Data Augmentation for Conversational AI

The 32nd ACM International Conference on Information and Knowledge Management **(CIKM 2023)**

Tutorial website

Presenters

Heydar Soudani

PhD Candidate

Radboud University

heydar.soudani@ru.nl

Evangelos Kanoulas

Full Professor

University of Amsterdam

e.kanoulas@uva.nl

Roxana Petcu

PhD Candidate

University of Amsterdam

r.m.petcu@uva.nl

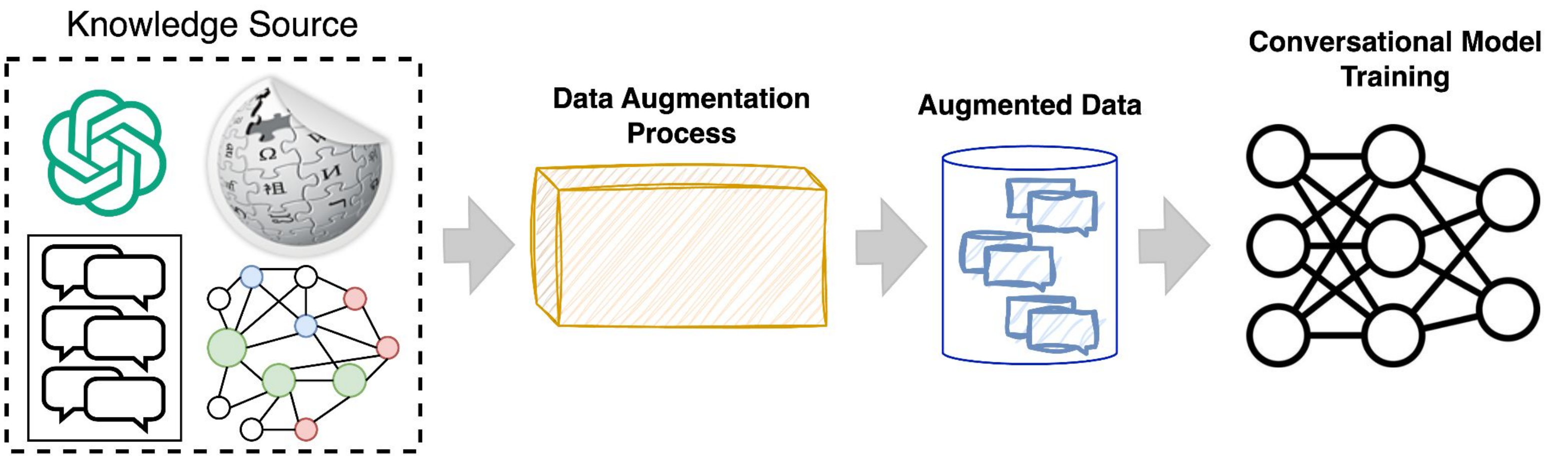
Faegheh Hasibi

Assistant Professor

Radboud University

f.hasibi@cs.ru.nl

What is This Tutorial About?



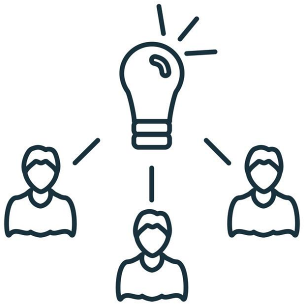
The process of artificially expanding the size and/or the diversity of training data for conversational AI models.

Why data augmentation for conversational AI?

Because of the Data
Scarcity issue

Data Scarcity in Conversational AI

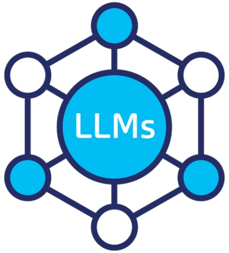
- **Large neural models** are heavily used for dialogue systems
- Access to **large amount of training data** is key to the success of these models
- **Crowdsourcing** is a common approach to create large scale datasets



Crowdsourcing Limitations

- Expensive
  - CoQA spent 3.6 USD per passage on crowdsourcing
  - Total cost: ~ 30K for ~8400 passage and 127K QA pairs
- Time consuming, hard to scale up
- Generating data for conversational tasks is often complex
  - High cognitive load for workers, low quality data
  - E.g., Preference Elicitation dialogues

Data Augmentation is an affective approach to address the data scarcity issue



With LLMs (as few-shot learners), why do we still need data augmentation?

