf?, USA*) "Is Obama a citizen of USA?" | [**Query**] |
| **IKAI:** | I do not know what "**CitizenOf**" means? Can you provide me an example? | [***Ask for Clue***] |
| **USER:** | (*David Cameron, CitizenOf, UK*). "David Cameron is a citizen of UK." | [**SF1**] |
| **IKAI:** | Got it. Can you tell me how "**Obama**" and "**Honolulu**" are related? | [***CLQ***] |
| **USER:** | (*Obama, BornIn, Honolulu*). "Obama was born in Honolulu." | [**SF2**] |
| **IKAI:** | Got it. Can you tell also me how "**Honolulu**" and "**Hawaii**" are related? | [***MLQ***] |
| **USER:** | (*Honolulu, CapitalOfState, Hawaii*). "Honolulu is the state capital of Hawaii." | [**SF3**] |
| **IKAI:** | (*Obama, CitizenOf?, USA*) holds. "Yes, Obama is a US citizen." | [**Answer**] |

- Mazumder, Liu, Ma and Wang, Continuous and Interactive Factual Knowledge Learning in Verification Dialogues. HAMLETS - NeurIPS-2020 Workshop, 2020.

# The IKAI approach

(Mazumder et. al. 2020)

- User question (h, r?, s)
- Step 1 - Interactive acquisition of supporting facts:
  - If **r is unknown**, IKAI asks the user to **provide a clue** [an example triple r]
  - If **s or t is unknown**, IKAI asks the user to **provide a link/relation to connect** the **unknown entity** s or t with an automatically selected existing entity
- Step 2 - Knowledge inference (Infer the query answer):
  - Uses a **path-ranking algorithm** C-PR (Mazumder and Liu 2017) to build a predictive model (Predictor) to predict whether (s, r?, t) is true.
    - Enumerate relation paths between two entities (s, t) in a KB (encoded as a multi-relation graph) and use those paths as features to train the predictor.



- Mazumder, Liu, Ma and Wang, Continuous and Interactive Factual Knowledge Learning in Verification Dialogues. HAMLETS - NeurIPS-2020 Workshop, 2020.

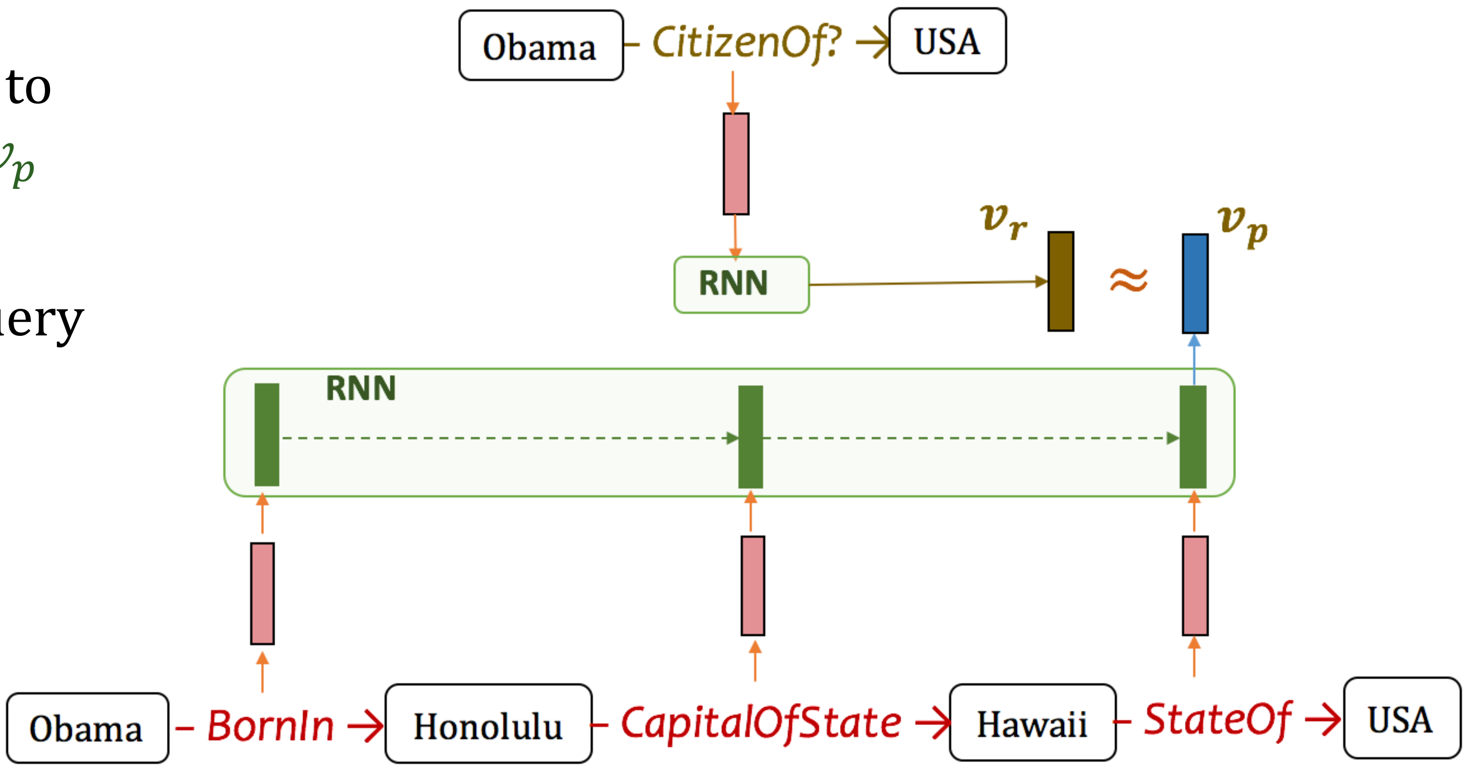
# C-PR + Compositional Vector Space*: Inference Module of IKAI

*(Neelakantan et. al. 2015)

**Query**: *CitizenOf* (*Obama*, *USA*)?

- Encodes the path feature $p \in P_c$ enumerated by C-PR using RNN to learn a vector representation of $v_p$
- Uses same RNN to encode the query relation $r$ as $v_r$
- Inference -

$$\mathbb{P}(r|s,t) = \mathrm{sigmoid}\left(\frac{1}{|P_c|}\sum_{p \in P_c} \cos(v_r, v_p)\right).$$






Arvind Neelakantan, Benjamin Roth, and Andrew McCallum. ***Compositional vector space models for knowledge base completion***. In ACL, 2015.

# Finite State Machine : Interaction Module ( $\mathcal{I}$ )

$$(\mathcal{S}, \mathcal{A}, \mathcal{S}_0, \mathcal{S}_F, \Delta)$$

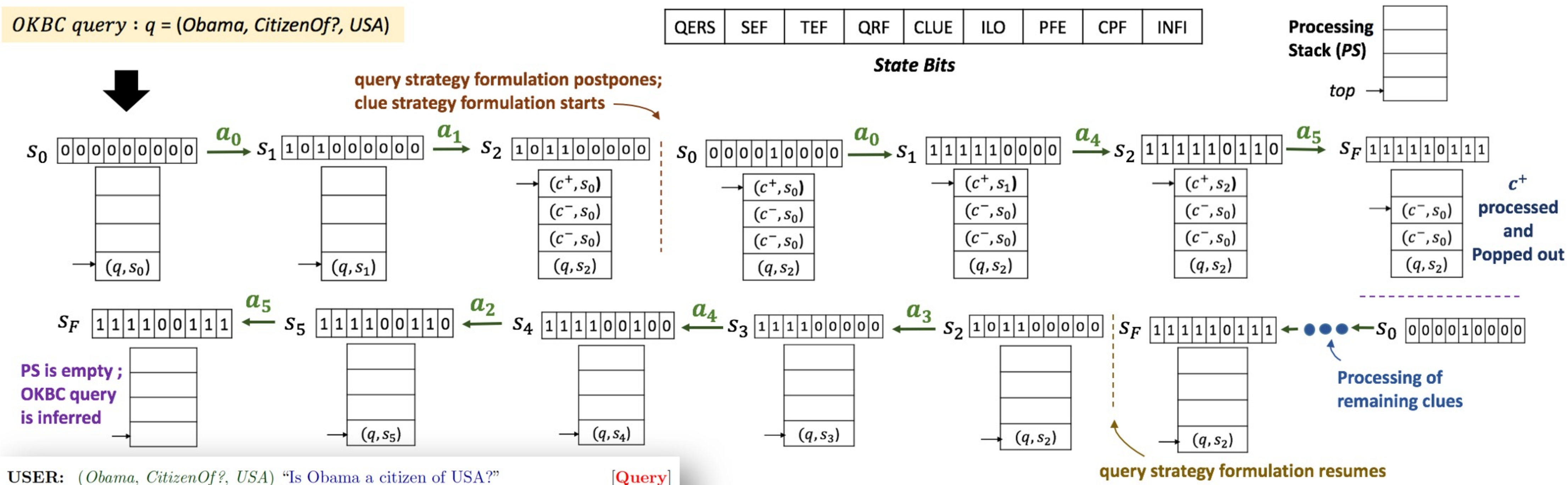
States | Actions | Initial states | Final states | $\Delta : (\mathcal{S}, \mathcal{A}) \rightarrow \mathcal{S}$ Transition Function

| SB | Name | Description |
|---|---|---|
| QERS | Query entities and relation searched | Whether the query source (s) and target (t) entities and query relation (r) have been searched in KB. |
| SEF | Source Entity Found | Whether the source entity (s) has been found in KB. |
| TEF | Target Entity Found | Whether the target entity (t) has been found in KB. |
| QRF | Query Relation Found | Whether the query relation (r) has been found in KB. |
| CLUE | Clue bit set | Whether the triple (to be processed) is a clue from user. |
| ILO | Interaction Limit Over | Whether the interaction limit is over for the query. |
| PFE | Path Feature Extracted | Whether path feature extraction has been done. |
| CPF | Complete Path Found | Whether the extracted path features are complete. |
| INFI | Inference Invoked | Whether inference module has been invoked. |

| State Transition Conditions ( for current state bits $S_j$ [.] ) | Action Id : Operation |
|---|---|
| $QERS = 0$ | $a_0$ : Search source (s), target (t) entities and query relation (r) in KB. |
| $ILO = 0 \wedge CLUE = 0 \wedge QERS = 1 \wedge QRF = 0$ | $a_1$ : Ask user to provide an clue/example for query relation r. |
| $PFE = 1 \wedge ILO = 0 \wedge CPF = 0$ | $a_2$ : Ask user to provide a missing link for path feature completion. |
| $QERS = 1 \wedge ( SEF = 0 \vee TEF = 0 ) \wedge ILO = 0$ | $a_3$ : Ask user to provide a connecting link to add a new entity to the KB. |
| $QERS = 1 \wedge PFE = 0 \wedge SEF = 1 \wedge TFE = 1$ | $a_4$ : Extract path features between source (s) and target (t) entities using C-PR. |
| $QRF = 1 \wedge CPF = 1$ | $a_5$ : Invoke prediction model for inference. |

- Mazumder, Liu, Ma and Wang, Continuous and Interactive Factual Knowledge Learning in Verification Dialogues. HAMLETS - NeurIPS-2020 Workshop, 2020.

# Working of IKAI' FSM : a given Session




USER: (*Obama, CitizenOf?, USA*) "Is Obama a citizen of USA?" [Query]

IKAI: I do not know what "**CitizenOf**" means? Can you provide me an example? [*Ask for Clue*]

USER: (*David Cameron, CitizenOf, UK*). "David Cameron is a citizen of UK." [SF1]

IKAI: Got it. Can you tell me how "**Obama**" and "**Honolulu**" are related? [*CLQ*]

USER: (*Obama, BornIn, Honolulu*). "Obama was born in Honolulu." [SF2]

IKAI: Got it. Can you tell also me how "**Honolulu**" and "**Hawaii**" are related? [*MLQ*]

USER: (*Honolulu, CapitalOfState, Hawaii*). "Honolulu is the state capital of Hawaii." [SF3]

IKAI: (*Obama, CitizenOf?, USA*) holds. "Yes, Obama is a US citizen." [Answer]

$a_0$ : Search source ($s$), target ($t$) entities and query relation ($r$) in KB.

$a_1$ : Ask user to provide an clue/example for query relation $r$.

$a_2$ : Ask user to provide a missing link for path feature completion.

$a_3$ : Ask user to provide a connecting link to add a new entity to the KB.

$a_4$ : Extract path features between source ($s$) and target ($t$) entities using C-PR.

$a_5$ : Invoke prediction model for inference.

- Mazumder, Liu, Ma and Wang, Continuous and Interactive Factual Knowledge Learning in Verification Dialogues. HAMLETS - NeurIPS-2020 Workshop, 2020.

