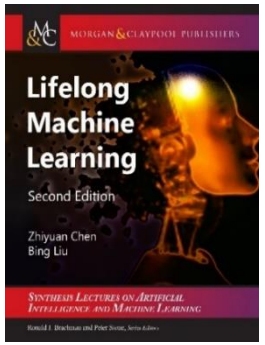
Self-Initiated Open World Learning for Autonomous AI Agents

https://arxiv.org/abs/2110.11385



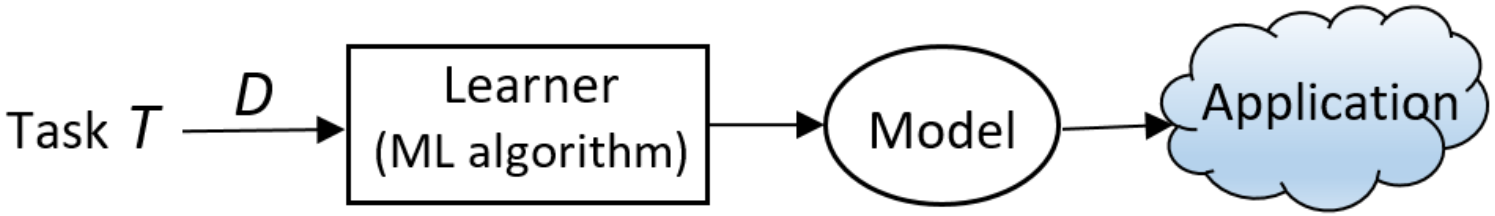
Bing Liu[1], Eric Robertson[2], Scott Grigsby[2] & Sahisnu Mazumder[1]

[1] University of Illinois at Chicago

[2] PAR Government Systems Corporation

Introduction

- Classic machine learning (ML): Isolated single-task learning



- Key weaknesses of the classic ML paradigm
  - Closed-world assumption (i.i.d): nothing new or novel in application
  - Model is fixed during application: no model revision in application
  - No knowledge accumulation: learn each task independently using a large amount of labeled training data
- Suitable for only well-defined tasks in narrow domains