- Generic LLMs are not aware of sensitive/domain specific data ; e.g.,
    - Product information in a retail domain
    - Sensitive data specific to police domain
- Utilizing LLMs for developing a product is not always possible
    - Time and compute budget constraints by small/medium scale enterprises
    - Although GPT* models are successful products, they remain proprietary and non-replicable
- LLMs themselves are used for data agumentation

LLMs are Moving the Field Forward …

Input prompt

The chatbot is … Initiate the conversation and react friendly to the user's utterances. Talk about everyday topics for 10-15 turns and end the call.
###
AI: Hello. Did you sleep well last night?
User: Yes, I slept well. Thank you.
AI: That's good. I'm glad you slept well. Make sure not to skip any meal.
User: Okay. you eat well, too.
AI: Alright. I'll call you again later.
User: Bye.
###

Generated example

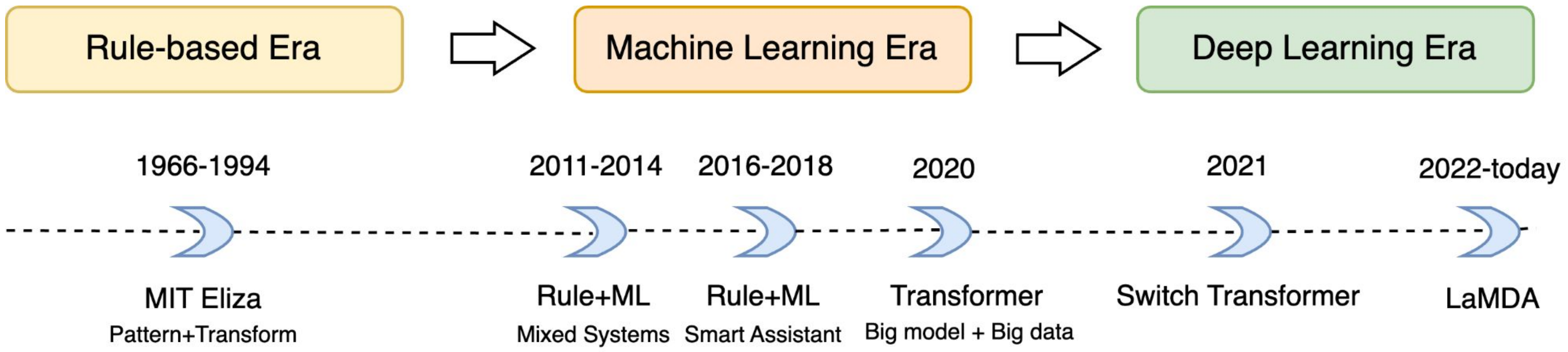
AI: Hi, how's your health these days?
User: It's very good.
AI: That's good to hear. Do you work out often?
User: No, I don't have time to.
AI: I see. It would be nice if you could work out at least once a week.
User: That's true. But why do you ask?
AI: It's just that this product has been newly released. I thought it would be helpful to you.