# IKAI - Performance Evaluation

Path completion via Blind Guessing

+ve F1 score

*Macro F1 score*

| Dataset | Models | Rel - K / Ent -K | | Rel - K / Ent -UNK | | Rel - UNK / Ent - K | | Rel - UNK / Ent -UNK | | Overall | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | F1(+) | Macro-F1 | F1(+) | Macro-F1 | F1(+) | Macro-F1 | F1(+) | Macro-F1 | F1(+) | Macro-F1 |
| Freebase | BG | 0.584 | 0.629 | 0.494 | 0.569 | 0.432 | 0.532 | 0.388 | 0.501 | 0.508 | 0.579 |
| | w/o PTL | 0.555 | 0.652 | **0.533** | **0.620** | **0.528** | 0.419 | **0.525** | 0.418 | **0.538** | 0.584 |
| | IKAI | **0.587** | **0.671** | 0.493 | 0.591 | 0.525 | **0.616** | 0.440 | **0.577** | 0.532 | **0.627** |
| WordNet | BG | 0.548 | 0.466 | 0.532 | 0.525 | 0.486 | 0.476 | 0.498 | 0.484 | 0.526 | 0.482 |
| | w/o PTL | **0.666** | **0.741** | **0.561** | **0.624** | 0.461 | 0.281 | 0.485 | 0.323 | 0.556 | 0.588 |
| | IKAI | 0.655 | 0.694 | 0.552 | 0.604 | **0.612** | **0.659** | **0.509** | **0.506** | **0.612** | **0.653** |

*Continuous learning past tasks (relations) Is disabled*

*IKAI achieves best performance overall*

*% of test Triples observed*

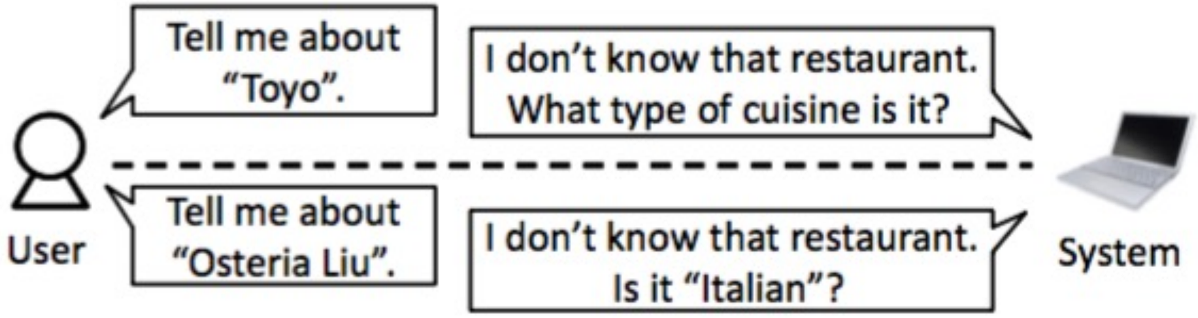
| % TTO | Freebase | | | WordNet | | |
|---|---|---|---|---|---|---|
| | *kwn* [9] | *unk* [293] | *all* [302] | *kwn* [21] | *unk* [105] | *all* [126] |
| 50% | 0.0 | 0.492 | 0.507 | 0.947 | 0.799 | 0.819 |
| 100% | **0.545** | **0.580** | **0.578** | **0.950** | **0.870** | **0.884** |

IKAI Performance improvement due to User Interaction

- Mazumder, Liu, Ma and Wang, Continuous and Interactive Factual Knowledge Learning in Verification Dialogues. HAMLETS - NeurIPS-2020 Workshop, 2020.

# Lexical knowledge acquisition in dialogues
(Otsuka et. al. 2013)

- Goal: acquire the **attributes of unknown concepts** from users during dialogues



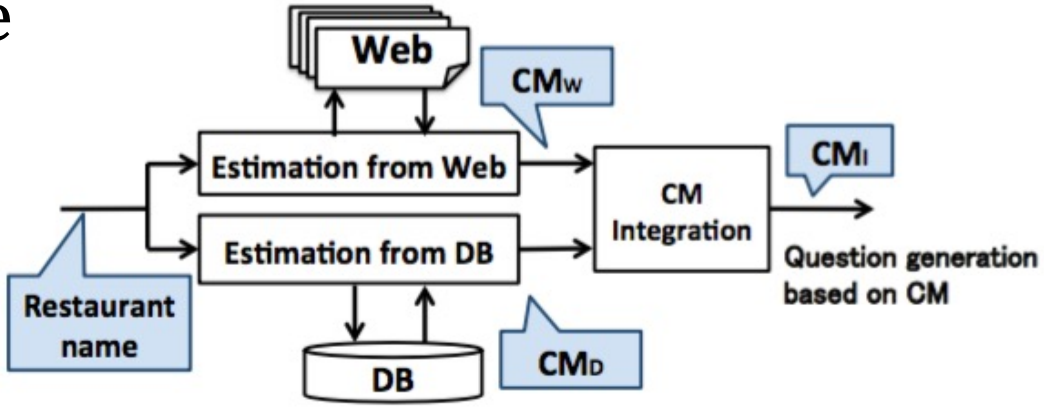

- propose a method for **generating more specific questions** than simple wh-questions to acquire the attributes
  - **Task: estimate the cuisine type of a restaurant from its name**, which is assumed to be unknown to the system
  - well-distributed **confidence measure** (CM) on the attributes to generate more specific questions.
  - Two basic CMs : (1) character and word distributions in the target database and (2) frequency of occurrence of restaurant attributes on Web pages.



- Otsuka et. al. Generating More Specific Questions for Acquiring Attributes of Unknown Concepts from Users. SIGDIAL 2013

# Question Generation
(Otsuka et. al. 2013)

- Determines a question type on the basis of CM.
  - The CM is estimated for each cuisine type $c_j$ in the target database.




$$num = min(n) \text{ s.t. } \sum_{j=1}^{n} CM(c_j) > \theta,$$

$CM(c_j)$ is a confidence measure for cuisine type $c_j$ its descending order

| $num$ | Question form | Example |
|---|---|---|
| 1 | Yes-No question | Is it cuisine $c_1$? |
| 2 | Alternative question | Which cuisine is it, $c_1$ or $c_2$? |
| 3 | 3-choice question | Which cuisine is it, $c_1$, $c_2$, or $c_3$? |
| ≥4 | Wh-question | What cuisine is it? |



- Otsuka et. al. Generating More Specific Questions for Acquiring Attributes of Unknown Concepts from Users. SIGDIAL 2013

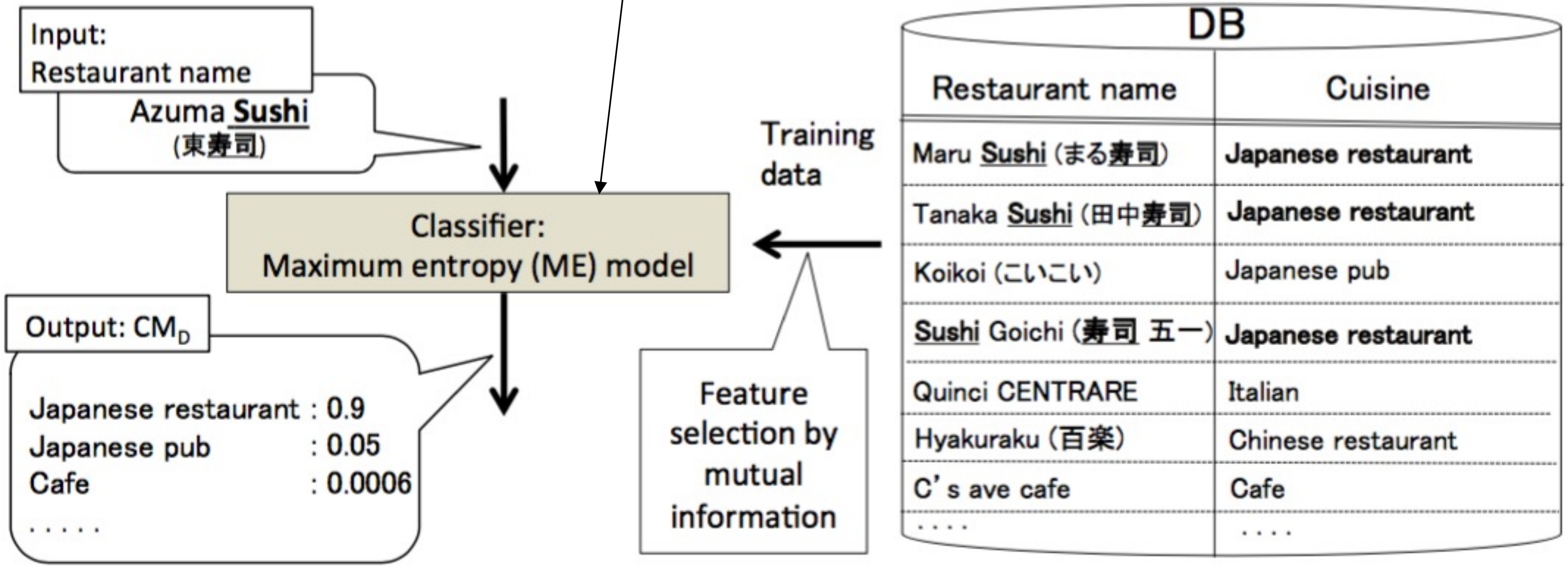
# $CM_D$ Calculation: using Word and Character Distribution in Database (Otsuka et. al. 2013)

cuisine type

restaurant name

feature vector (n-gram, words etc.) obtained from a restaurant name

$$CM_D(s_i, c_j) = p(c_j|s_i) = \frac{1}{Z}\exp\left[\vec{\lambda}(c_j)\cdot\vec{\phi}(s_i)\right]$$

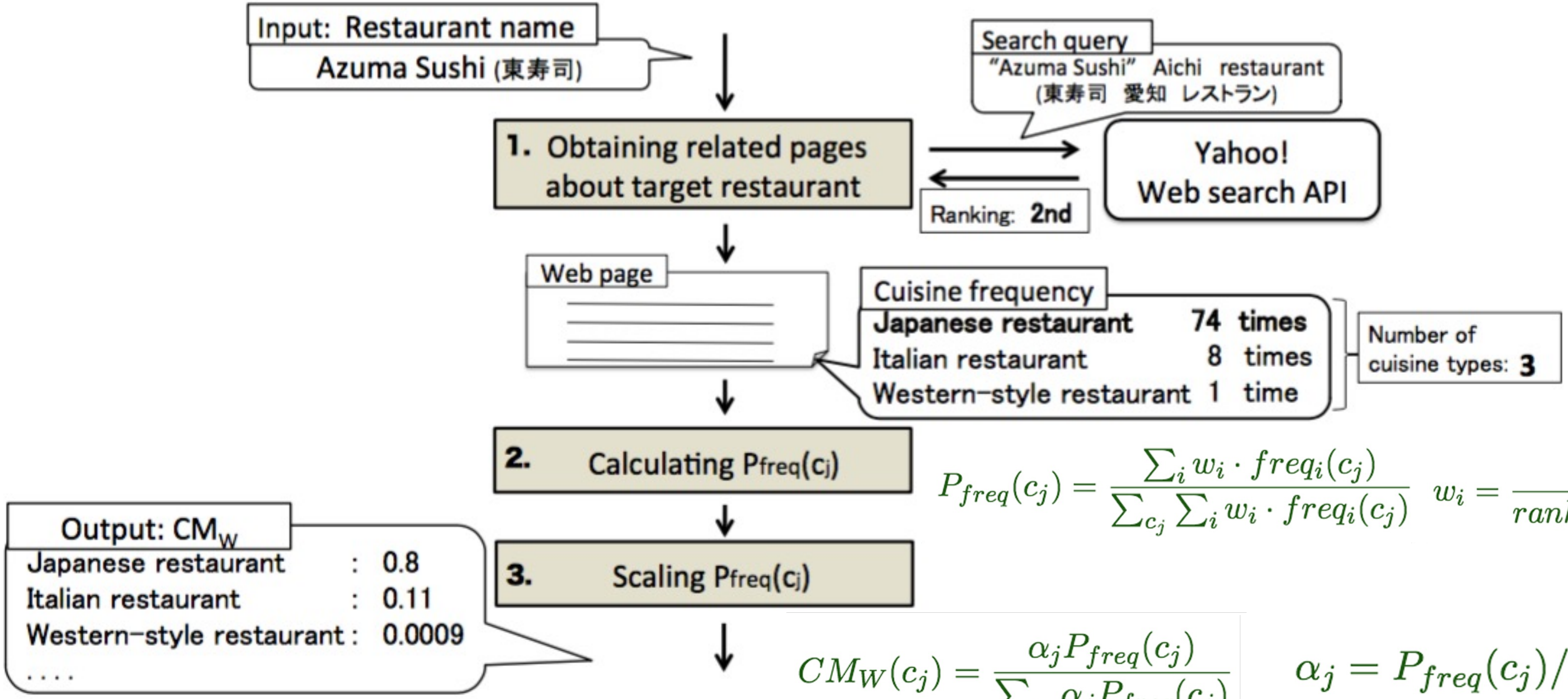





- Otsuka et. al. Generating More Specific Questions for Acquiring Attributes of Unknown Concepts from Users. SIGDIAL 2013

# $CM_W$ Calculation: Using the Web

(Otsuka et. al. 2013)




$$P_{freq}(c_j) = \frac{\sum_i w_i \cdot freq_i(c_j)}{\sum_{c_j} \sum_i w_i \cdot freq_i(c_j)} \quad w_i = \frac{1}{rank(i) \cdot cuisine(i)}$$

$$CM_W(c_j) = \frac{\alpha_j P_{freq}(c_j)}{\sum_{c_j} \alpha_j P_{freq}(c_j)} \quad \alpha_j = P_{freq}(c_j)/P_{freq}(c_1)$$



- Otsuka et. al. Generating More Specific Questions for Acquiring Attributes of Unknown Concepts from Users. SIGDIAL 2013

# Lexical acquisition in dialogues: Performance
(Otsuka et. al. 2013)

*Distribution of estimation results by CM values*

| | $CM_D$ | | $CM_W$ | | $CM_I$ | |
|---|---|---|---|---|---|---|
| $CM$ range | Correct | Incorrect | Correct | Incorrect | Correct | Incorrect |
| 0.0 – 0.1 | 0 | 0 | 0 | 32 | 2 | 10 |
| 0.1 – 0.2 | 0 | 0 | 0 | 11 | 9 | 15 |
| 0.2 – 0.3 | 1 | 16 | 14 | 22 | 15 | 18 |
| 0.3 – 0.4 | 6 | 19 | 28 | 19 | 10 | 8 |
| 0.4 – 0.5 | 11 | 25 | 29 | 21 | 13 | 12 |
| 0.5 – 0.6 | 21 | 29 | 56 | 9 | 13 | 12 |
| 0.6 – 0.7 | 22 | 28 | 85 | 7 | 15 | 7 |
| 0.7 – 0.8 | 41 | 16 | 42 | 3 | 17 | 6 |
| 0.8 – 0.9 | 21 | 9 | 19 | 1 | 19 | 9 |
| 0.9 – 1.0 | 131 | 4 | 1 | 1 | 184 | 10 |
| Total | 254 | 146 | 274 | 124 | 297 | 103 |

- Otsuka et. al. Generating More Specific Questions for Acquiring Attributes of Unknown Concepts from Users. SIGDIAL 2013

# Knowledge acquisition in a rule-based system
(Liu and Mei, 2020)

- Many existing chatbots are written mainly with rules.
- We learn with ***knowledge distillation pattern***: (*p*, *F*, *B*),
  - *where p*: a dialogue pattern; *F*: implied facts; *B*: implied beliefs.
  - E.g., ***p***: [* stayed in X at Y]   ***F***: {(X, is-a, hotel)}   ***B***: {(X, has-address, Y)}
- If user says, "*I stayed in Hilton at 150 Pine Street,*" the agent gets
  - a fact: (Hilton, is-a, hotel) & a belief: (Hilton, has-address, 150 Pine Street).
- If user says, "*I stayed in Hilton,*" the agent gets
  - A fact: (Hilton, is-a, hotel) and also
  - a question to ask the user: "What is the address of the hotel?"
- With such patterns, a great deal of knowledge can be learned from end-users

- Liu and Mei. Lifelong Knowledge Learning in Rule-based Dialogue Systems. arXiv:2011.09811, 2020.