Chen and Liu. Lifelong machine learning. Morgan & Claypool. 2015, 2018

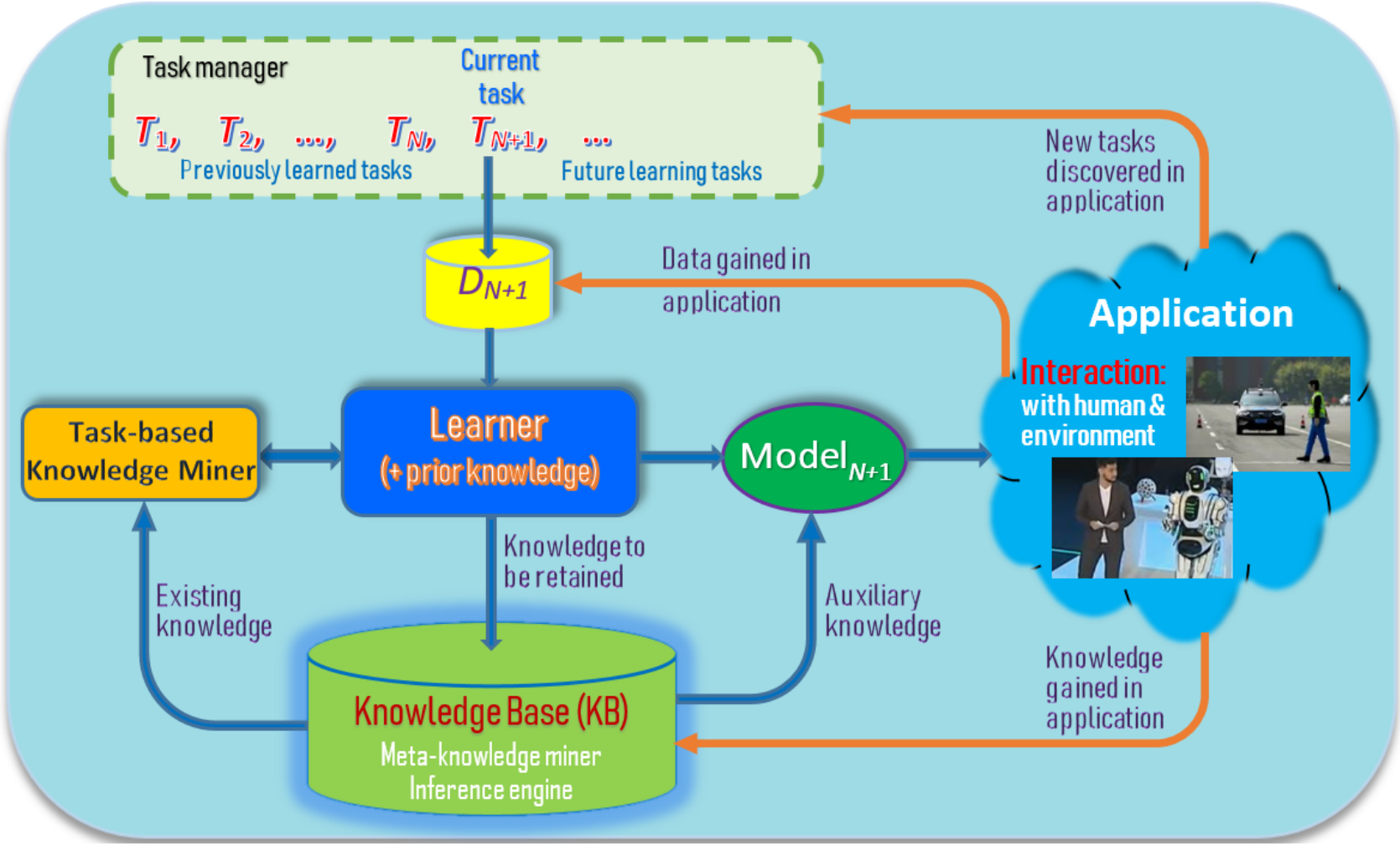
Introduction

- Increasingly, AI agents are facing the real open world,
  - which is full of unknowns or novelties
  - **Open world learning (OWL)** is not a new idea (Bendale and Boult, CVPR-2015; Fei, Wang and Liu, KDD-2016; Senator, SAIL-ON-2019; Langley, AAAI-2021)
- We ask the question of how to make AI agents learn fully autonomously and continually/lifelong.
  - learn by themselves in a self-motivated and self-supervised manner
    - rather than being re-trained periodically - initiated by human engineers.
  - Self-initiated Open World Learning (SiOWL)
    - Put SiOWL in the context of existing research

Liu, Robertson, Grigsby, and Mazumder. Self-Initiated Open World Learning for Autonomous AI Agents. arXiv:2110.11385, 2021.

Lifelong/continual learning in the open world

(Chen & Liu, 2018, Liu, 2020)



Liu. Learning on the Job: Online Lifelong and Continual Learning. AAAI-2020

Chen and Liu. Lifelong machine learning. Morgan & Claypool. 2015, 2018

Characteristics of open world continual learning

(Chen and Liu, 2018, Liu, 2020)

- **Continuous and incremental learning process** (no forgetting)
- **Knowledge transfer/adaptation** (across tasks)
- **Knowledge accumulation in KB** (long-term memory)
- **Learning on the job (after model deployment)**. *Self-supervision* using the *accumulated knowledge* and *interaction* with **humans** & **environment.**

Main steps:

- Identify new tasks to learn (open-world learning or OOD detection)
- Acquire ground-truth training data (collecting training data interactively)
- Learn the tasks incrementally (continual learning)

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 on the job (after model deployment)

(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.
    - Have to learn continually and adapt
  - Without this capability, an AI agent is not truly intelligent.

Liu. Learning on the Job: Online Lifelong and Continual Learning. AAAI-2020
Chen and Liu. Lifelong machine learning. Morgan & Claypool. 2015, 2018

Example - a greeting bot in a hotel

(Chen and Liu, 2018; Liu et al., 2021)

- See an existing guest.
  - Bot: "Hello John, how are you today?"
- See a new guest - recognize he/she is new **(OOD and create a new task)**
  - Bot: "Welcome to our hotel! What is your name, sir?" **(get class label)**
  - Guest: "David" **(got class label: David)**
  - Bot learns to recognize David automatically
    - take pictures of David **(get training data)**
    - learn to recognize David **(continual learning)**
- See David next time.
  - Bot: "Hello David, how are you today?" **(use the new knowledge)**

Liu, Robertson, Grigsby, and Mazumder. Self-Initiated Open World Learning for Autonomous AI Agents. arXiv:2110.11385, 2021.

Chen and Liu. Lifelong machine learning. Morgan & Claypool. 2015, 2018.

