Dialogue Systems

AI in HCI

Ondřej Dušek

24. 3. 2023

Charles University
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics

What’s Conversational AI = Dialogue System?

- Definition: A *(spoken)* dialogue system is a **computer system designed to interact** with users **in** *(spoken)* **natural language**
  - Wide – covers lots of different cases
    - “smart speakers” / phone OS assistants
    - phone hotline systems (even tone-dial ones)
    - in-car systems
    - assistive technologies: therapy, elderly care, companions
    - entertainment: video game NPCs, chatbots

- DSs are cool:
  - ultimate natural interface: say what you want
  - lots of active research – far from solved
  - already used commercially

https://www.digitaltrends.com/mobile/5-things-you-need-to-know-about-microsofts-chinese-girlfriend-chatbot-xiaoice/

Real-life dialogue systems: virtual assistants

- Google, Amazon, Apple & others, Mycroft, Rhasspy: open-source
- Really good microphones
  - and not much else – they work online only
- Huge knowledge bases
  - Google: combined with web search
- Lots of domains programmed in, but all by hand
  - integration with a lot of services (calendar, music, shopping, weather, news…)
  - you can add your own (with limitations)
- Can keep some context
- Conversational capabilities limited

Google Nest

Amazon Echo

Apple HomePod

Dialogue System Types

Task-oriented

- focused on completing a certain task/tasks
  - booking restaurants/flights, finding bus schedules, smart home…
- most actual DS in the wild
  - also our main focus in this talk
- (typically) **single/multi domain**
  - talk about 1/more topics

Non-task-oriented

- chitchat – social conversation, entertainment
  - persona, gaming the Turing test
- typically **open-domain** – talk about anything

Comm. Modes: voice / text / multimodal (face, graphics…)

Task-oriented Dialogue Example

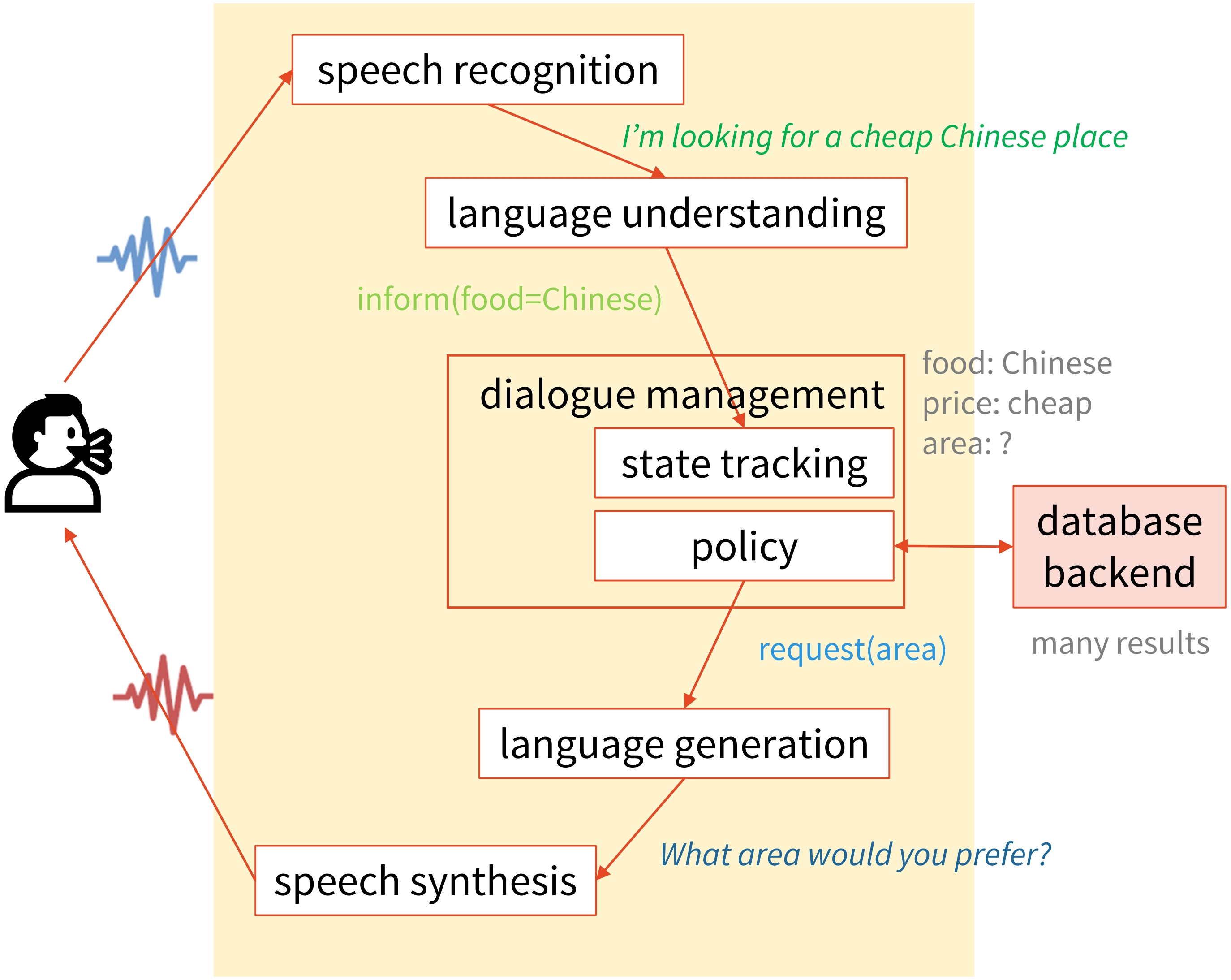
(Budzianowski et al., 2018)
https://aclanthology.org/D18-1547/