# Outline

I. Lifelong and Continual Learning: An Introduction
II. Continuous Knowledge Learning during Conversation
III. **Continual Language Learning and Grounding**
IV. Open-Domain Dialogue Learning
V. Continual Learning for Task-oriented Dialogue Systems
VI. Continual Learning of Conversational Skills
VII. Other Challenges & Summary

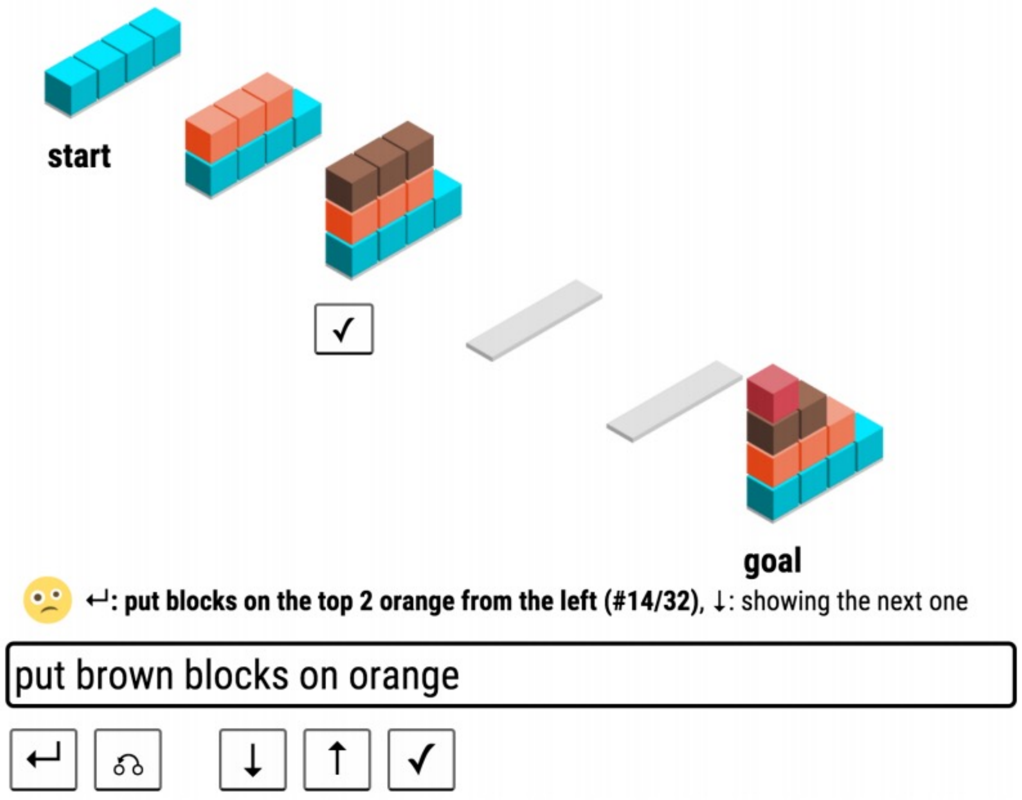
# Learning to ground natural language (NL) commands

- Task-oriented chatbots like virtual assistants (e.g., Siri, Alexa, etc) are Natural Language (command) Interfaces (NLI)
  - ✓ allow users to issue natural language (NL) commands to be mapped to some actions for execution by the underlying application.

- Interactive Language Learning after deployment:
  - ✓ via user demonstrations (Wang et. al. 2016).
  - ✓ via multi-turn NL dialogues with the user (Mazumder et. al. 2020)

# Learning Language Games through Interaction
(Wang et. al. 2016)

- A language learning setting for building adaptive natural language interfaces.
  - inspired by Wittgenstein's language games:
    - a **human wishes to accomplish some task** but **can only communicate with a computer**, who performs the actual actions.
    - The **computer initially knows nothing** about language and therefore **must learn it from scratch through interaction**, while the human adapts to the computer's capabilities

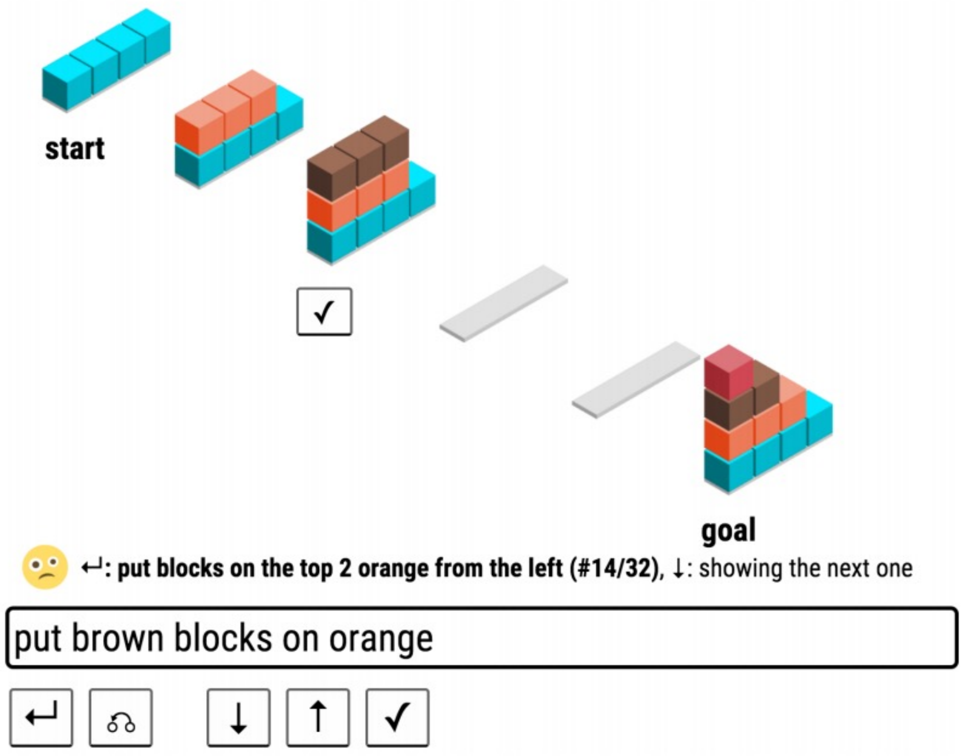



The SHRDLURN game

- Wang, Liang, Manning. Learning Language Games through Interaction. ACL 2016

# Interactive learning through language game (ILLG) Setting (Wang et. al. 2016)

- Goal : transform a start state into a goal state, but the only action the human can take is entering an utterance
  - ✓ The **computer parses the utterance** and produces a **ranked list of possible interpretations** based on its current model.
  - ✓ **Human scrolls through the list** and chooses the intended one.
  - ✓ For the computer to be successful, it **has to learn the human's language quickly over the course of the game**.




The SHRDLURN game

- Wang, Liang, Manning. Learning Language Games through Interaction. ACL 2016

# Compositional action space for SHRDLURN
(Wang et. al. 2016)

| Rule | Semantics | Description |
|---|---|---|
| Set | `all()` | all stacks |
| Color | `cyan\|brown\|red\|orange` | primitive color |
| Color → Set | `with`($c$) | stacks whose top block has color $c$ |
| Set → Set | `not`($s$) | all stacks except those in $s$ |
| Set → Set | `leftmost\|rightmost`($s$) | leftmost/rightmost stack in $s$ |
| Set Color → Act | `add`($s, c$) | add block with color $c$ on each stack in $s$ |
| Set → Act | `remove`($s$) | remove the topmost block of each stack in $s$ |

*'remove rightmost orange block'*

→ remove(rightmost(with(orange)))

- Wang, Liang, Manning. Learning Language Games through Interaction. ACL 2016

# ILLG as a semantic parser
(Wang et. al. 2016)

- Maps natural language utterances (e.g., *'remove red'*) into logical forms (e.g., remove(with(red))).
  - ✓ Uses a log-linear model over logical forms (actions) $z \in Z$ given utterance $x$:

Prob. Assigned to possible mappings

$$p_\theta(z \mid x) \propto \exp(\theta^\top \boxed{\phi(x, z)})$$

Features defined over utterance and logical form

  - ✓ Parser **generates many candidate logical forms**.
  - ✓ Based on the human's feedback, it **performs online gradient updates** on the parameters corresponding to simple lexical features.
    - ➢ n-grams (including skip-grams) conjoined with tree-grams on the logical form side.



- Wang, Liang, Manning. Learning Language Games through Interaction. ACL 2016

# ILLG: Performance (Wang et. al. 2016)

## Evaluation on 100 players on Mechanical Turk

**Most successful players (1st–20th)**

| | | |
|---|---|---|
| rem cy pos 1, stack or blk pos 4, rem blk pos 2 thru 5, rem blk pos 2 thru 4, stack bn blk pos 1 thru 2, fill bn blk, stack or blk pos 2 thru 6, rem cy blk pos 2 fill rd blk **(3.01)** | remove the brown block, remove all orange blocks, put brown block on orange blocks, put orange blocks on all blocks, put blue block on leftmost blue block in top row **(2.78)** | Remove the center block, Remove the red block, Remove all red blocks, Remove the first orange block, Put a brown block on the first brown block, Add blue block on first blue block **(2.72)** |

**Average players (21th–50th)**

| | | |
|---|---|---|
| reinsert pink, take brown, put in pink, remove two pink from second layer, Add two red to second layer in odd intervals, Add five pink to second layer, Remove one blue and one brown from bottom layer **(9.17)** | remove red, remove 1 red, remove 2 4 orange, add 2 red, add 1 2 3 4 blue, emove 1 3 5 orange, add 2 4 orange, add 2 orange, remove 2 3 brown, add 1 2 3 4 5 red, remove 2 3 4 5 6, remove 2, add 1 2 3 4 6 red **(8.37)** | move second cube, double red with blue, double first red with red, triple second and fourth with orange, add red, remove orange on row two, add blue to column two, add brown on first and third **(7.18)** |

**Least successful players (51th–)**

| | | |
|---|---|---|
| holdleftmost, holdbrown, holdleftmost, blueonblue, brownonblue1, blueonorange, holdblue, holdorange2, blueonred2 , holdends1, holdrightend, hold2, orangeonorangerightmost **(14.15)** | ‘add red cubes on center left, center right, far left and far right’, ‘remove blue blocks on row two column two, row two column four’, remove red blocks in center left and center right on second row **(12.6)** | laugh with me, red blocks with one aqua, aqua red alternate, brown red red orange aqua orange, red brown red brown red brown, space red orange red, second level red space red space red space **(14.32)** |

- Wang, Liang, Manning. Learning Language Games through Interaction. ACL 2016

# ILLG: Performance

(Wang et. al. 2016)

- memorize: featurize entire utterance and logical form non-compositionally;
- half model: featurize the utterances with unigrams, bi-grams, and skip-grams but conjoin with the entire logical form;
- full model: the proposed full model
- full+prag: the proposed full model with online pragmatics algorithm

| | players ranked by # of scrolls | | | |
|---|---|---|---|---|
| Method | top 10 | top 20 | top 50 | all 100 |
| memorize | 25.4 | 24.5 | 22.5 | 17.6 |
| half model | 38.7 | 38.4 | 36.0 | 27.0 |
| half + prag | 43.7 | 42.7 | 39.7 | 29.4 |
| full model | 48.6 | 47.8 | 44.9 | 33.3 |
| full + prag | 52.8 | 49.8 | 45.8 | 33.8 |

Average online accuracy under various settings

- Wang, Liang, Manning. Learning Language Games through Interaction. ACL 2016

# Natural Language to Natural Language (NL2NL) matching
## (Mazumder et. al. 2020)

- Goal: An adaptable system to automatically serve as NLI to API-based applications .
  - ✓ One system for many diverse API-driven applications
  - ✓ Learning continuously or lifelong from users via interactions.
- Approach: Natural Language to Natural Language (NL2NL) matching
  - ✓ Each action (API) is attached with one or more natural language (NL) representation - a set of one or more API seed commands (SCs) just like a NL command from the user to invoke the API.

| API (arg : arg type) | Seed Commands (SCs) | Example NL command |
|---|---|---|
| SwitchOnLight(X1: *location*) | Switch on the light at **X1**; Put on light on **X1** | Power on the light at **bedroom (X1)** |
| SwitchOffLight(X1: *location*) | Switch off the light at X1; Power off the light at X1 | Turn off the light at **living room (X1)** |
| ChangeLightColor (X1: *location*, X2: *color*) | Change the X1 light to X2; I want X1 light as X2 | Please make the color of **bedroom (X1)** light **blue (X2)** |

  - ✓ When the user issues a NL command, the system simply matches the command with one of the system's SCs.



- Mazumder, Liu, Wang, Esmaeilpour. An Application-Independent Approach to Building Task-Oriented Chatbots with Interactive Continual Learning. HLDS Workshop at NeurIPS 2020.

# Command Matching & Learning (CML)

(Mazumder et. al. 2020)

- CML works based on NL2NL matching idea.
- Consists of Three components
  - SC (seed command) specification
    - ✓ enable application developer to specify a set of SCs for each of their APIs
  - Command grounding module
    - ✓ ground a user command *C* to an action SC by matching C with the correct SC (whose associated action API is then executed)
  - Interactive learner
    - ✓ It interacts with end-users in multi-turn dialogues to continually learn new SCs and paraphrases of API argument values.

- Mazumder. On-the-job Continual and Interactive Learning of Factual Knowledge and Language Grounding. PhD diss., UIC, 2021.
- Mazumder, Liu, Wang, Esmaeilpour. An Application-Independent Approach to Building Task-Oriented Chatbots with Interactive Continual Learning. HLDS Workshop at NeurIPS 2020.