Example: novelty, characterization and response

- First, if the bot uses a video camera, when it sees a **novel object**, it experiences a data distribution change.
  - Detect the new object as quickly as possible.
- Next, how does the system know that the novel object is actually a person, not a dog?
  - If the system can recognize the object as a person, how does it know that he/she is a hotel guest, not a policeman?
- In order to **respond** to the novel object, the system must first **characterize** the novel object.
  - Without it, the agent does not know how to respond.

Liu, Robertson, Grigsby, and Mazumder. Self-Initiated Open World Learning for Autonomous AI Agents. arXiv:2110.11385, 2021.
Chen and Liu. Lifelong machine learning. Morgan & Claypool. 2015, 2018.

Example: characterization, response relevance and risk

- Classification is needed to decide if it is a person with luggage.
  - If the object is a person but has no luggage, the bot may not respond or learn to recognize the person
    - it is **irrelevant** to its performance task.
- If it is a moving object but not a person, it may notify a hotel employee and possibly learn to recognize it.
  - For each **characterization**, there is a corresponding **response**, which can be **NIL** (i.e., **do nothing**)
- **Risk:** there is a risk involved if an incorrect decision is made.

Liu, Robertson, Grigsby, and Mazumder. Self-Initiated Open World Learning for Autonomous AI Agents. arXiv:2110.11385, 2021.
Chen and Liu. Lifelong machine learning. Morgan & Claypool. 2015, 2018.

Self-Initiated Open World Learning (SiOWL)

- A primary performance task with a pair of key modules (*T, S*),
  - *T* is the primary task-performer (e.g., the dialogue system of the greeting bot)
  - *S* is a set of supporting or peripheral functions (e.g., the vision system and the speech system of the bot) that supports the primary task-performer.
- The primary task-performer as well as each support function may consist of four main sub-components (*L, E, N, I*),
  - where *L* is a SiOWL learner, *E* is the executor that performs the task or function, *N* is a novelty characterizer for characterizing each novelty so that *E* can formulate an appropriate response to the novelty, and *I* is an interactive module for the agent or *L* to communicate with humans or other agents (e.g., to gain ground-truth training data)

Classic data shifts + novel objects

(Moreno-Torres et al. 2012)

Definition (covariate shift). *Covariate shift* refers to the distribution change of the input variable $\boldsymbol{x}$ between training and test phases, i.e., $P_{tr}(y|\boldsymbol{x}) = P_{te}(y|\boldsymbol{x})$ and $P_{tr}(\boldsymbol{x}) \neq P_{te}(\boldsymbol{x})$, where P_{te} is the test distribution.

Definition (prior probability shift). *Prior probability shift* refers to the distribution change of the class variable y between training and test phases, i.e., $P_{tr}(\boldsymbol{x}|y) = P_{te}(\boldsymbol{x}|y)$ and $P_{tr}(y) \neq P_{te}(y)$.

Definition (concept drift). *Concept drift* refers to the change in the posterior probability distribution between training and test phases, i.e., $P_{tr}(y|\boldsymbol{x}) \neq P_{te}(y|\boldsymbol{x})$ and $P_{tr}(\boldsymbol{x}) = P_{te}(\boldsymbol{x})$.

We only focus on the appearance of novel objects

+ Novel objects

Moreno-Torres, Raeder, Alaiz-Rodriguez, Chawla and Herrera. A unifying view on dataset shift in classification. Pattern recognition 45(1), 2012

Novelty detection

- Novelty is agent-specific. World knowledge is the training data

$\mathcal{D}_{tr} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ with $\boldsymbol{x}_i \in X$ and $y_i \in Y_{tr}$. Let $h(\boldsymbol{x})$ be the latent or internal representation of $\boldsymbol{x}$ in the agent's mind, $h(D_{tr}^i)$ be the latent representation of the training data of class y_i, and k $(= |Y_{tr}|)$ be the total number of training classes. We use $\mu(h(\boldsymbol{x}), h(D_{tr}^i))$ to denote the novelty score of a test instance $\boldsymbol{x}$ with respect to $h(D_{tr}^i)$. The degree of novelty of $\boldsymbol{x}$ with respect to D_{tr}, $\mu(h(\boldsymbol{x}), h(D_{tr}))$, is defined as the minimum novelty score with regard to every class,

$$\mu(h(\boldsymbol{x}), h(D_{tr})) = \min(\mu(h(\boldsymbol{x}), h(D_{tr}^1)), ..., \mu(h(\boldsymbol{x}), h(D_{tr}^k)))$$

Novelty detection (cond.)