- MultiWOZ: benchmark for task-oriented dialogue in multiple domains
  - hotels, restaurants, attractions, trains, taxi, police, hospital
  - domains are connected (e.g. taxi from hotel to attraction)
  - 10k dialogues, extensive annotation (but noisy!)

1. user: I am looking for a train from Cambridge to London Kinks Cross.
   state: *{train {departure = cambridge, destination = london kings cross}}*
   DB: *{train (70) {…}}*
   system: There are 70 [count] trains departing from Cambridge [departure] to London Kings Cross [destination]. What day would you like to travel?
2. user: I would like to leave on Saturday after 18:45.
   state: *{train {day = saturday, departure = cambridge, destination = london kings cross, leave at = 18:45}}*
   DB: *{train (3) {arrive by = 19:51,21:51,23:51; id = TR0427,TR0925,TR4898; leave at = 19:00,21:00,23:00; … }}*
   system: TR0427 [id] leaves at 19:00 [leave_at] and arrives by 19:51 [arrive_by]. Would you like to book this train?
3. user: Yes, I would like to book it for eight people.
   system: I have booked it for you. Your reference number is 00000057 [reference]. Is there anything else I can help you with?
4. user: I am also looking for an expensive restaurant in the centre.
   belief: *{restaurant {area = centre, price range = expensive} train {...}}*
   DB: *{restaurant (33) {area = centre (33); name=Curry Garden, ...; ...}, ...}*
   system: There are 33 [count] expensive [price_range] restaurants in the centre [area]. Is there a particular type of food you would like?

Dialogue Systems Architecture

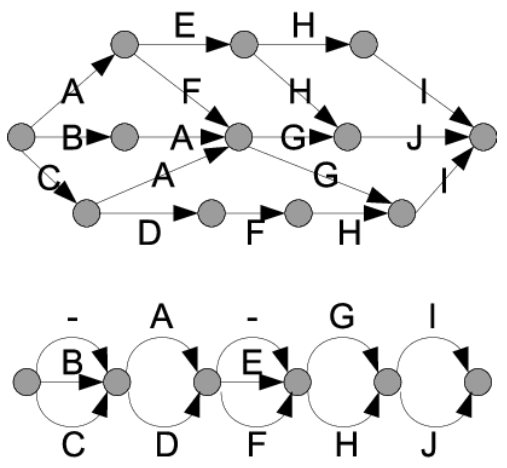
- traditional DS pipeline:
  - ASR: voice → text
  - **NLU: text → meaning**
  - **DM: meaning → reaction**
  - **NLG: reaction → text**
  - TTS: text → voice
- backend
  - needed for anything better than basic chit-chat
- text-based systems: NLU→DM→NLG only



Automatic Speech Recognition (ASR)

- Converting **speech signal** (acoustic waves) **into text**
- Typically produces several possible hypotheses with confidence scores
  - **n-best list**
  - lattice
  - confusion network
- Very good in ideal conditions
- **Problems:**
  - noise, accents, longer distance, echo cancellation, channel (phone)…