# SC Specification (blocks-world)

## (Mazumder et. al. 2020)

Table 2: Action SC specifications for Blocks-World application and some example NL commands from user for each API. (*) denotes that the variable do not take part in command reduction (Utility Constraints), which is automatically detected and marked by CML (see Sec 3.2) (X denotes input).

| Action API Function | AID | Action SCs (';' separated) | Variable: Argument Type | Example commands |
|---|---|---|---|---|
| AddBlock (X1) | 1 | add a block at Z1 {Z1=X1} | X1: 'location' (*) | add a block at (2, 3); put a block at (2, 3) |
| Remove (X1) | 2 | remove Z1 {Z1=X1} | X1: 'block_set' | delete blue block; take away blue block |
| Move (X1, X2) | 3 | move Z1 to Z2 {Z1=X1, Z2=X2} | X1: 'block_set', X2: 'location' (*) | move blue block to the left of cube; shift green cube to (4, 5) |
| MoveByUnits (X1, X2, X3) | 4 | move Z1 along Z2 by Z3 units {Z1=X1, Z2=X2, Z3=X3} | X1: 'block_set', X2: 'direction', X3: 'number' | move blue block left by 2 units; shift green cube down by 3 units |
| UpdateColor (X1, X2) | 5 | change color of Z1 to Z2 {Z1=X1, Z2=X2}; color Z1 with Z2 {Z1=X1, Z2=X2} | X1:'block_set', X2: 'color' (*) | color A red; change color of B to blue |
| UpdateShape (X1, X2) | 6 | change shape of Z1 to Z2 {Z1=X1, Z2=X2} | X1:'block_set', X2: 'shape' (*) | set the shape of A to cube; make B square |
| Rename (X1, X2) | 7 | rename block Z1 to Z2 {Z1=X1, Z2=X2} | X1: 'block_set', X2: 'name' (*) | Name the block at (4, 5) as C; rename A to D |


- Mazumder. On-the-job Continual and Interactive Learning of Factual Knowledge and Language Grounding. PhD diss., UIC, 2021.
- Mazumder, Liu, Wang, Esmaeilpour. An Application-Independent Approach to Building Task-Oriented Chatbots with Interactive Continual Learning. HLDS Workshop at NeurIPS 2020.

# Command grounding module (CGM)
(Mazumder et. al. 2020)

- Rephraser and Tagger (*R*):
  - ✓ Given the user command C, *R* repharses C and tags each word or phrase in the rephrased C with either 'O' (i.e., not an argument type) or one of the possible argument types of the action SCs.
- SC Matcher (*M*):
  - ✓ Given the rephrased and tagged command C and the set T of (action or utility) SCs, Matcher *M* computes a match score *f*(t, C) for each *t* in T and returns the top ranked SC.
  - ✓ This work uses an information retrieval (IR) based unsupervised matching model for *M*


- Mazumder. On-the-job Continual and Interactive Learning of Factual Knowledge and Language Grounding. PhD diss., UIC, 2021.
- Mazumder, Liu, Wang, Esmaeilpour. An Application-Independent Approach to Building Task-Oriented Chatbots with Interactive Continual Learning. HLDS Workshop at NeurIPS 2020.

# Command grounding module (contd.)

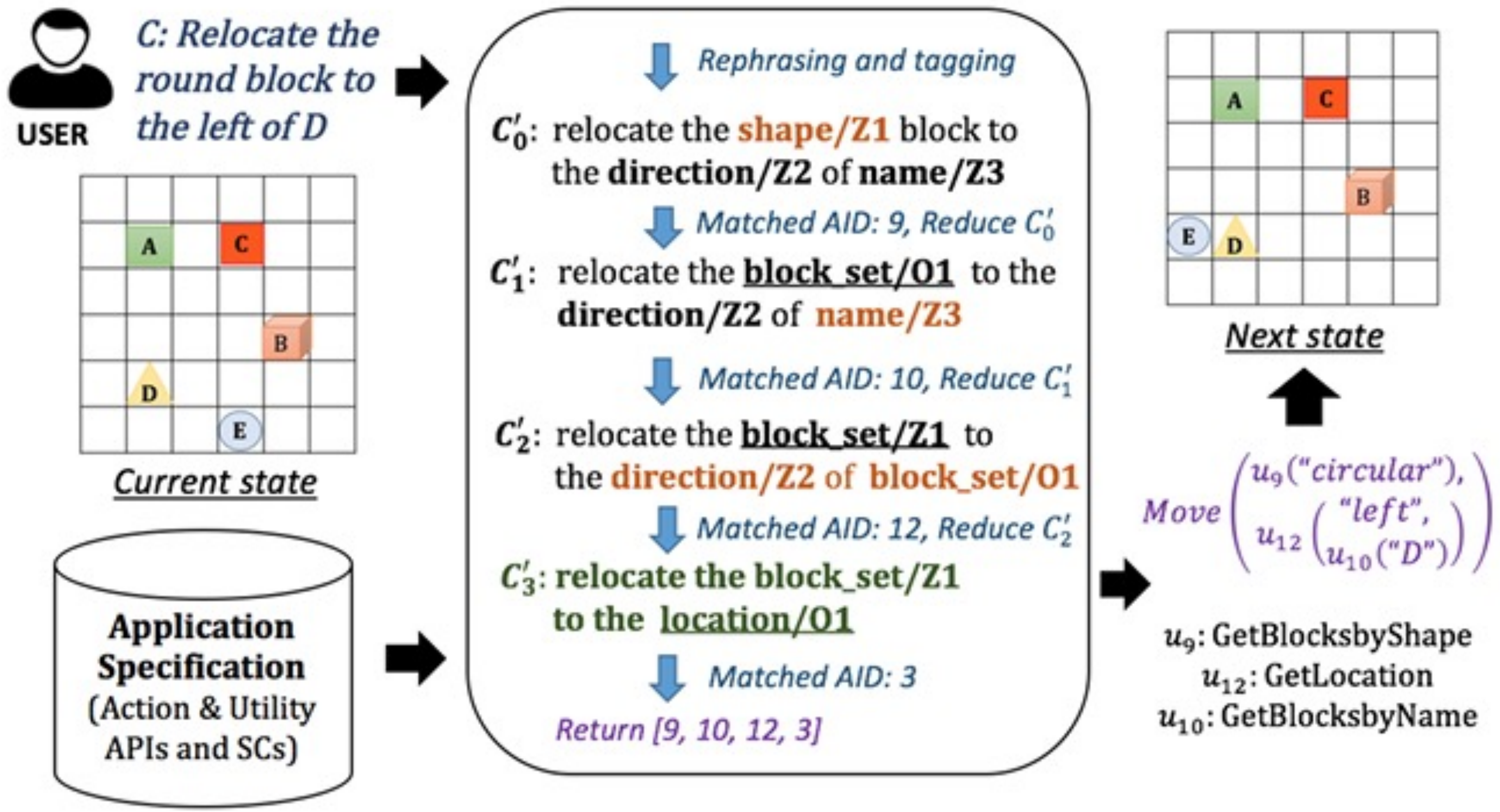
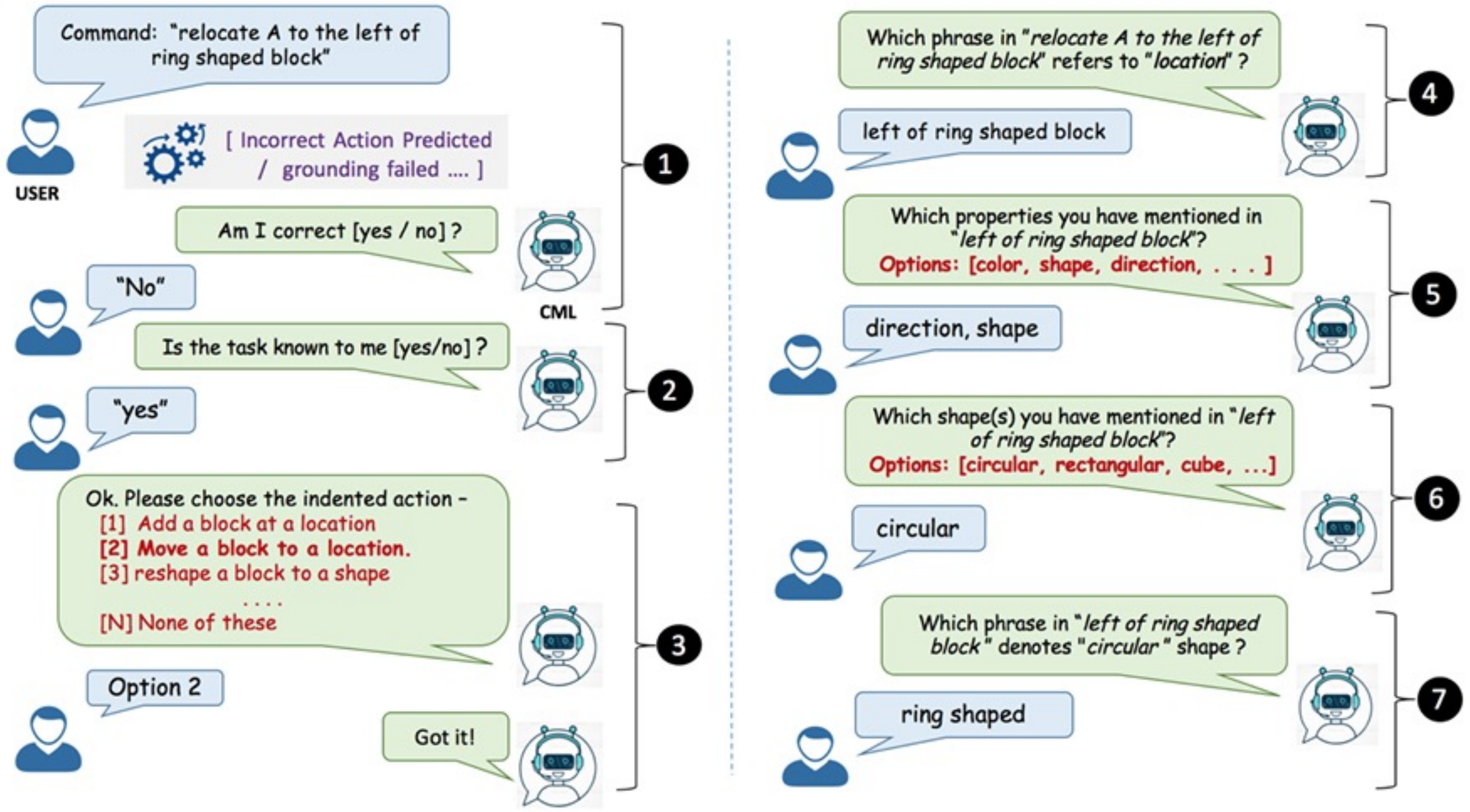



Figure 1: Working of CGM on a user command for Blocks-World. AID denotes the API IDs (see Tables 2 and 3).

- Mazumder. On-the-job Continual and Interactive Learning of Factual Knowledge and Language Grounding. PhD diss., UIC, 2021.
- Mazumder, Liu, Wang, Esmaeilpour. An Application-Independent Approach to Building Task-Oriented Chatbots with Interactive Continual Learning. HLDS Workshop at NeurIPS 2020.

# Continual interactive learning (Mazumder et. al. 2020)




- Mazumder. On-the-job Continual and Interactive Learning of Factual Knowledge and Language Grounding. PhD diss., UIC, 2021.
- Mazumder, Liu, Wang, Esmaeilpour. An Application-Independent Approach to Building Task-Oriented Chatbots with Interactive Continual Learning. HLDS Workshop at NeurIPS 2020.

# Experiment results
(Mazumder et. al. 2020)

**BERT-JISF**: joint intent detection and slot filling - fine-tunes a pre-trained BERT model to solve NLU (Chen et al, 2019).

A-acc: action intent prediction

Arg-F1: argument F1

Datasets

- BW: blocks-world
- WPD: Webpage design
- FB: flight booking

Table 5: Performance comparison of CML variants and BERT-JISF. Here, CML-vsm(-U) and CML-vsm results are the same for FB as utility APIs are absent in FB specifications].

| Models | BW | | WPD | | FB | |
|---|---|---|---|---|---|---|
| | A-acc | Arg-F1 | A-acc | Arg-F1 | A-acc | Arg-F1 |
| BERT-JISF | 57.56 | 52.42 | 57.26 | 50.36 | 33.64 | 54.41 |
| CML-jac | 66.44 | 63.70 | **72.68** | **81.26** | **88.58** | **96.75** |
| CML-vsm | 66.44 | 63.70 | 71.80 | **81.26** | 87.15 | **96.75** |
| CML-embed | **66.77** | **64.03** | 63.87 | 76.42 | 83.00 | **96.75** |
| CML-vsm (-R) | 63.48 | 61.15 | 66.07 | 72.56 | 75.48 | 84.15 |
| CML-vsm (-U) | 3.94 | 2.96 | 5.72 | 5.72 | - | - |
| CML-jac + SCL | **68.75** | **66.00** | **76.21** | **82.14** | 92.47 | **96.75** |
| CML-vsm + SCL | 68.09 | 65.35 | 74.89 | 81.71 | **93.32** | **96.75** |
| CML-jac + SCL + APL | **71.05** | **70.72** | **79.73** | **84.02** | 93.74 | **97.72** |
| CML-vsm + SCL + APL | 70.06 | 69.73 | 78.85 | 83.59 | **94.50** | **97.72** |


- Mazumder. On-the-job Continual and Interactive Learning of Factual Knowledge and Language Grounding. PhD diss., UIC, 2021.
- Mazumder, Liu, Wang, Esmaeilpour. An Application-Independent Approach to Building Task-Oriented Chatbots with Interactive Continual Learning. HLDS Workshop at NeurIPS 2020.