Definition (novel instance): A test instance $\boldsymbol{x}$ is novel if its novelty score $\mu(\boldsymbol{x}, .)$ is greater than or equal to a threshold value γ such that $\boldsymbol{x}$ can be assigned a new class not in Y_{tr}.

Definition (novel class): A newly created class y_{new} ($y_{new} \notin Y_{tr}$) assigned to some novel instances is called a *novel class* (*unknown* or *unseen class*). The classes in Y_{tr} are called *known* or *seen classes*.

Self-initiated Open World Learning

Definition (closed-world assumption): No new or novel classes appear in testing. Other types of data shift may occur.

Definition (closed-world learning): It refers to the learning paradigm that makes the closed-world assumption.

Definition (learning with data shift): It refers to the learning paradigm that deals with certain types of data shift in testing or deployment, but not new classes.

Definition (open world learning): It refers to the learning paradigm that can detect data shifts and novel instances in testing or deployment. The learner can learn the novel classes labeled by humans from the identified novel instances and update the model using the new data. The retraining or model updating is initiated by human engineers. That is, there is no self-initiated learning after model deployment.

- **Self-initiated open world learning**): Open world learning that can initiate a new learning process on its own during application (after deployment) with no involvement of human engineers.

Steps in Self-Initiated Open World Learning

- **Step 1 - Novelty detection.** Detecting data instances (1) whose classes do not belong to Ytr or (2) have covariate shift, and it is done automatically.
- **Step 2 - Acquiring class labels and creating a new learning task**: Clustering the detected novel instances. Each cluster represents a new class. It may be done automatically or through interactions with humans using the interaction module *I*.
- **Step 3 - Incrementally learn the new task**. After ground-truth training data is obtained, the learner *L* incrementally learns the new task. This is lifelong/continual learning (Chen and Liu, 2018)

Chen and Liu. Lifelong machine learning. Morgan & Claypool. 2015, 2018.

Novelty Characterization and Response

Definition (novelty and response): It is a pair (c, r), where c is the *characterization of the novelty* and r is the *response* to the novelty, which is a plan of dynamically formulated actions based on the characterization of the novelty and the agent's interactions with the novel item. If the system cannot characterize a novelty, it will take a *default response*. In some situations, the agent does not know what to do. The response is *Learn*, i.e., to learn the actions to take.

Definition (characterization of novelty): It is a description of the novelty, according to which the agent chooses a specific course of actions to respond to the novelty.

Learning to respond

- **Asking a human.** E.g., in a self-driving car, when it does not know what to do, it may ask the passenger using the interactive module ***I*** (e.g., in natural language) and then follow the instruction and also remember or learn it for future use.
- **Imitation learning**. E.g., on seeing a novel object, if the car in front drives through it with no issue, the car may choose the same course of action as well and learn it for future use.
- **Performing reinforcement learning**. By interacting with the environment through trial-and-error exploration, the agent learns a good response policy for future use.

Risk

- There is risk in achieving performance goals of an agent when making an incorrect decision.
  - E.g, classifying a known guest as unknown negatively affect guest impressions.
  - For a self-driving car, misidentifications can result in wrong responses: life and death.
- Thus, risk assessment must be made in making each decision.
- Risk assessment can be learned from experiences or mistakes.
  - E.g., a car passing over tar, after the experience of passing over shiny black surfaces safely many times, if the car slips in one instance, the car agent must assess the risk of continuing the prior procedure.
  - Given the danger, a car may weight the risk excessively, slowing down on new encounters of shiny black surfaces.

Example SiOWL: natural language interface (NLI)

- An application-independent approach to building task-oriented chatbots with interactive continual learning.
  - Based on *natural language to natural language* matching (**NL2NL**)
  - Our system CML builds NLIs for API-driven applications semi-automatically.
- To build a new NLI (or add a new skill to an existing NLI),
  - The application developer only needs to write a set S_i of seed commands (SCs) in NL to represent each action *i*.
    - SCs in S_i are just like paraphrased NL commands from users to invoke *i*, but the objects to be acted upon in each SC are replaced with variables, the arguments of *i*.
  - An interactive learning mechanism to enable CML to continually learn new (paraphrased) SCs from users.

Mazumder, Liu, Wang, and Esmaeilpour. An Application-Independent Approach to Building Task-Oriented Chatbots with Interactive Continual Learning. *NeurIPS-2020 Workshop on Human in the Loop Dialogue Systems (HLDS-2020)*

An example

- Microsoft Paint tool: The API action

  drawCircle(X1, X2)

  - drawing a circle having color X1 at coordinate X2 .
- Let an SC be "draw an X1 circle at X2" for this API,
  - where X1 and X2 are variables representing the arguments of the API.
- User command: "draw a blue circle at (20, 40)"
  - It can be matched or grounded to this SC, where the grounded API arguments are X1 = 'blue' and X2 = (20, 40).