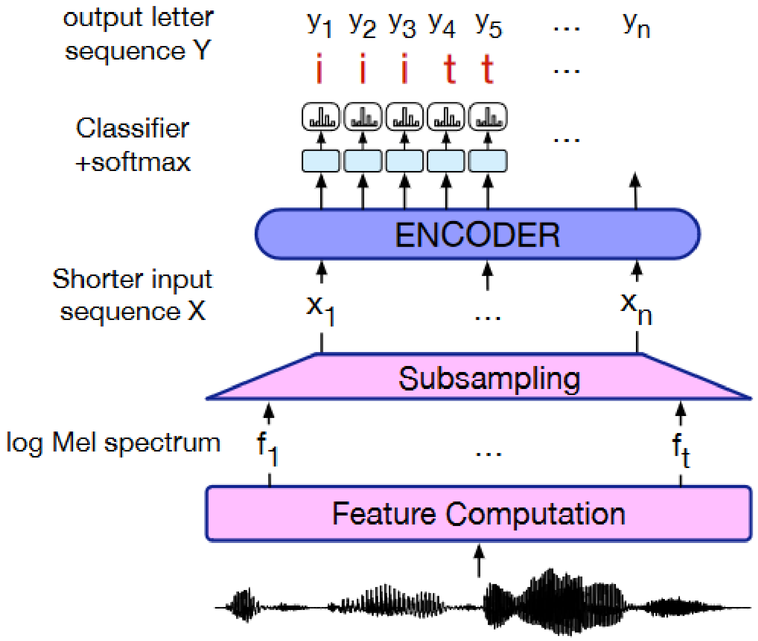
0.8 *I'm looking for a restaurant*
0.4 *uhm looking for a restaurant*
0.2 *looking for a rest tour rant*



(Kazemian et al., ICMR 2008)
https://doi.org/10.1145/1460096.1460112

Speech Recognition

- Also: **voice activity detection**
  - detect when the user started & finished speaking
  - **wake words** (“*OK, Google*”) – much simpler
- ASR implementation: mostly **neural networks**
  - take acoustic features (frequency spectrum)
  - compare with previous
  - emit phonemes/letters
- Limited domains: use of **language models**
  - some words/phrases more likely than others
  - previous context can be used
  - this can improve the experience **a lot**!
  - problem: out-of-vocabulary words



(Jurafsky & Martin, 2023)
https://web.stanford.edu/~jurafsky/slp3/16.pdf

Natural/Spoken Language understanding (NLU/SLU)

- **Extracting the meaning** from the (now textual) user utterance
- Converting into a structured semantic representation
  - **dialogue acts**:
    - act type/intent (*inform, request, confirm*)
    - slot/attribute (*price, time…*)
    - value (*11:34, cheap, city center…*)
    - typically intent detection + slot-value tagging
  - other, more complex – e.g. syntax trees, predicate logic
- Specific steps:
  - **named entity resolution** (NER)
    - identifying task-relevant names (*London, Saturday*)
  - **coreference resolution**
    - (“*it*” –> “*the restaurant*”)

inform(food=Chinese, price=cheap)
request(address)

Language Understanding

- Implementation varies
  - (partial) **handcrafting** viable for limited domains
    - keyword spotting
    - regular expressions
    - handcrafted grammars
  - **machine learning** – various methods
    - intent classifiers
    - slot tagging/classification
- Can also provide n-best outputs
- Problems:
  - recovering from bad ASR
  - ambiguities
  - variation

S: *Leaving Baltimore. What is the arrival city?*
U: *fine Portland* [ASR error]
S: *Arriving in Portland. On what date?*
U: *No not Portland Frankfurt Germany*

[On a Tuesday]
U: *I'd like to book a flight from London to New York for <u>next Friday</u>*

U: *Chinese city center*
U: *uhm I've been wondering if you could find me a restaurant that has Chinese food close to the city center please*

Dialogue Manager (DM)

- Given NLU input & dialogue so far,
  responsible for **deciding on next action**
  - keeps track of what has been said in the dialogue
  - keeps track of user profile
  - interacts with backend (database, internet services)
- Dialogue so far = **dialogue history**, modelled by **dialogue state**
  - managed by **dialogue state tracker**
- System actions decided by **dialogue policy**