# Outline

# Open-domain Dialogue Learning after Deployment

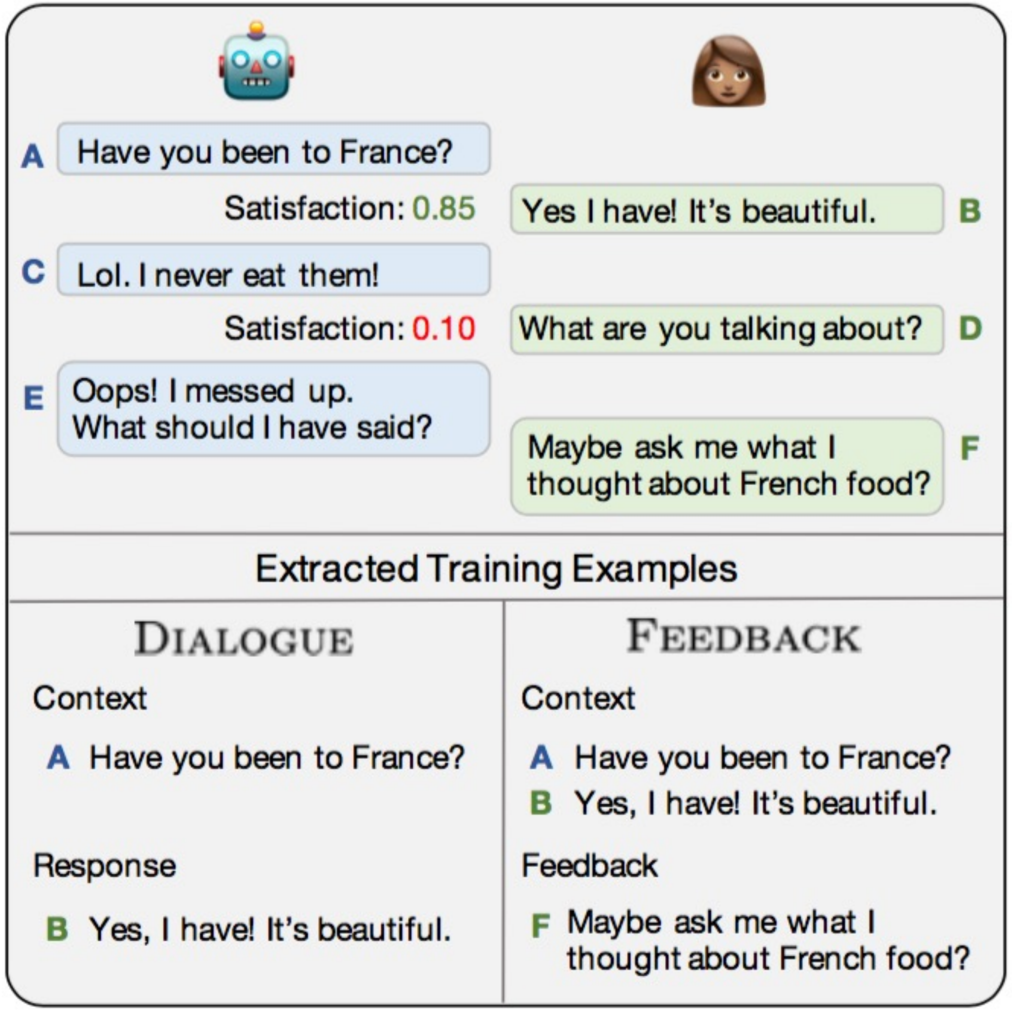
- Learning by extracting new training examples from conversations
  - ✓ Self-feeding Chatbot (Hancock et. al. 2019): extracts (context, response) pairs from the conversations and use it for continual training.
- Dialogue learning via role-playing games
  - ✓ LIGHT WILD (Shuster et. al. 2020) : Human players converse with agents situated in an open-domain fantasy world and showed that by training agents on in-game conversations they progressively improve.


- Hancock, Bordes, Mazare, and Weston. Learning from Dialogue after Deployment: Feed Yourself, Chatbot!. ACL 2019.
- Shuster, Urbanek, Dinan, Szlam and Weston. Deploying Lifelong Open-Domain Dialogue Learning. arXiv preprint arXiv:2008.08076, 2020 .

# Self-feeding Chatbot

(Hancock et. al. 2019)

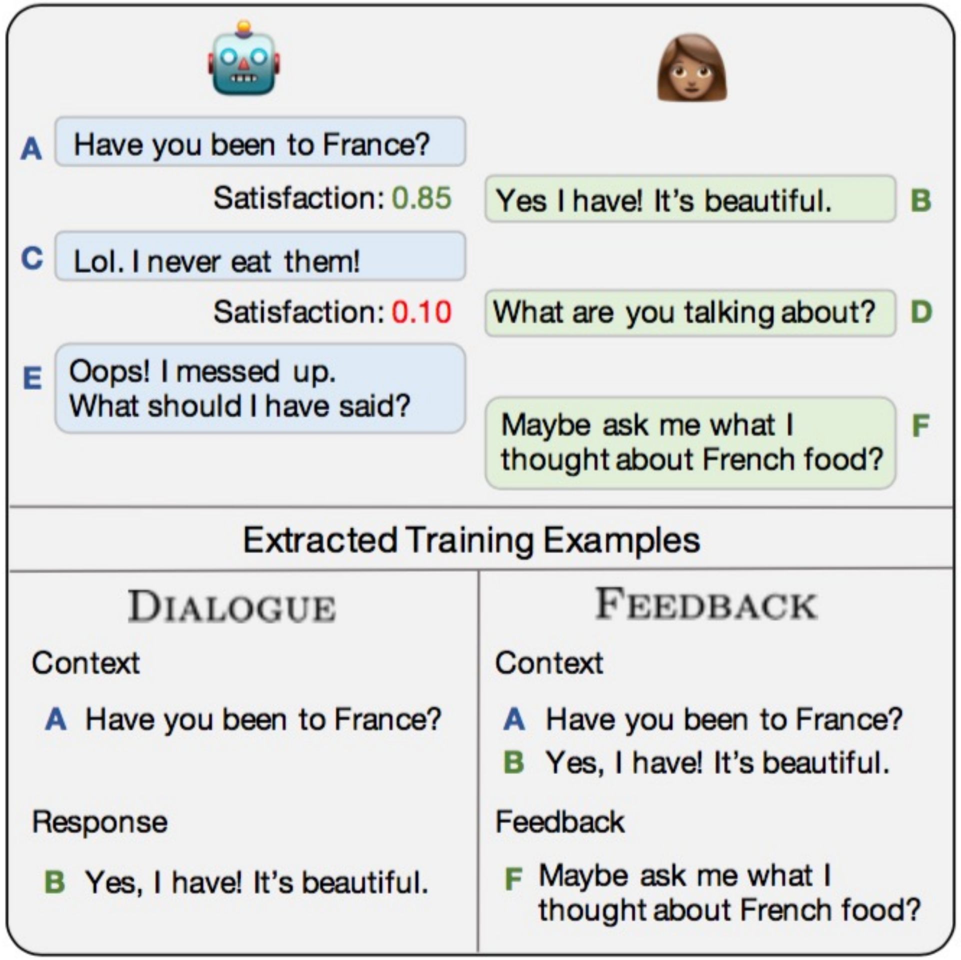
- Learning by extracting new training examples from conversations
  - ➢ If the **conversation appears to be going well**, the user's responses become new training examples to imitate.
  - ➢ Otherwise, on **making a mistake**, it asks the user for feedback to obtain a relevant response.
- The agent is retrained periodically using all available data




- Hancock, Bordes, Mazare, and Weston. Learning from Dialogue after Deployment: Feed Yourself, Chatbot!. ACL 2019.

# Self-feeding Chatbot : Learning
(Hancock et. al. 2019)

- Initial training phase
  - The agent is trained on two tasks using supervised Human-Human (HH) dialogue examples
    - **Task-1: DIALOGUE**
      - next utterance prediction, or what should I say next?
    - **Task-2: SATISFACTION**
      - how satisfied is my speaking partner with my responses?



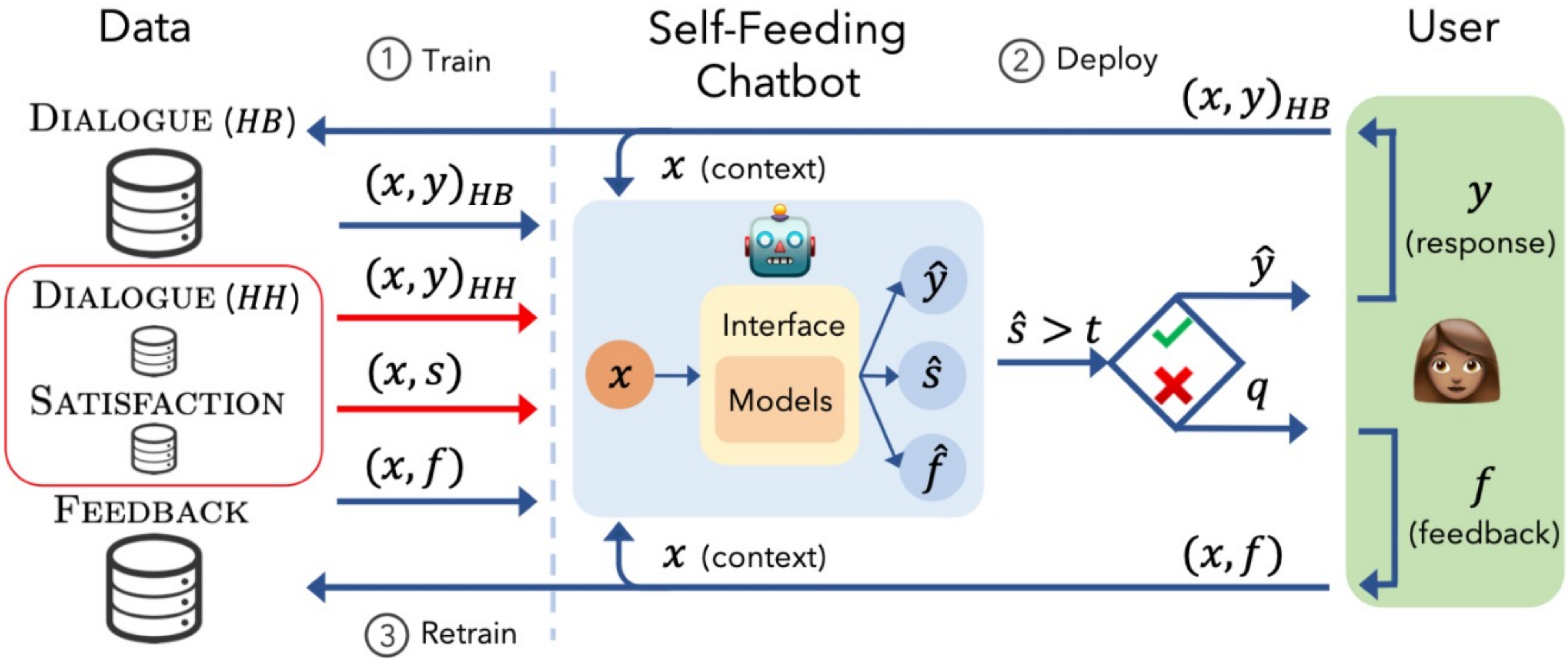

- Hancock, Bordes, Mazare, and Weston. Learning from Dialogue after Deployment: Feed Yourself, Chatbot!. ACL 2019.

# Self-feeding Chatbot : Learning
(Hancock et. al. 2019)

- Deployment phase
  - The agent engages in multi-turn conversations with users, extracting new *deployment* examples of two types.
    - Each turn, the agent **observes the context $x$** (i.e., the conversation history) and uses it to **predict its next utterance $\hat{y}$ and its partner's satisfaction $\hat{s}$**.
    - If $\hat{s} > t$ **(threshold),** the agent extracts a new Human-Bot (HB) DIALOGUE example using context $x$ and human's response $y$ and continues the conversation.
    - If $\hat{s} < t$**,** the agent requests feedback with a question $q$, and the resulting feedback response $f$ is used to create a new example for the **TASK-3 : FEEDBACK task** (what feedback am I about to receive?).

- Hancock, Bordes, Mazare, and Weston. Learning from Dialogue after Deployment: Feed Yourself, Chatbot!. ACL 2019.

# Self-feeding Chatbot : Working
(Hancock et. al. 2019)




- Hancock, Bordes, Mazare, and Weston. Learning from Dialogue after Deployment: Feed Yourself, Chatbot!. ACL 2019.

# Self-feeding Chatbot: Model Architecture
(Hancock et. al. 2019)

- Dialogue agent: built on the traditional Transformer architecture
- SATISFACTION task: The context $x$ is encoded with a Transformer and converted to the scalar satisfaction prediction $\hat{s}$.
- DIALOGUE and FEEDBACK tasks: set up as ranking problems: the model ranks a collection of candidate responses and returns the top-ranked one as its response.
  - Context $x$ is encoded with one Transformer and $\hat{y}$, $\hat{f}$ candiates encoded with another.
  - The score is calculated as the dot product of the encoded context and encoded candidate.


- Hancock, Bordes, Mazare, and Weston. Learning from Dialogue after Deployment: Feed Yourself, Chatbot!. ACL 2019.

# Self-feeding Chatbot: Performance
(Hancock et. al. 2019)

| **Human-Bot (HB)** | | **Human-Human (HH) Dialogue** | | | |
|---|---|---|---|---|---|
| **Dialogue** | **Feedback** | 20k | 40k | 60k | 131k |
| - | - | 30.3 (0.6) | 36.2 (0.4) | 39.1 (0.5) | 44.7 (0.4) |
| 20k | - | 32.7 (0.5) | 37.5 (0.6) | 40.2 (0.5) | 45.5 (0.7) |
| 40k | - | 34.5 (0.5) | 37.8 (0.6) | 40.6 (0.6) | 45.1 (0.6) |
| 60k | - | 35.4 (0.4) | 37.9 (0.7) | 40.2 (0.8) | 45.0 (0.7) |
| - | 20k | 35.0 (0.5) | 38.9 (0.3) | 41.1 (0.5) | 45.4 (0.8) |
| - | 40k | 36.7 (0.7) | 39.4 (0.5) | 41.8 (0.4) | 45.7 (0.6) |
| - | 60k | 37.8 (0.6) | 40.6 (0.5) | 42.2 (0.7) | 45.8 (0.7) |
| 60k | 60k | **39.7** (0.6) | **42.0** (0.6) | **43.3** (0.7) | **46.3** (0.8) |

Accuracy (hits@1/20) on the DIALOGUE task's hidden test set

- Hancock, Bordes, Mazare, and Weston. Learning from Dialogue after Deployment: Feed Yourself, Chatbot!. ACL 2019.

# Dialogue learning via role-playing games

(Shuster et. al. 2020)

- Goal: study the ability of an open-domain dialogue model to **iteratively learn from conversations with intrinsically motivated humans**.
  - ✓ Built and deploy a role-playing game
    - ➢ Human players role-play characters and converse with other characters (that are learning models) situated within the game world.
  - ✓ System iterates between collecting data of human-model interactions, retraining updated models on newly collected data, and redeploying them.
  - ✓ provides a natural metric to evaluate and compare models online using the continuation rate of players (how long they continue playing).

- Shuster, Urbanek, Dinan, Szlam, and Weston. Dialogue in the wild: Learning from a deployed role-playing game with humans and bots." In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021

# Open-domain dialogue as a game
(Shuster et. al. 2020)

- Core Game: involves pairing two agents in a given setting
  - Where one is a human and other is a dialogue agent (with an underlying machine learning model).
  - Two players are assigned characters, with given names, backstories (personas), their current location and its description.
  - Each player's goal is simply to act out (role-play) their character's dialogue in the given situation.