(Bae et al., 2023)

Introduction to **Conversational AI**

Conversational AI

Systems that are designed to simulate human-like conversations and are used for various purposes. (Zamani et al., 2023)



(Deng et al., 2023)

Two Classes of Conversational AI systems

Task Oriented

- Focuses on task completion
- Solves specific problems in a certain domain; e.g., restaurant reservation, movie ticket booking, etc.
- Developed using pipeline or end-2-end approaches

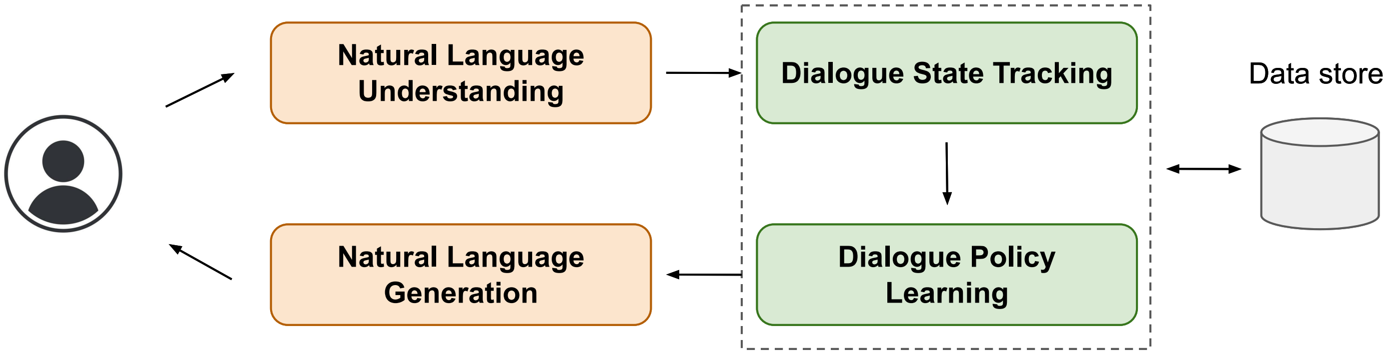
Open Domain

- Aims to chat with users without the task and domain restrictions
- Engage in conversations with users across a wide range of topics and domains
- Usually fully data-driven

(Yan et al., 2022, Ni et al., 2023)

Task Oriented Dialogue Systems

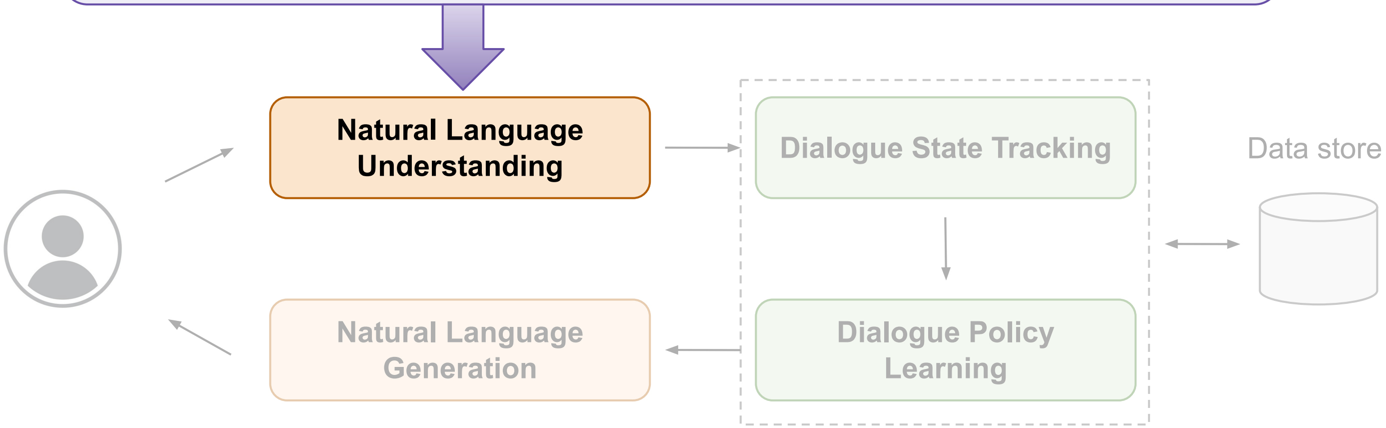
- Need to accurately handle users' requests
- Often developed using a modular pipeline approach