CML has three components

- SC (seed command) specification
  - to enable the 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 to learn new SCs and paraphrases of API argument values.

Continual interactive learning

- If CML does not understand a user commend C.
- CML learns a new SC from the user command C through interactive dialogue
- It also learns new paraphrased argument values from in *C* to improve repheaser *R* over time.

Algorithm 2 Interactive Knowledge Learning

Input: C': Reduced user command by Algorithm 1; $\mathcal{T}$: action and utility SC Store; $\mathcal{Q}$: Question Template Store; $\mathcal{M}$: SC Matcher;

1: $r_1 \leftarrow$ **Verify_Pred_SC**($\mathcal{Q}$, C') {r_i is user's response}
2: **if** $r_1 =$ "no" **then**
3: $r_2 \leftarrow$ **ShowSC_List**(C_{rnk}) {C_{rnk} is the action SC rank list returned by $\mathcal{M}$}
4: **end if**
5: **for all** variable x_i in r_2 **do**
6: $r_{expr} \leftarrow$ **Ask_Ref_Expr**(x_i, C)
7: $r_{prop} \leftarrow$ **Ask_Ref_Prop**(r_{expr})
8: $r_{val} \leftarrow$ **Choose_Prop_Val**(r_{prop}, r_{expr})
9: $r_{para} \leftarrow$ **Ask_Para_Expr**(r_{val}, r_{expr})
10: Update $\mathcal{R}$ with all (r_{val}, r_{para}) pairs
11: **end for**
12: Rephrase C to get a new SC and update $\mathcal{T}$

Continuous knowledge learning in dialogues

(Mazumder et al. 2018, 2019)

- Dialogue systems are increasingly using knowledge bases (KBs) storing real-world facts to help generate responses.
  - KBs are inherently incomplete and remain fixed,
  - which limit dialogue systems' conversation capability
- CILK: ***Continuous and Interactive Learning of Knowledge*** for dialogue systems
  - 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

Opportunities to learn in conversations

1. Extracting knowledge directly from user utterances. E.g.,
   - User: Obama was born in Hawaii.
   - Agent extracts: (Obama, BornIn, Hawaii) – expressed in triples **(h, r, t)**
2. Asking user questions & expecting correct answers, e.g.,
   - Agent: Where was Obama born?
   - User: Hawaii => (Obama, BornIn, Hawaii)
3. When the agent cannot answer user questions, it asks the user for some supporting facts and then infers the answers.
   - We focus on this setting (which covers 1 and 2)

(2) Continual learning of factual knowledge

- When a user query cannot be answered, chatbot learns new facts.

Knowledge Base

$\mathcal{K}$

Stores acquired Facts (Triples)

KB: Collection of Triples

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

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

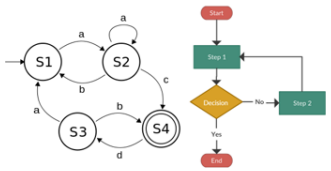
Triple Store

(Boston, *LocatedInCountry*, USA)