- Shuster, Urbanek, Dinan, Szlam, and Weston. Dialogue in the wild: Learning from a deployed role-playing game with humans and bots." In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021

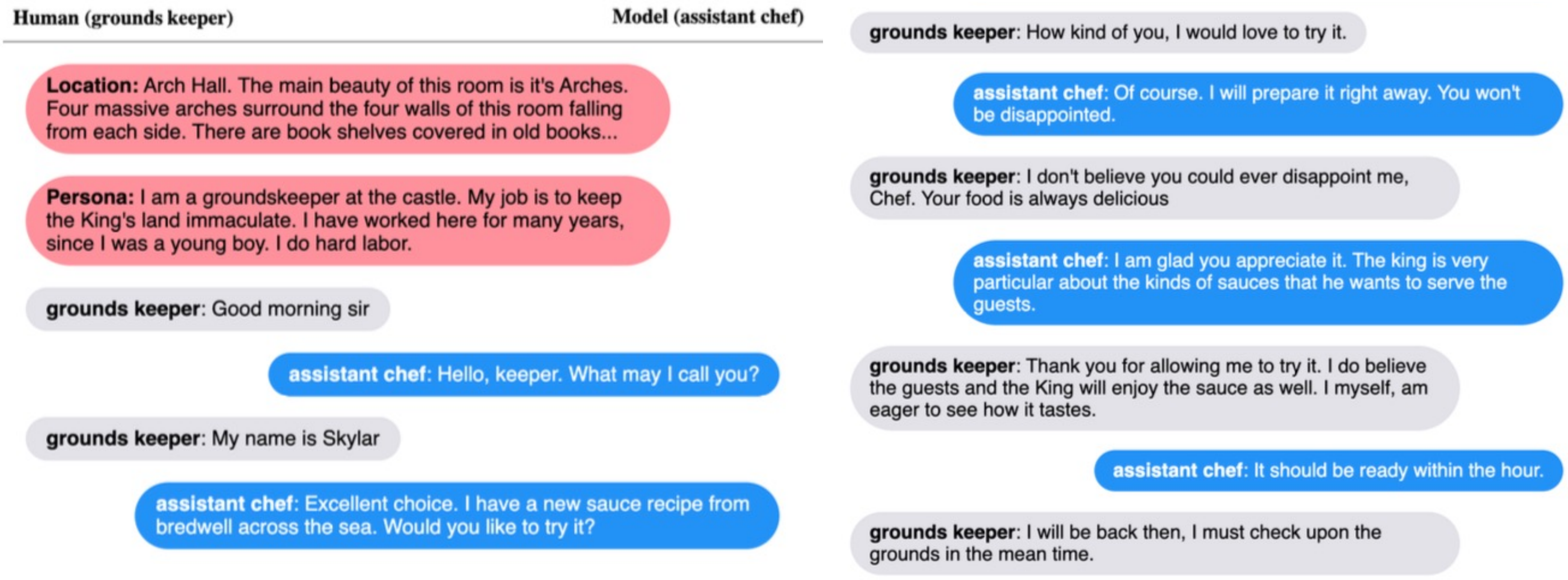
# Open-domain dialogue as a game
(Shuster et. al. 2020)

- Role-Playing (Acting) Score: an (automated) dungeon master (DM), to assess the quality of the player's role-playing.
  - For each dialogue turn, apply a learned model to the human player's dialogue, to **assess how likely their utterance is given the context**.
  - **convert the score in between 1 and 5 stars**, that is presented to the human player, to reward them for good acting.
- Acting scores are accumulated, and a player's total score is presented on a leaderboard for further motivation.


- Shuster, Urbanek, Dinan, Szlam, and Weston. Dialogue in the wild: Learning from a deployed role-playing game with humans and bots." In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021

# Open-domain dialogue as a game: Example

(Shuster et. al. 2020)




- Shuster, Urbanek, Dinan, Szlam, and Weston. Dialogue in the wild: Learning from a deployed role-playing game with humans and bots." In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021

# Dialogue learning via role-playing games: Performance

(Shuster et. al. 2020)

test locations do not overlap
with the training set locations

| | Retrieval Model (Hits@1/20 ⇑) | | | Generative Model (PPL ⇓) | | |
|---|---|---|---|---|---|---|
| Model | LIGHT Test | LIGHT Test Unseen | WILD Test | LIGHT Test | LIGHT Test Unseen | WILD Test |
| Round 1 | 87.12 | 82.43 | 81.61 | 12.67 | 11.81 | 13.42 |
| Round 2 | 87.65 | 82.70 | 84.60 | 12.57 | 11.74 | 12.31 |
| Round 3 | 87.72 | 83.48 | 87.63 | 12.54 | 11.75 | 11.79 |

- Round 1 : models trained on LIGHT MTurk data only.
- Round 2 : models trained on LIGHT MTurk data + 50k WILD examples collected from Round 1
- Round 3 : models trained on LIGHT MTurk data + 50k examples from Round 1 + an additional 180k examples collected from Round 2 deployment



- Shuster, Urbanek, Dinan, Szlam, and Weston. Dialogue in the wild: Learning from a deployed role-playing game with humans and bots." In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021

# Outline

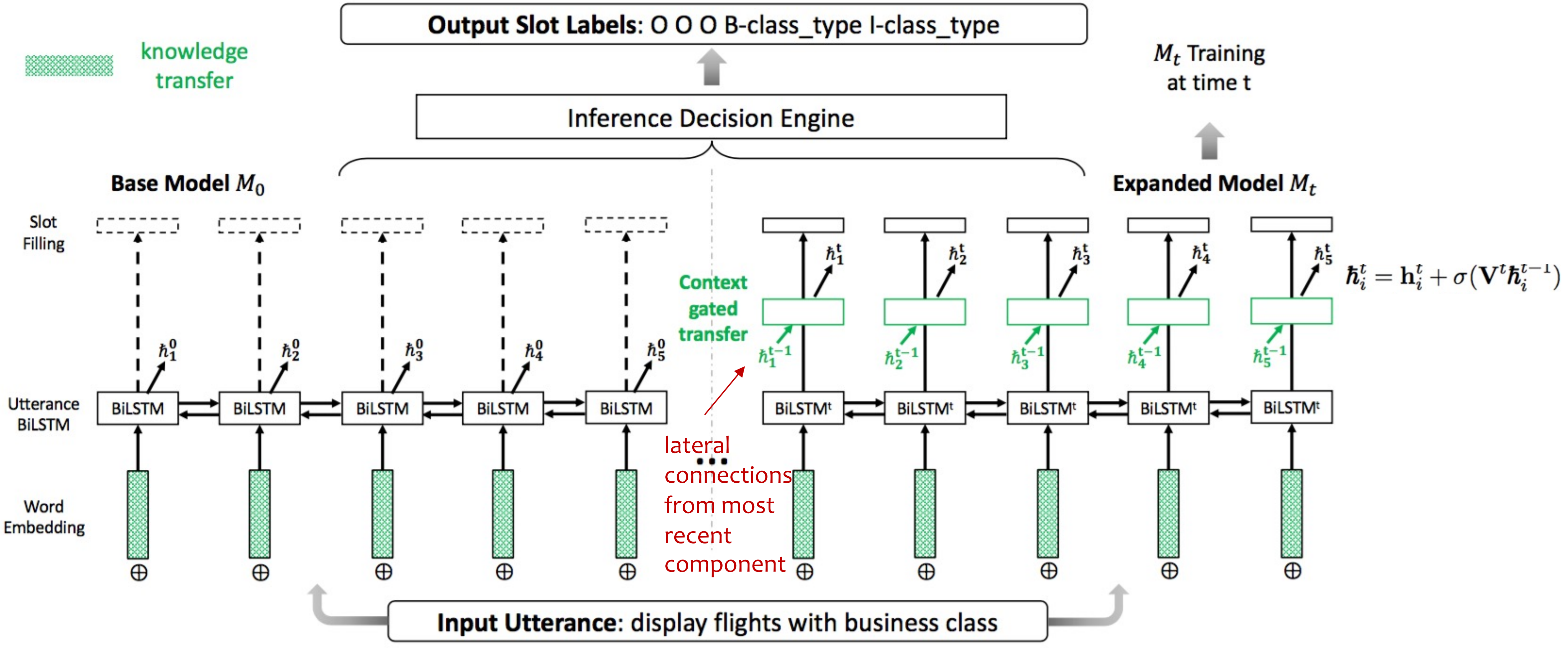
I. Lifelong and Continual Learning: An Introduction
II. Continuous Knowledge Learning during Conversation
III. Continual Language Learning and Grounding
IV. Open-Domain Dialogue Learning After Deployment
V. **Continual Learning for Task-oriented Dialogue Systems**
VI. Continual Learning of Conversational Skills
VII. Other Challenges & Summary

# Continual Learning for Semantic Slot Filling
(Shen et. al. 2019)

- A progressive slot filling model, *ProgModel*.
  - ✓ gradually **expands at each batch** by using a context gate for knowledge transfer.
    - *Word Embeddings Transfer and Gated Utterance Context Transfer*
  - ✓ Using the transferred knowledge, each newly expanded component is **trained in a progressive manner with new data**.
  - ✓ **previously trained components remains untouched** to avoid catastrophic forgetting.

- Shen, Zeng, & Jin. A progressive model to enable continual learning for semantic slot filling. *EMNLP-IJCNLP 2019*.

# ProgModel Architecture
(Shen et. al. 2019)




- Shen, Zeng, & Jin. A progressive model to enable continual learning for semantic slot filling. *EMNLP-IJCNLP 2019*.

# ProgModel : Inference Decision Engine

(Shen et. al. 2019)

- non-trainable separate component to avoid the potential catastrophic forgetting.
- consider two types of decision engines:
  - ✓ *t-IDE* : ProgModel using **only the output** $M^t$ **as decision engine**
  - ✓ *c-IDE*: for $i^{th}$ word, it **combines all outputs** from each component $M^t$

$$\sum_{k=0}^{t} P^k(i) I^k(i)$$

Aggregate (slot label) prob. for each word from all components

  - ➢ $I^k(i)$ → indicator function which is 1 if $i^{th}$ word is in the vocabulary of $M^k$
  - ➢ $P_j^k(i)$ → output probability of slot *j* for the $i^{th}$ word from $M^k$
  - ➢ The label with maximum probability is selected.


- Shen, Zeng, & Jin. A progressive model to enable continual learning for semantic slot filling. *EMNLP-IJCNLP 2019*.

# ProgModel Performance: ATIS Dataset (Shen et. al. 2019)

- *FT-AttRNN* : fine tunes current model only using new training data $U_t$
- *FT-Lr-AttRNN* : fine tunes current model using adjusted lower learning rate on $U_t$
- *FT-Cp-AttRNN* : copies the previous model and fine tunes the new copied model on $U_t$.
- *t-ProgModel*: using only output of $M_t$ as decision engine
- *c- ProgModel*: ProgModel using combined inference decision engine.

| Approach | Batch 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| AttRNN (upper bound) | 92.12 | 92.89 | 93.04 | 93.56 | 95.13 |
| FT-AttRNN | 92.12 | 91.85 | 89.98 | 91.25 | 88.03 |
| FT-Lr-AttRNN | | 91.96 | 86.46 | 88.03 | 86.58 |
| FT-Cp-AttRNN | | 92.10 | 90.06 | 91.98 | 89.67 |
| t-ProgModel | | 92.33 | 92.43 | 92.57 | 92.58 |
| c-ProgModel | | **92.40** | **92.64** | **92.71** | **93.91** |


- Shen, Zeng, & Jin. A progressive model to enable continual learning for semantic slot filling. *EMNLP-IJCNLP 2019*.
- Liu and Lane. Attention-based recurrent neural network models for joint intent detection and slot filling. *INTERSPEECH 2016*

# ProgModel Performance: Snips Dataset
(Shen et. al. 2019)

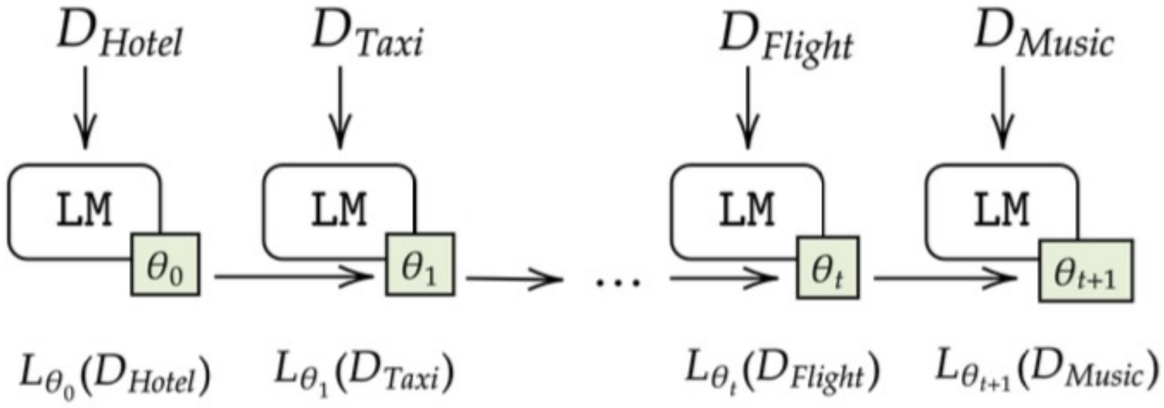
| Domain | Approach | Batch 0 | Batch 1 | Batch 2 |
|---|---|---|---|---|
| Add To Playlist | AttRNN (upper bound) | 79.58 | 86.74 | 88.89 |
| | FT-AttRNN | | 81.23 | 87.07 |
| | FT-Lr-AttRNN | | 78.99 | 86.61 |
| | FT-Cp-AttRNN | 79.58 | 84.67 | 87.15 |
| | t-ProgModel | | **86.12** | **88.30** |
| | c-ProgModel | | 85.51 | 87.25 |
| Book Restaurant | AttRNN (upper bound) | 79.49 | 89.78 | 90.03 |
| | FT-AttRNN | | 88.71 | 88.09 |
| | FT-Lr-AttRNN | | 88.57 | 87.89 |
| | FT-Cp-AttRNN | 79.49 | 89.06 | 88.14 |
| | t-ProgModel | | **89.45** | **89.54** |
| | c-ProgModel | | 89.40 | 89.40 |
| Get Weather | AttRNN (upper bound) | 76.48 | 91.12 | 93.56 |
| | FT-AttRNN | | 89.52 | 88.93 |
| | FT-Lr-AttRNN | | 89.09 | 88.56 |
| | FT-Cp-AttRNN | 76.48 | 89.82 | 90.09 |
| | t-ProgModel | | **90.73** | **93.12** |
| | c-ProgModel | | 89.92 | 90.95 |