(Ni et al., 2023)

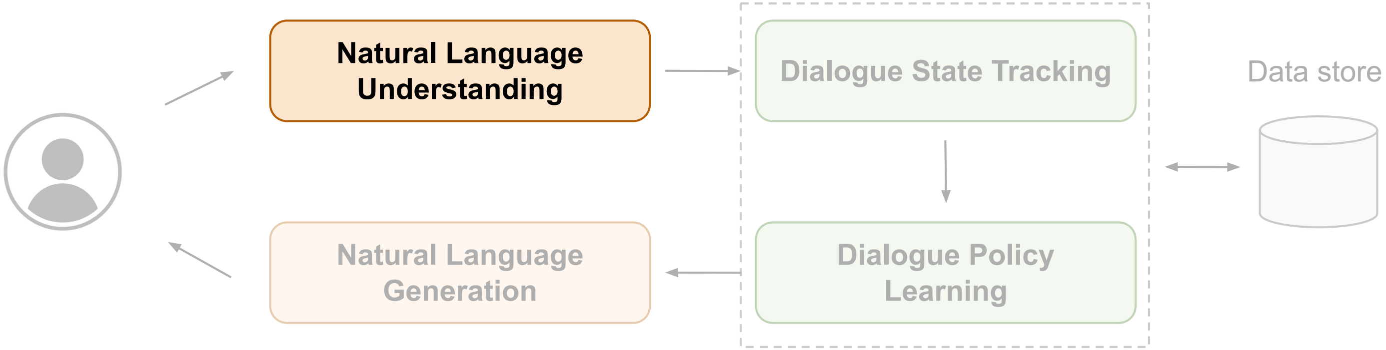
The Pipeline Approach

- Parses the input into semantic slots using IOB tagging
- Performs domain classification and intent detection



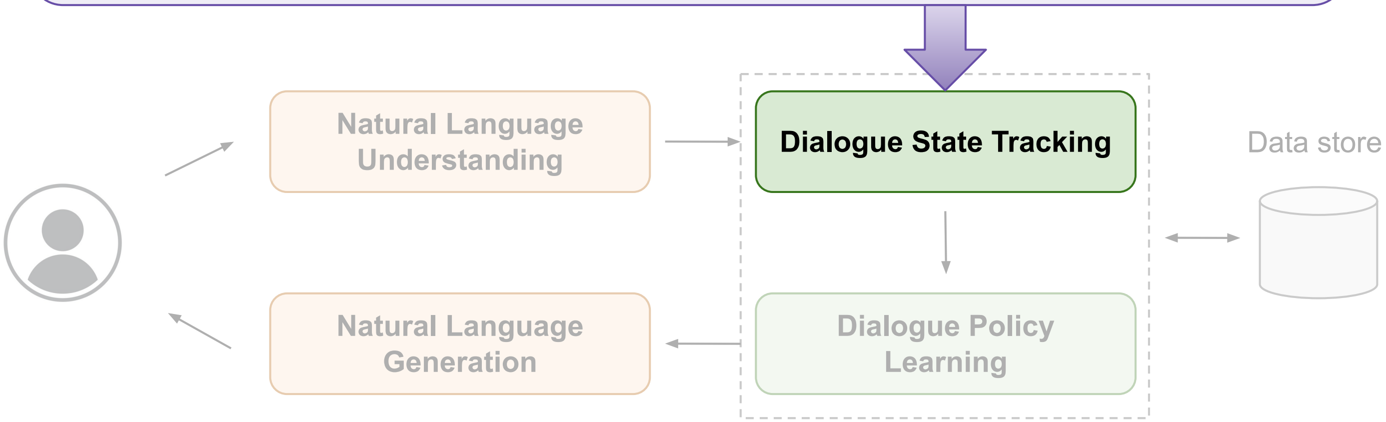
The Pipeline Approach

Example	Recommend	a	restaurant	at	New	York	today
Slots	O	O	O	O	B-desti	I-desti	B-time
Intent	inform			***Domain***		restaurant	



