Natural Language Processing

Lecture 12: Chatbots

András Simonyi

Eötvös University, Department of Artificial Intelligence

2023

Acknowledgement

Acknowledgement

The following slides are based on and very closely follow the online freely available Chapter 24 of Jurafsky and Martin's *Speech and Language Processing 3rd ed.* (2019), please read the original!

Dialogue system types

Task-oriented dialogue agent

The goal of the dialogue is to complete a task or tasks in a predefined task set, e.g., order something, make a call, transfer money, get directions etc.

Chatbot

- The goal is open-ended and unstructured, extended conversation.
- There is no predetermined task (or set of tasks) whose successful execution would be the goal.
- The main result in many cases is simply “entertainment”.
- Can be additional components of mainly task-oriented systems.

Chatbot requirements

The system needs to be able to reproduce the important features of human-human conversations, among others

- **grounding**: there is a constantly evolving **common ground** established by the speakers who constantly acknowledge understanding what the other said.
- **adjacency pairs**: different utterance types are associated with different expectations as to the next utterance:
  - question $\Rightarrow$ answer
  - proposal $\Rightarrow$ acceptance
  - compliment $\Rightarrow$ downplayer etc.
- **inferences** based on the assumption of utterances being
  - relevant,
  - informative,
  - truthful, and
  - clear and brief (or at least that the speakers aim at this).

Chatbot approaches

1. Rule-based

   Traditionally, rule-based, "pattern-match and substitute" type systems were used, famously

   - **Eliza** (1966), simulating a Rogerian psychologist, and
   - **PARRY** (1971), for studying schizophrenia.

2. Corpus-based

   The more modern alternative is, of course, to build a **corpus-based** system, which is trained on a data set containg a large number of dialogues.

Corpus-based chatbot architectures

- **Response by retrieval** Respond with the utterance in the data set that is
  - most similar to the last turn, or
  - is the response to the utterance which is most similar to the last turn.
  - similarity can be totally pretrained, or trained/fine-tuned embedding based.
- **Response by generation** Train a genator model on the data set, typical architectures:
  - RNN or Transformer based encoder-decoder, or
  - "Predict next", language-model, e.g., a GPT-like architecture.

Task oriented dialog agents

Slot	Type	Question Template
ORIGIN CITY	city	"From what city are you leaving?"
DESTINATION CITY	city	"Where are you going?"
DEPARTURE TIME	time	"When would you like to leave?"
DEPARTURE DATE	date	"What day would you like to leave?"
ARRIVAL TIME	time	"When do you want to arrive?"
ARRIVAL DATE	date	"What day would you like to arrive?"

Figure 24.10 A frame in a frame-based dialogue system, showing the type of each slot and a question used to fill the slot.

Figure 1: Traditional frame-based architecture (from Jurafsky and Martin 2019).

Task oriented dialog agents cont.

The task is to determine the domain, intent, and slot fillers for each user utterance. E.g, for

Show me morning flights from Boston to San Francisco on Tuesday

We want the analysis

DOMAIN	AIR-TRAVEL
INTENT	SHOW-FLIGHTS
ORIGIN-CITY	Boston
ORIGIN-DATE	Tuesday
ORIGIN-TIME	morning
DEST-CITY	San Francisco

Dialog-state systems

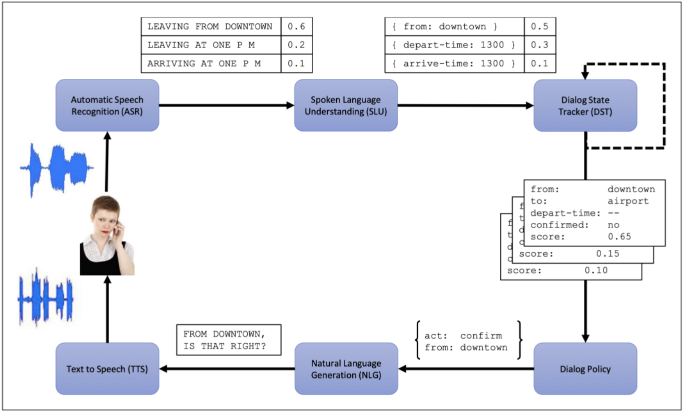

Figure 24.12 Architecture of a dialogue-state system for task-oriented dialogue from Williams et al. (2016a).