| Domain | Approach | Batch 0 | Batch 1 | Batch 2 |
|---|---|---|---|---|
| Play Music | AttRNN (upper bound) | 77.48 | 87.79 | 89.13 |
| | FT-AttRNN | | 84.71 | 84.63 |
| | FT-Lr-AttRNN | | 84.53 | 84.16 |
| | FT-Cp-AttRNN | 77.48 | 84.85 | 86.10 |
| | t-ProgModel | | 86.05 | 87.26 |
| | c-ProgModel | | **87.00** | **88.45** |
| Rate Book | AttRNN (upper bound) | 92.64 | 98.45 | 99.07 |
| | FT-AttRNN | | 96.87 | 96.83 |
| | FT-Lr-AttRNN | | 96.20 | 96.86 |
| | FT-Cp-AttRNN | 92.64 | 97.06 | 97.93 |
| | t-ProgModel | | 97.50 | **98.89** |
| | c-ProgModel | | **98.19** | 98.20 |
| Search Creative Work | AttRNN (upper bound) | 66.32 | 89.01 | 89.67 |
| | FT-AttRNN | | 85.93 | 85.46 |
| | FT-Lr-AttRNN | | 84.69 | 84.45 |
| | FT-Cp-AttRNN | 66.32 | 87.25 | 86.36 |
| | t-ProgModel | | 88.21 | 88.25 |
| | c-ProgModel | | **88.79** | **88.83** |
| Search Screening Event | AttRNN (upper bound) | 89.30 | 95.68 | 97.34 |
| | FT-AttRNN | | 93.40 | 94.53 |
| | FT-Lr-AttRNN | | 91.87 | 93.56 |
| | FT-Cp-AttRNN | 89.30 | 93.81 | 94.56 |
| | t-ProgModel | | **95.01** | **96.90** |
| | c-ProgModel | | 93.62 | 94.31 |

- Shen, Zeng, & Jin. A progressive model to enable continual learning for semantic slot filling. *EMNLP-IJCNLP 2019*.

# Continual Learning in Task-oriented Dialogue Systems (ToDs) (Madotto et. al. 2020)

- A continual learning benchmark for ToDS with 37 domains -




✓ four settings: intent recognition, state tracking, natural language generation, and end-to-end

✓ implement and compare multiple existing continual learning baselines

➢ *regularization*, *rehearsal* and *architectural*.

✓ propose a simple and effective architectural method based on residual adapters (Houlsby et al., 2019).

- Madotto et. al. Continual Learning in Task-Oriented Dialogue Systems. *EMNLP 2021*

# Continual Learning in ToDs: Problem Formulation

(Madotto et. al. 2020)

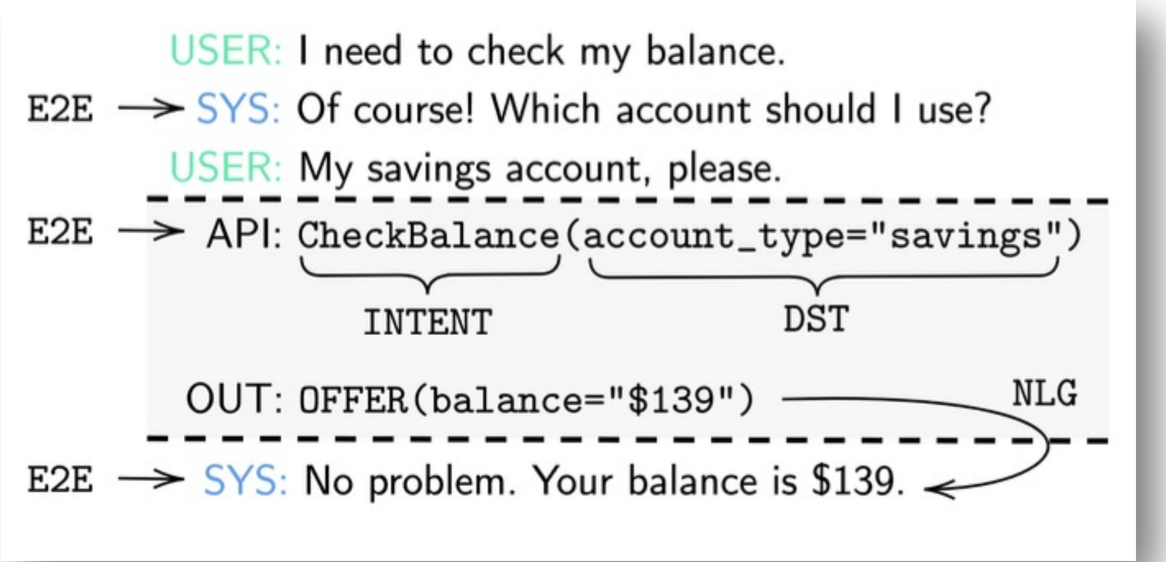
H: Dialogue History
I: intent or the api-name
S: NL response

- Modularized setting by their input-out pairs:

$$H \to \mathbf{I} \qquad \text{(INTENT)}$$
$$H \to \mathbf{I}(s_1 = v_1, \ldots, s_k = v_p) \qquad \text{(DST)}$$
$$\underbrace{\mathbf{I}(s_1 = v_1, \ldots, s_k = v_p)}_{S_{OUT}} \to S \qquad \text{(NLG)}$$

- End-to-End (E2E) formulation:

$$H \to \underbrace{\mathbf{I}(s_1 = v_1, \ldots, s_k = v_p)}_{S_{API}}$$
$$H + \underbrace{\mathbf{I}(s_1 = v_1, \ldots, s_k = v_p)}_{S_{OUT}} \to S$$




- Dialogue Dataset: input-out pair from one of the four settings in consideration



- Madotto et. al. Continual Learning in Task-Oriented Dialogue Systems. *EMNLP 2021*

# AdapterCL
## (Madotto et. al. 2020)

- Employ a decoder-only pre-trained Language Models (e.g. GPT-2)
  - ✓ trained by minimizing the negative log-likelihood
  - ✓ only the task-specific parameter are trained while the original weights are left frozen.

- parameterizes each task using Residual Adapters (Houlsby et al., 2019) and uses a perplexity-based classifier to select which adapter to use at testing time.
  - ❑ **Residual adapter**: trainable parameters added on top of each transformer layer, which **steer the output distribution** of a pre-trained model **without modifying its original weights.**

- Madotto et. al. Continual Learning in Task-Oriented Dialogue Systems. *EMNLP 2021*
- Houlsby et. al. Parameter-Efficient Transfer Learning for NLP. ICML 2019

# AdapterCL
## (Madotto et. al. 2020)

- Residual adapter computation:

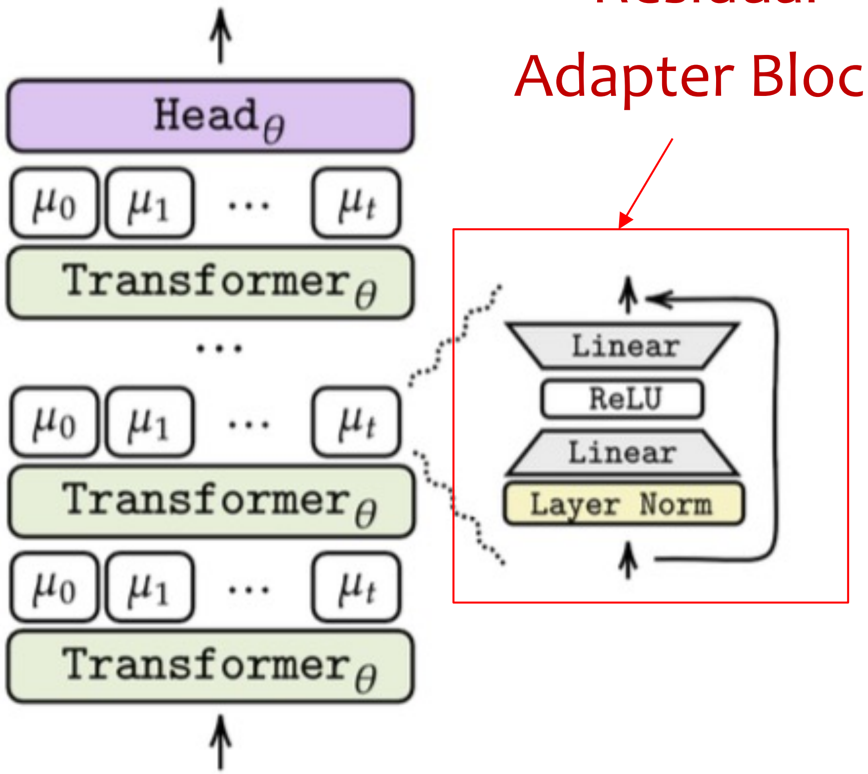
$$\text{Adapter}_{\mu_i^l}(H) = \text{ReLU}(\text{LN}(x)W_l^E)W_l^D + H$$

Seq. length Hidden size

$$H \in \mathbb{R}^{p \times d}$$

- To learn new task (Dataset $D_t$), spawn a new Adapter ( $\mu_t$ ) and train its parameters.
- Loss is optimized over $\mu_t$ to guarantee that each task is independently learned.

Residual Adapter Block




High-level representation of the AdapterCL




- Madotto et. al. Continual Learning in Task-Oriented Dialogue Systems. *EMNLP 2021*
- Houlsby et. al. Parameter-Efficient Transfer Learning for NLP. ICML 2019

# AdapterCL : Perplexity-Based Classifier
(Madotto et. al. 2020)

- In the CL setting →
  - ✓ during **training** task-id is provided → $\mu_t$ is optimized over $D_t$
  - ✓ during **testing** task-id is **not** provided → model has to predict which adapter to use for accomplishing the task.
- Following (Wortsman et al. 2020), utilize the perplexity of each adapter over the input X as a measure of uncertainty.
  - ❑ selecting the adapter with lowest perplexity → select the most confident model to generate the output sequence.

N - Adapters

$$\alpha_t = \mathrm{PPL}_{\mu_t}(X)\ \forall t \in 1, \cdots, N$$

Adapter Selected

$$t^* = \operatorname{argmin} \alpha_0, \cdots, \alpha_N$$

- Madotto et. al. Continual Learning in Task-Oriented Dialogue Systems. *arXiv preprint arXiv:2012.15504* (2020).

# Continual Learning in Task-Oriented Dialogue Systems: Performance (Madotto et. al. 2020)

| | | | INTENT | DST | NLG | |
|---|---|---|---|---|---|---|
| **Method** | **+Param.** | **Mem.** | *Accuracy*↑ | *JGA*↑ | *EER*↓ | *BLEU*↑ |
| *VANILLA* | - | $\emptyset$ | $4.08 \pm 1.4$ | $4.91 \pm 4.46$ | $48.73 \pm 3.81$ | $6.38 \pm 0.6$ |
| *L2* | $\|\theta\|$ | $\emptyset$ | $3.74 \pm 1.4$ | $3.81 \pm 3.44$ | $55.68 \pm 7.09$ | $5.4 \pm 0.9$ |
| *EWC* | $2\|\theta\|$ | $\emptyset$ | $3.95 \pm 1.3$ | $5.22 \pm 4.46$ | $58.2 \pm 3.66$ | $5.06 \pm 0.5$ |
| *AGEM* | - | $t\|M\|$ | $34.04 \pm 6.36$ | $6.37 \pm 4.0$ | $62.09 \pm 6.88$ | $4.54 \pm 0.6$ |
| *LAMOL* | - | $\emptyset$ | $7.49 \pm 6.35$ | $4.55 \pm 3.48$ | $66.11 \pm 6.97$ | $3.0 \pm 0.9$ |
| *REPLAY* | - | $t\|M\|$ | $81.08 \pm 1.37$ | $30.33 \pm 1.24$ | $\mathbf{17.72} \pm 0.85$ | $\mathbf{17.4} \pm 0.68$ |
| *ADAPT* | $t\|\mu\|$ | $\emptyset$ | $\mathbf{90.46} \pm 0.6$ | $\mathbf{35.06} \pm 0.52$ | $31.78 \pm 1.28$ | $16.76 \pm 0.34$ |
| *MULTI* | - | - | $95.45 \pm 0.1$ | $48.9 \pm 0.2$ | $12.56 \pm 0.2$ | $23.61 \pm 0.1$ |

E2E results in term of Intent accuracy, Joint-Goal-Accuracy (JGA), Slot-Error-Rate (EER) and BLUE. +Param shows the additional number of parameters per task, and Mem the episodic memory size need per task.

- Madotto et. al. Continual Learning in Task-Oriented Dialogue Systems. *arXiv preprint arXiv:2012.15504* (2020).