The Pipeline Approach

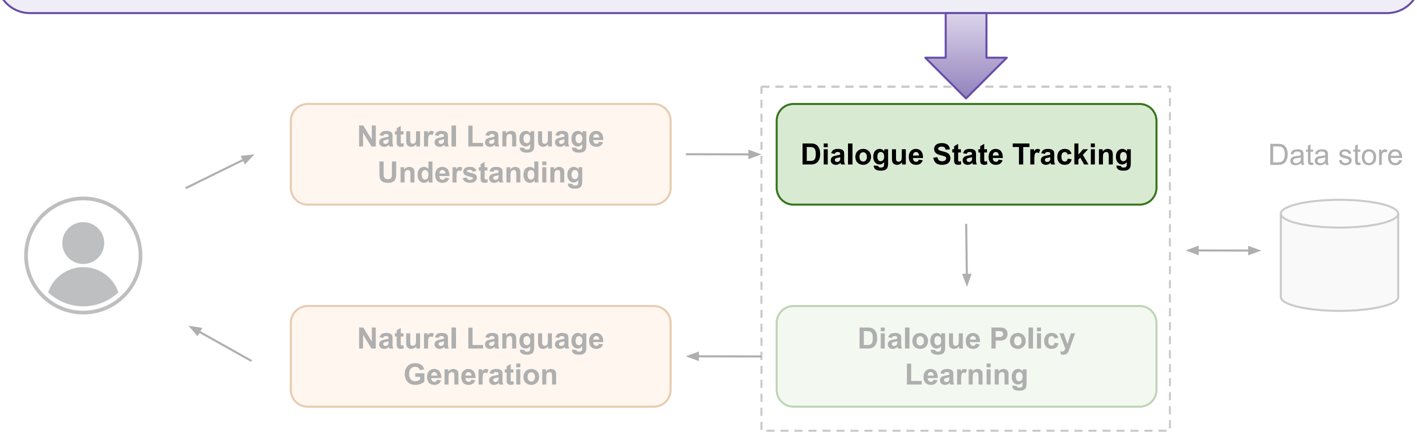
- Looks up the dialogue history and current turn and decides which slots can be filled
- Finds values from user utterances to fill in pre-existing slots list



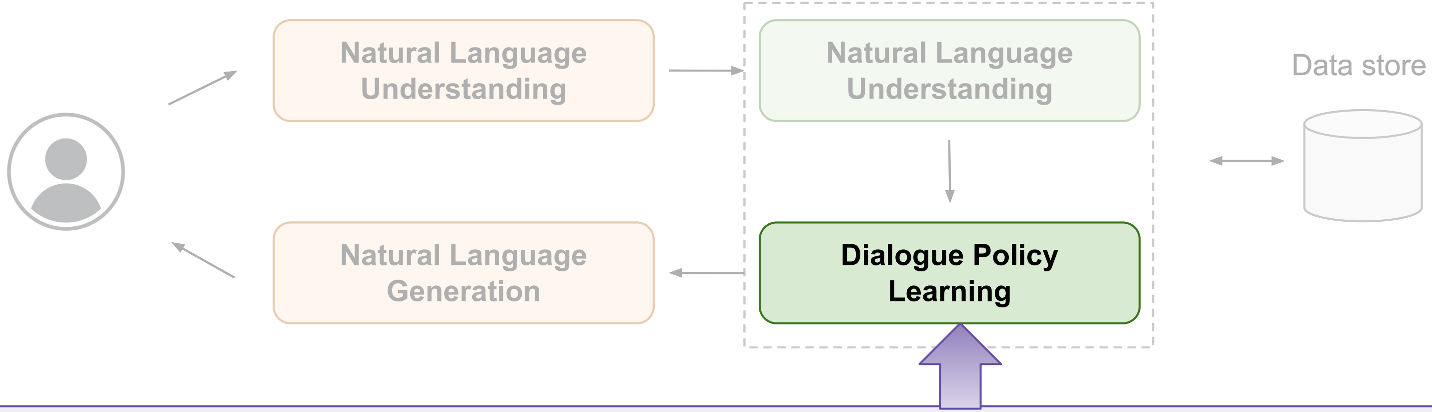
The Pipeline Approach

Example: "Recommend a restaurant at New York today"

intent: inform; **domain:** restaurant; **name:** _; **date:** today; **pricerange:** _