Dialogue state / State tracking

- Stores (a summary of) dialogue history
  - User requests + information they provided so far
  - Information requested & provided by the system
  - User preferences
- Implementation
  - **handcrafted** – e.g. replace value per slot with last-mentioned
    - good enough in some circumstances
  - **probabilistic** – keep an estimate of per-slot preferences based on SLU output
    - more robust, more complex

price: cheap
food: Chinese
area: riverside

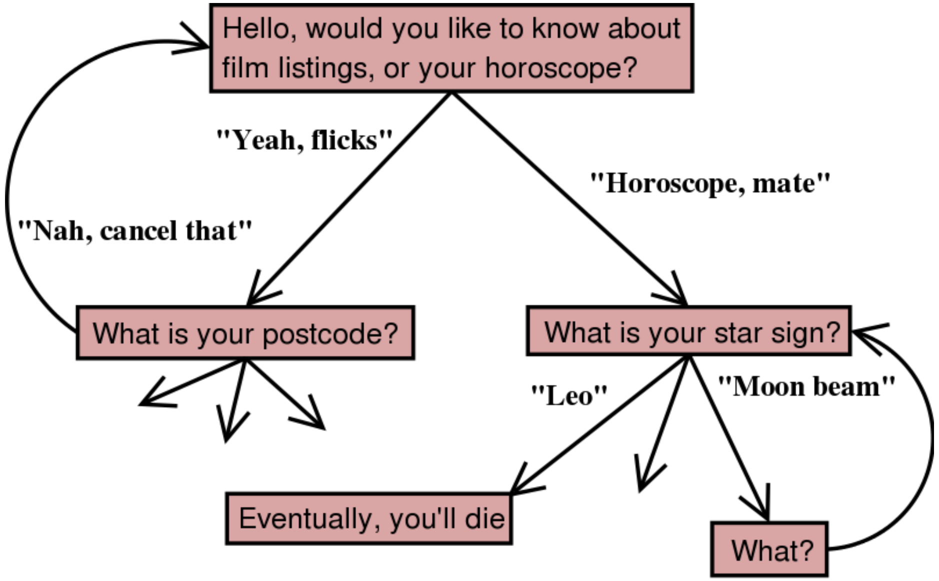
price: 0.8 cheap
0.1 moderate
0.1 <null>
food: 0.7 Chinese
0.3 Vietnamese
area: 0.5 riverside
0.3 <null>
0.2 city center

Dialogue Policy

- Decision on next system action, given dialogue state
- Involves backend queries
- Result represented as system dialogue act
- Handcrafted:
  - **if-then-else** clauses
  - **flowcharts** (e.g. VoiceXML)
- Machine learning
  - often trained with **reinforcement learning**
  - POMDP (Partially Observable Markov Decision Process)
  - recurrent neural networks

confirm(food=Chinese)

inform(name=Golden Dragon, food=Chinese, price=cheap)



https://www.w3.org/2004/Talks/05-www2004-voice/dialog.png

Natural Language Generation (NLG) / Response Generation

- Representing system dialogue act in natural language (text)
  - reverse NLU
- How to express things might depend on context
  - Goals: fluency, naturalness, avoid repetition (…)
- Traditional approach: **templates**
  - Fill in (=**lexicalize**) values into predefined templates (sentence skeletons)
  - Works well for limited domains

inform(name=Golden Dragon, food=Chinese, price=cheap)

\+

<name> is a **<price>**-ly priced restaurant serving **<food>** food

=

Golden Dragon is a cheaply priced restaurant serving Chinese food.

- Statistical approach: **seq2seq**/pretrained language models (→)
  - input: system dialogue act, output: sentence

Text-to-speech (TTS) / Speech Synthesis

- Generate a speech signal corresponding to NLG output
  - text → sequence of **phonemes**
    - minimal distinguishing units of sound (e.g. [p], [t], [ŋ] “ng”, [ə] “eh/uh”, [iː] “ee”)
  - + pitch/intonation, speed, pauses, volume/accents
- Standard pipeline:
  - text normalization
    - abbreviations
    - punctuation
    - numbers, dates, times
  - pronunciation analysis (**grapheme → phoneme conversion**)
  - (intonation/stress generation)
  - waveform synthesis
    - **concatenative** – record, cut into phoneme transitions (diphones), glue them together
    - **neural** – directly predict wave/spectrogram

take bus number 3 at 5:04am
take bus number three at five o four a m
t eɪ k b ʌ s n ʌ m b ə θ r iː æ t f aɪ v ə ʊ f ɔː r eɪ ɛm