head *relation* *tail*

or

Knowledge Graph

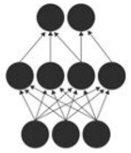


Interaction Module

$\mathcal{I}$

Interacts with user to acquire Facts

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



Inference Module

$\mathcal{M}$

Infers new Knowledge to answer user's query

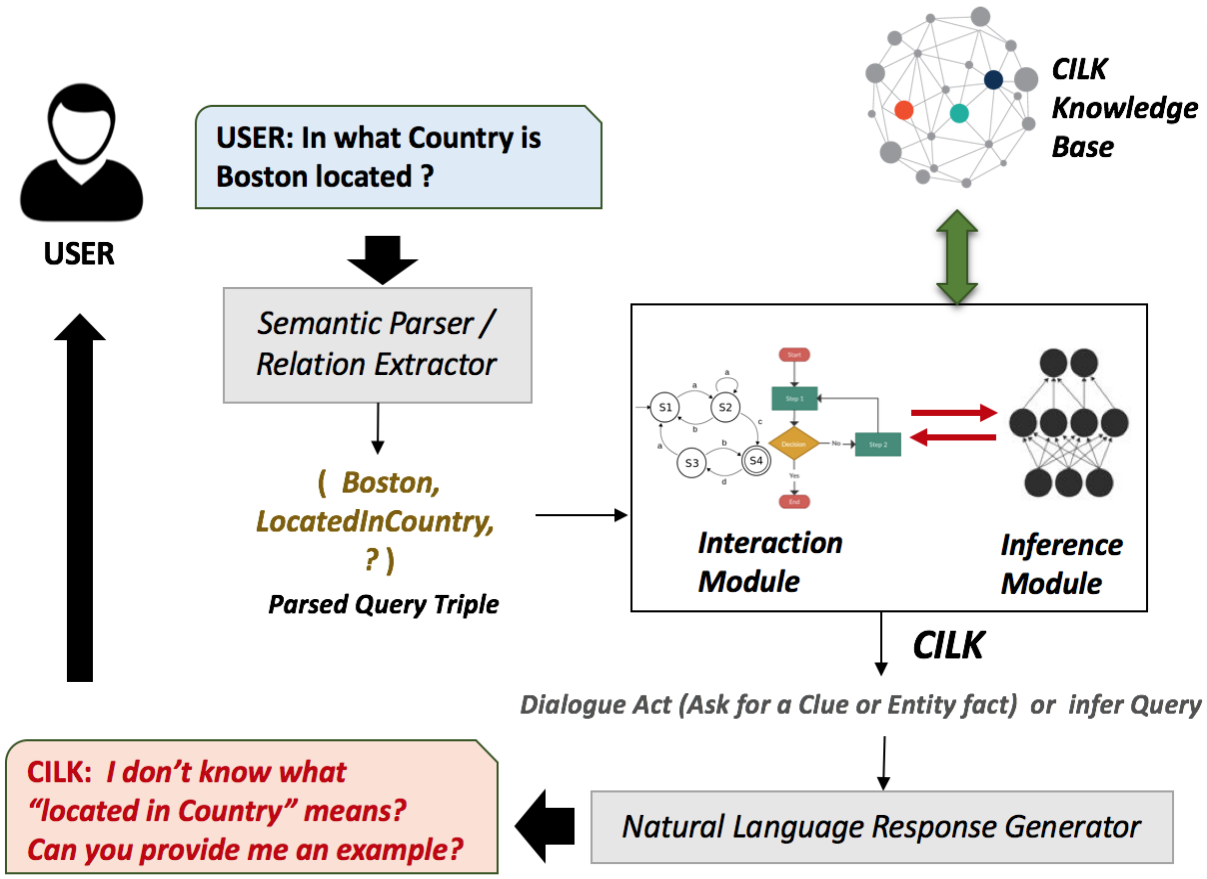
- Performs inference over the acquired Facts and existing KB

Mazumder, Ma, and Liu. Towards a Continuous Knowledge Learning Engine for Chatbots. arXiv:1802.06024 [cs.CL], 16 Feb. 2018

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

Interactive knowledge learning in dialogue

(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

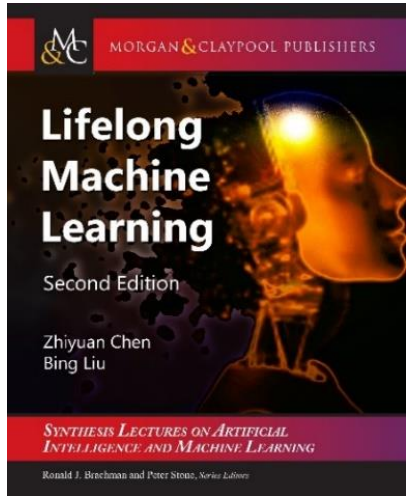
Challenges of SiOWL

- AI agents must learn continuously in the open world by itself
  - Learning on the job (Chen and Liu, 2018; Liu, 2020)
- Current techniques are in their infancy. Many challenges:
  - Novelty detection and new task formulation
  - Obtaining training data on the fly.
  - Few-shot continual learning.
  - Novelty characterization.
  - Correctness and applicability of learned knowledge.
  - Interactive learning.

Liu, Robertson, Grigsby, and Mazumder. Self-Initiated Open World Learning for Autonomous AI Agents. arXiv:2110.11385, 2021.
Chen and Liu. Lifelong machine learning. Morgan & Claypool. 2015, 2018.

Thank You

Q&A



Book: https://www.cs.uic.edu/~liub/2018-LML-2nd-edition.pdf

Research: https://www.cs.uic.edu/~liub/lifelong-learning.html