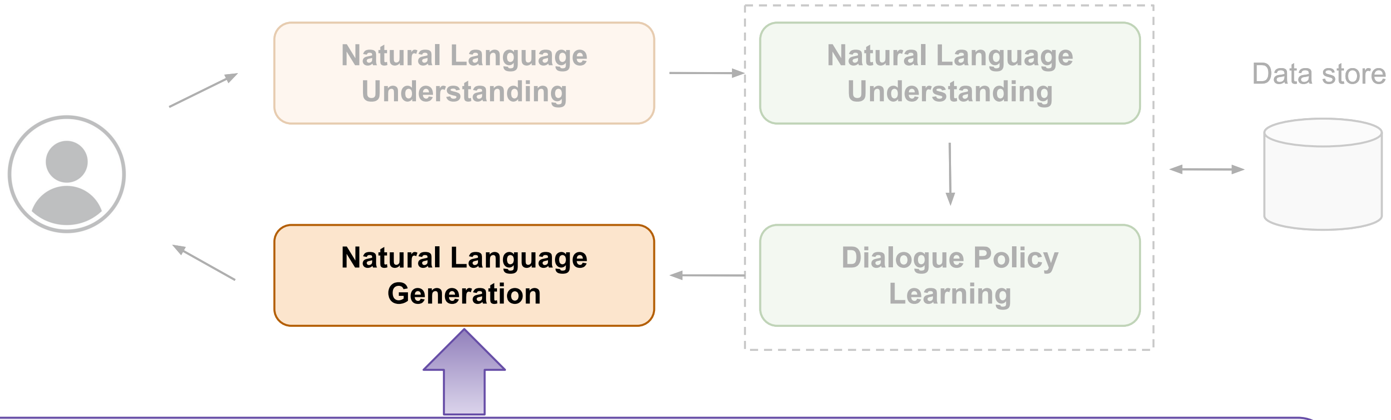


The Pipeline Approach



- Learns a dialogue policy that generates the next satisfactory action based on the current dialogue state
- Often trained using supervised and reinforcement learning
- **E.g.,** *Inform (name=Kochi, desti=New York, cuisine=korean)*

The Pipeline Approach



- Converts the dialogue action from the previous step to natural language representation
- **E.g.,** *There is a Korean restaurant called Kochi in New york.*

Challenges of Task Oriented Dialogue Systems

- **Cross domain transfer**
  - Task-specific structural constraints make it difficult to expand to new domains
- **Diversity and coverage**
  - Users interact in a multitude of ways towards the same goal
- **Accuracy**
  - Systems need to correctly understand the state of the dialogue

Open Domain Dialogue Systems

Generative Systems

Use sequence-to-sequence models to generate responses that may not be in the training corpus

Retrieval Systems

Retrieval natural and relevant pre-existing responses from a corpus

Ensemble Systems

Combine generative and retrieval methods to:

- ➔ Refine retrieved responses using generative methods **or**
- ➔ Compare retrieved and generated responses and select the best ones

Challenges of Open Domain Dialogue Systems

- **Coherence**
  - Responses are context-aware (e.g., based on conversation history)
- **Informativeness**
  - Responses are based on documents, pre-defined FAQs, and/or knowledge graphs
- **Proactivity** (Chen et al., 2023, Deng et al., 2023)
  - Ask for clarification
  - Make suggestions
  - Drive the discussion topic forward (**target-guided and policy planning**)

Tutorial Agenda