# Outline

I. Lifelong and Continual Learning: An Introduction
II. Continuous Knowledge Learning during Conversation
III. Continual Language Learning and Grounding
IV. Open-Domain Dialogue Learning After Deployment
V. Continual Learning for Task-oriented Dialogue Systems
VI. **Continual Learning of Conversational Skills**
VII. Other Challenges & Summary

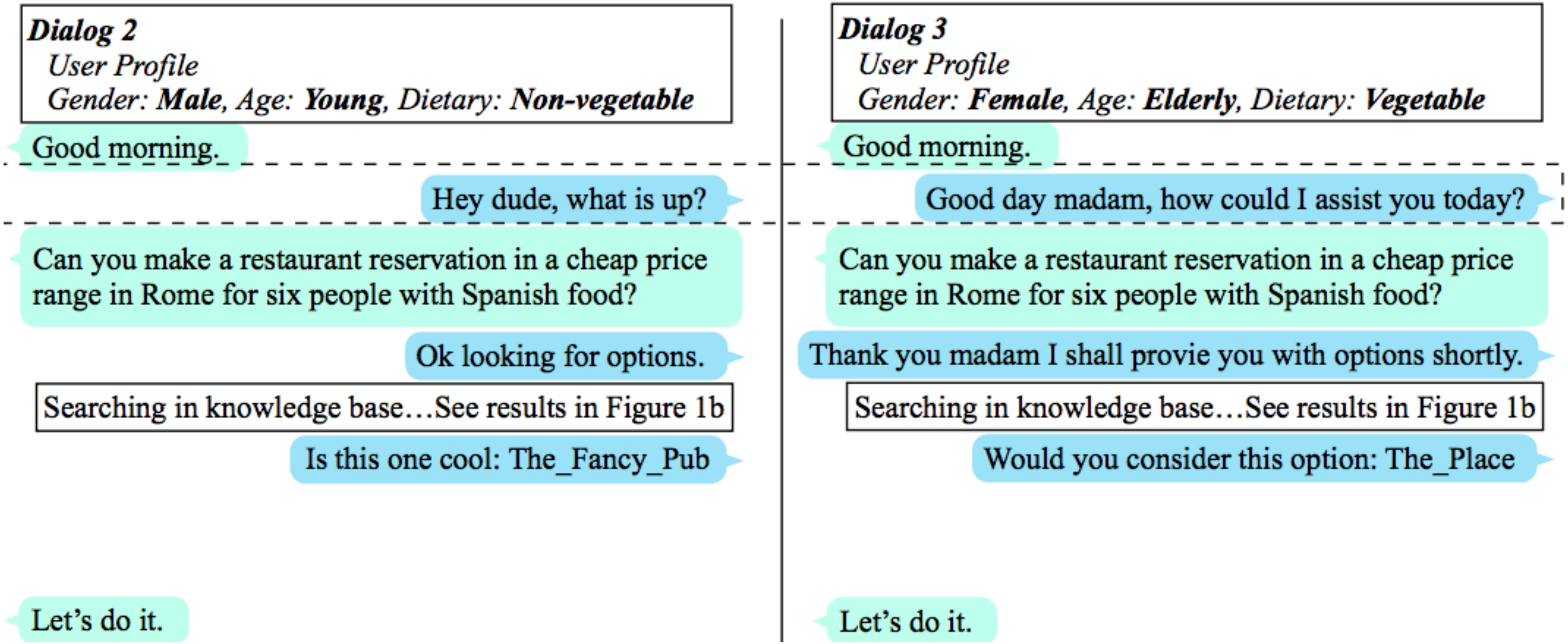
# Learning user behaviors and preferences

- Given a conversation context, the chatbot can learn the user's behavioral and preference profile.
  - ✓ whether a user feels more excited or gets annoyed while conversing on a particular topic, what his/her likes and dislikes are etc.
  - ✓ PERSONALIZED MEMN2N (Luo et. al. 2019); $P^2$ BOT (Liu et. al. 2020)

Utilize user profile knowledge in modeling future conversations to make them more engaging with the user.


- Luo, Huang, Zeng, Nie and Sun. *Learning personalized end-to-end goal-oriented dialog*. AAAI 2019.
- Liu, Chen, Chen, Lou, Chen, Zhou, and Zhang. *You impress me: Dialogue generation via mutual persona perception*. ACL 2020.

# Example – Personalized goal-oriented dialogues





- Luo, Huang, Zeng, Nie and Sun. *Learning personalized end-to-end goal-oriented dialog*. AAAI 2019.
- Liu, Chen, Chen, Lou, Chen, Zhou, and Zhang. *You impress me: Dialogue generation via mutual persona perception*. ACL 2020.

# Learning emotions, moods and sentiments

- Recognizing emotional state and sentiments (Zhou et al. 2018; Pamungkas 2019) of the user and leverage it to generate empathetic responses can be useful to building therapeutic chatbots.
  - ✓ Emotional Chatting Machine (Zhou et al. 2018) : can generate appropriate responses not only in content (*relevant and grammatical*) but also in emotion (*emotionally consistent*).

| User: Worst day ever. I arrived late because of the traffic. |
|---|
| Basic Seq2Seq: You were late. |
| ECM (*Like*): I am always here to support you. |
| ECM (*Happy*): Keep smiling! Things will get better. |
| ECM (*Sad*): It's depressing. |
| ECM (*Disgust*): Sometimes life just sucks. |
| ECM (*Angry*): The traffic is too bad! |


- Zhou, Huang, Zhang, Zhu, and Liu. Emotional chatting machine: Emotional conversation generation with internal and external memory. AAAI 2018.
- Pamungkas. *Emotionally-aware chatbots: A survey*. arXiv preprint arXiv:1906.09774, 2019.

# Modeling situation-aware conversations

- Understanding the situation and spatial-temporal context of a person to decide the conversation strategy is a key characteristic of the human conversation process
  - ✓ Continuously learning from the conversation history of the user provides a scope for chatbots to learn user's conversation profile, e.g.,
    - what time of a day the user generally likes to talk or remains busy
    - whether the user is in a meeting or not …

# Outline

I. Lifelong and Continual Learning: An Introduction
II. Continuous Knowledge Learning during Conversation
III. Continual Language Learning and Grounding
IV. Open-Domain Dialogue Learning After Deployment
V. Continual Learning for Task-oriented Dialogue Systems
VI. Continual Learning of Conversational Skills
VII. **Other Challenges & Summary**

# Other Challenges of CL in Dialogues

- Dealing with Wrong Knowledge from Users
  - ✓ How to deal with the issue of acquiring intentional or unintentional wrong knowledge from them ?
  - ✓ Can be addressed through a cross-verification strategy.
    - After acquiring a piece of new knowledge, the agent can **store it in an unverified knowledge buffer.**
    - Next, while chatting with some other users **in future sessions** to accomplish related tasks, the chatbot can ask them to **verify the unverified knowledge**

- Liu and Mazumder. *Lifelong and Continual Learning Dialogue Systems: Learning during Conversation.* AAAI-2021

# Other Challenges of CL in Dialogues

- Revision of Knowledge
  - ✓ How to revise or correct the wrong knowledge once it is detected?
  - ✓ Requires -
    - a **knowledge monitoring system** : **detect contradictions** in the knowledge base
    - a **knowledge revision method** : **revise the wrong knowledge** and also all the consequences inferred from it.


- Liu and Mazumder. *Lifelong and Continual Learning Dialogue Systems: Learning during Conversation.* AAAI-2021

# Other Challenges of CL in Dialogues

- Learning New Task Completion Skills from Users
  - ✓ Modern task-oriented chatbots are deployed with a finite set of task completion skills which they have been *preprogrammed* with to perform
  - ✓ Can end users use natural language dialogues to program their own chatbots and endow them with new skills after deployment?
    - ➢ lead to personalization of virtual assistants.

- Liu and Mazumder. *Lifelong and Continual Learning Dialogue Systems: Learning during Conversation.* AAAI-2021

# Other Challenges of CL in Dialogues

- One-shot or few shot continual learning
  - The amount of **ground-truth data** that can be **acquired during interaction** with human users is **often very small, one or a few**.
  - To learn continually and effectively, we need one-shot or few-shot continual learning methods.
    - Current methods are still very weak.
- In general, the current deep learning-based Continual Learning methods still have serious catastrophic forgetting problems.
  - Not ready for real-world applications. Some engineering hacks or data augmentations will be needed to get around of it.

- Liu and Mazumder. *Lifelong and Continual Learning Dialogue Systems: Learning during Conversation.* AAAI-2021

# Summary

- Classic ML: isolated and closed-world **offline learning**
  - No learning after deployment
- Dialogue systems or any AI agent should continuously learn after deployment or on the job (Liu, 2020; Liu and Mazumder, 2021)
  - The agent becomes smarter and smarter
- Current techniques are still in their infancy, but
  - Some methods are ready for practical applications.

# Reading List [non-exhaustive]

- **Lifelong and Continual Learning: Introduction**
  - Sebastian Thrun. *Is learning the n-th thing any easier than learning the first?* In NIPS, 1996.
  - Sebastian Thrun. *Lifelong Learning Algorithms*. In Learning To Learn, Kluwer Academic Publishers. 1998.
  - Daniel L Silver, Qiang Yang, and Lianghao Li. *Lifelong Machine Learning Systems: Beyond Learning Algorithms.* In AAAI Spring Symposium: Lifelong Machine Learning, 2013.
  - Paul Ruvolo and Eric Eaton. *ELLA: An efficient lifelong learning algorithm*. In ICML 2013.
  - Geli Fei, Shuai Wang, and Bing Liu. *Learning Cumulatively to Become More Knowledgeable*. In KDD-2016
  - Zixuan Ke, Bing Liu, and Xingchang Huang. *Continual Learning of a Mixed Sequence of Similar and Dissimilar Tasks*. In NeurIPS 2020
  - Zhiyuan Chen and Bing Liu. *Lifelong machine learning*. Morgan & Claypool. 2018

# Reading List [non-exhaustive]

## On the job Continual Learning

- Bing Liu, Eric Robertson, Scott Grigsby, and Sahisnu Mazumder. *Self-Initiated Open World Learning for Autonomous AI Agents.* AAAI Spring Symposium 2022.
- Sahisnu Mazumder. *On-the-job Continual and Interactive Learning of Factual Knowledge and Language Grounding*. PhD diss., University of Illinois at Chicago, 2021.
- Bing Liu and Sahisnu Mazumder. *Lifelong and Continual Learning Dialogue Systems: Learning during Conversation*. In AAAI 2021
- Bing Liu. *Learning on the Job: Online Lifelong and Continual Learning*. In AAAI 2020

## Continuous Knowledge Learning during Conversation

- Tsugumi Otsuka, Kazunori Komatani, Satoshi Sato, and Mikio Nakano. *Generating More Specific Questions for Acquiring Attributes of Unknown Concepts from Users*. In SIGDIAL 2013
- Kohei Ono, Ryu Takeda, Eric Nichols, Mikio Nakano, and Kazunori Komatani. *Toward lexical acquisition during dialogues through implicit confirmation for closed-domain chatbots*. In Workshop Chatbots Conversational Agent Technologies, 2016.

# Reading List [non-exhaustive]

- **Continuous Knowledge Learning during Conversation (contd..)**
  - Kohei Ono, Ryu Takeda, Eric Nichols, Mikio Nakano, and Kazunori Komatani. *Lexical acquisition through implicit confirmations over multiple dialogues.* In SIGDIAL 2017.
  - Komatani, Kazunori, and Mikio Nakano. *User Impressions of Questions to Acquire Lexical Knowledge*. In SIGDIAL. 2020.
  - Sahisnu Mazumder, Nianzu Ma, and Bing Liu. *Towards a continuous knowledge learning engine for chatbots.* arXiv preprint arXiv:1802.06024, 2018.
  - Sahisnu Mazumder, Bing Liu, Shuai Wang, and Nianzu Ma. *Lifelong and Interactive Learning of Factual Knowledge in Dialogues*. In SIGDIAL-2019
  - Sahisnu Mazumder, Bing Liu, Nianzu Ma and Shuai Wang. *Continuous and Interactive Factual Knowledge Learning in Verification Dialogues*. In HAMLETS - NeurIPS-2020 Workshop, 2020.
  - Jon Ander Campos, Kyunghyun Cho, Arantxa Otegi, Aitor Soroa, Eneko Agirre, and Gorka Azkune. *Improving Conversational Question Answering Systems after Deployment using Feedback-Weighted Learning*. In COLING 2020.

# Reading List [non-exhaustive]

- **Continuous Knowledge Learning during Conversation (contd..)**
  - Ben Hixon, Peter Clark, and Hannaneh Hajishirzi. *Learning knowledge graphs for question answering through conversational dialog*. In NAACL-HLT 2015.
  - Bing Liu and Chuhe Mei. *Lifelong Knowledge Learning in Rule-based Dialogue Systems*. arXiv:2011.09811, 2020.
- **Continual Language Learning and Grounding**
  - Sida I. Wang, Percy Liang, and Christopher D. Manning. *Learning Language Games through Interaction*. In ACL 2016
  - Sida I. Wang, Samuel Ginn, Percy Liang, and Christopher D. Manning. *Naturalizing a Programming Language via Interactive Learning.* In ACL 2017.
  - Shobhit Chaurasia and Raymond Mooney. *Dialog for language to code*. In *IJCNLP* 2017.

# Reading List [non-exhaustive]

- **Continual Language Learning and Grounding (Contd..)**
  - Ziyu Yao, Yiqi Tang, Wen-tau Yih, Huan Sun, and Yu Su. *An Imitation Game for Learning Semantic Parsers from User Interaction*. In *EMNLP* 2020.
  - Sahisnu Mazumder, Bing Liu, Shuai Wang, Sepideh Esmaeilpour. *An Application-Independent Approach to Building Task-Oriented Chatbots with Interactive Continual Learning.* In HLDS Workshop at NeurIPS 2020.

- **Open-Domain Dialogue Learning**
  - Braden Hancock, Antoine Bordes, Pierre-Emmanuel Mazare, and Jason Weston. *Learning from Dialogue after Deployment: Feed Yourself, Chatbot!.* In ACL 2019.
  - Kurt Shuster, Jack Urbanek, Emily Dinan, Arthur Szlam, and Jason Weston. *Deploying Lifelong Open-Domain Dialogue Learning*. arXiv preprint arXiv:2008.08076, 2020.

# Reading List [non-exhaustive]

- **Continual Learning for Task-oriented Dialogue Systems**
  - Han Li, Jihwan Lee, Sidharth Mudgal, Ruhi Sarikaya, and Young-Bum Kim. *Continuous Learning for Large-scale Personalized Domain Classification.* In NAACL-HLT 2019.
  - Yilin Shen, Xiangyu Zeng, and Hongxia Jin. *A progressive model to enable continual learning for semantic slot filling*. In *EMNLP-IJCNLP 2019*.
  - Madotto, Andrea, Zhaojiang Lin, Zhenpeng Zhou, Seungwhan Moon, Paul Crook, Bing Liu, Zhou Yu, Eunjoon Cho, and Zhiguang Wang. *Continual Learning in Task-Oriented Dialogue Systems*. *arXiv preprint arXiv:2012.15504*, 2020.

# Reading List

- **Conversational Skill Learning**
  - Liangchen Luo, Wenhao Huang, Qi Zeng, Zaiqing Nie, and Xu Sun. *Learning personalized end-to-end goal-oriented dialog*. In AAAI 2019.
  - Qian Liu, Yihong Chen, Bei Chen, Jian-Guang Lou, Zixuan Chen, Bin Zhou, and Dongmei Zhang. *You impress me: Dialogue generation via mutual persona perception*. In ACL 2020.
  - Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. *Emotional chatting machine: Emotional conversation generation with internal and external memory.* In AAAI 2018.
  - Pamungkas, Endang Wahyu. *Emotionally-aware chatbots: A survey*. arXiv preprint arXiv:1906.09774, 2019.

# Thank You

Q&A