Figure 2: From Jurafsky and Martin (2019).

Dialog-state systems cont.

	Utterance	Dialogue act
U:	Hi, I am looking for somewhere to eat.	hello(task = find,type=restaurant)
S:	You are looking for a restaurant. What type of food do you like?	confreq(type = restaurant, food)
U:	I'd like an Italian somewhere near the museum.	inform(food = Italian, near=museum)
S:	Roma is a nice Italian restaurant near the museum.	inform(name = "Roma", type = restaurant, food = Italian, near = museum)
U:	Is it reasonably priced?	confirm(pricerange = moderate)
S:	Yes, Roma is in the moderate price range.	affirm(name = "Roma", pricerange = moderate)
U:	What is the phone number?	request(phone)
S:	The number of Roma is 385456.	inform(name = "Roma", phone = "385456")
U:	Ok, thank you goodbye.	bye()

Figure 24.14 A sample dialogue from the HIS System of Young et al. (2010) using the dialogue acts in Fig. 24.13.

Figure 3: From Jurafsky and Martin (2019).

Dialog-state systems cont.

In modern implementations,

- The **NLU** (natural language understanding) component can be implemented by text classifiers (domain, intent) and sequence labeling (slot/entity detection) models,
- The **Dialog State Tracker** uses the NLU module to keep track of slot values and type of dialogue act that was performed.
- The **Dialogue Policy** decides which action to take next, on the basis of the current dialogue state and maybe history – this is also a classifier.
- Finally, the **NLG** (natural language generation) component generates the actual utterance based on the required action and the dialog state. This can be rule/template-based, or by an encoder-decoder model, which possibly only produces a delexicalized template.

End-to-end trained dialog-state systems cont.

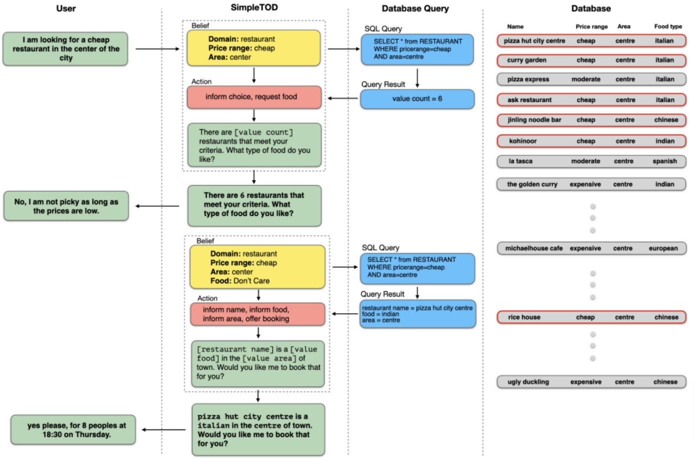

Figure 4: From Hosseini-Asl et al. (2020).

End-to-end trained dialog-state systems cont.

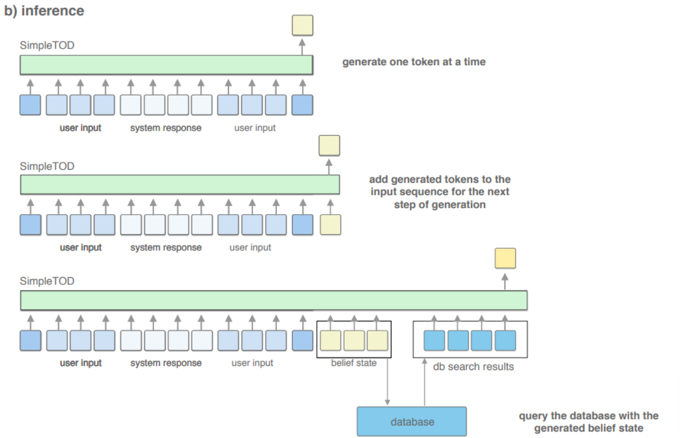

Figure 5: From Hosseini-Asl et al. (2020).

References

Hosseini-Asl, Ehsan, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. "A Simple Language Model for Task-Oriented Dialogue." *Advances in Neural Information Processing Systems* 33: 20179–91. https://arxiv.org/abs/2005.00796.

Jurafsky, Daniel, and James H Martin. 2019. "Speech and Language Processing (3rd Ed. Draft)." https://web.stanford.edu/~jurafsky/slp3/.