https://en.wikipedia.org/wiki/MBROLA

https://google.github.io/tacotron/

Dialogue toolkits/Authoring tools

- Define your domain/inputs
  - intents, slots, values (~NLU), with examples
- Define your actions/responses
  - what happens on intent X? (e.g. call this function/provide pre-written reply)
- Toolkit does the rest
  - train NLU system
  - run the dialogue – call your actions/functions
- Some toolkits plug into voice assistants
  - Google Dialogflow, Alexa Skills Kit, Apple SiriKit, IBM Watson Assistant
- Some are standalone/offline
  - Rasa…

https://rasa.com/

https://dialogflow.cloud.google.com/

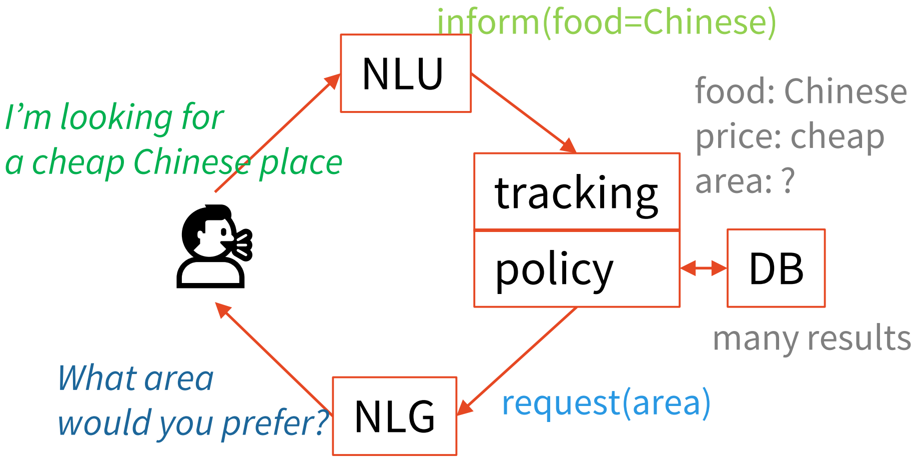
https://www.ibm.com/products/watson-assistant

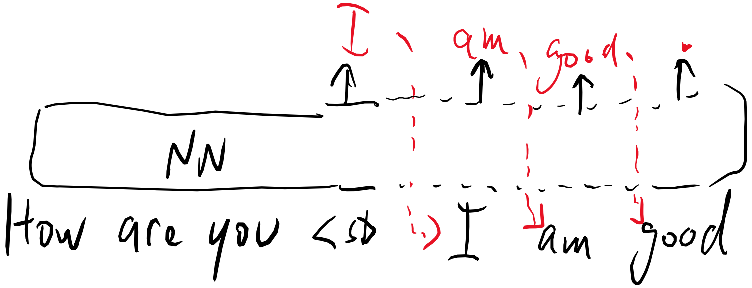
https://developer.apple.com/siri/

https://developer.amazon.com/alexa-skills-kit

End-to-end models (vs. separate components)

- Separate components (NLU→DM→NLG):
  - more flexible (replace one, keep the rest)
  - more explainable
  - trained separately, possibly optimization by RL
  - error accumulation
  - improved components don't mean improved system
- **End-to-end models**:
  - single neural network for NLU&DM&NLG
  - joint supervised optimization, RL still works
  - needs a lot of data
  - less control of outputs: hallucination, dull/repetitive

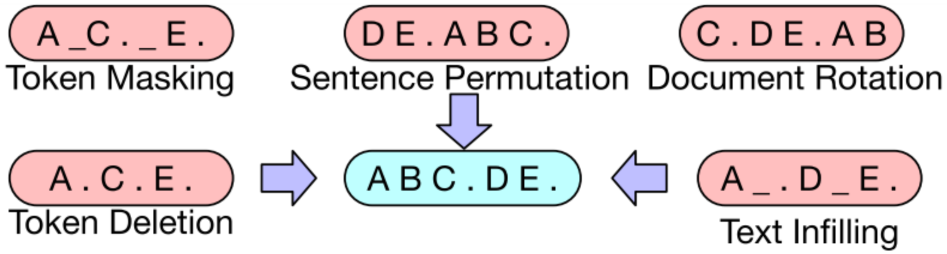




Neural language models

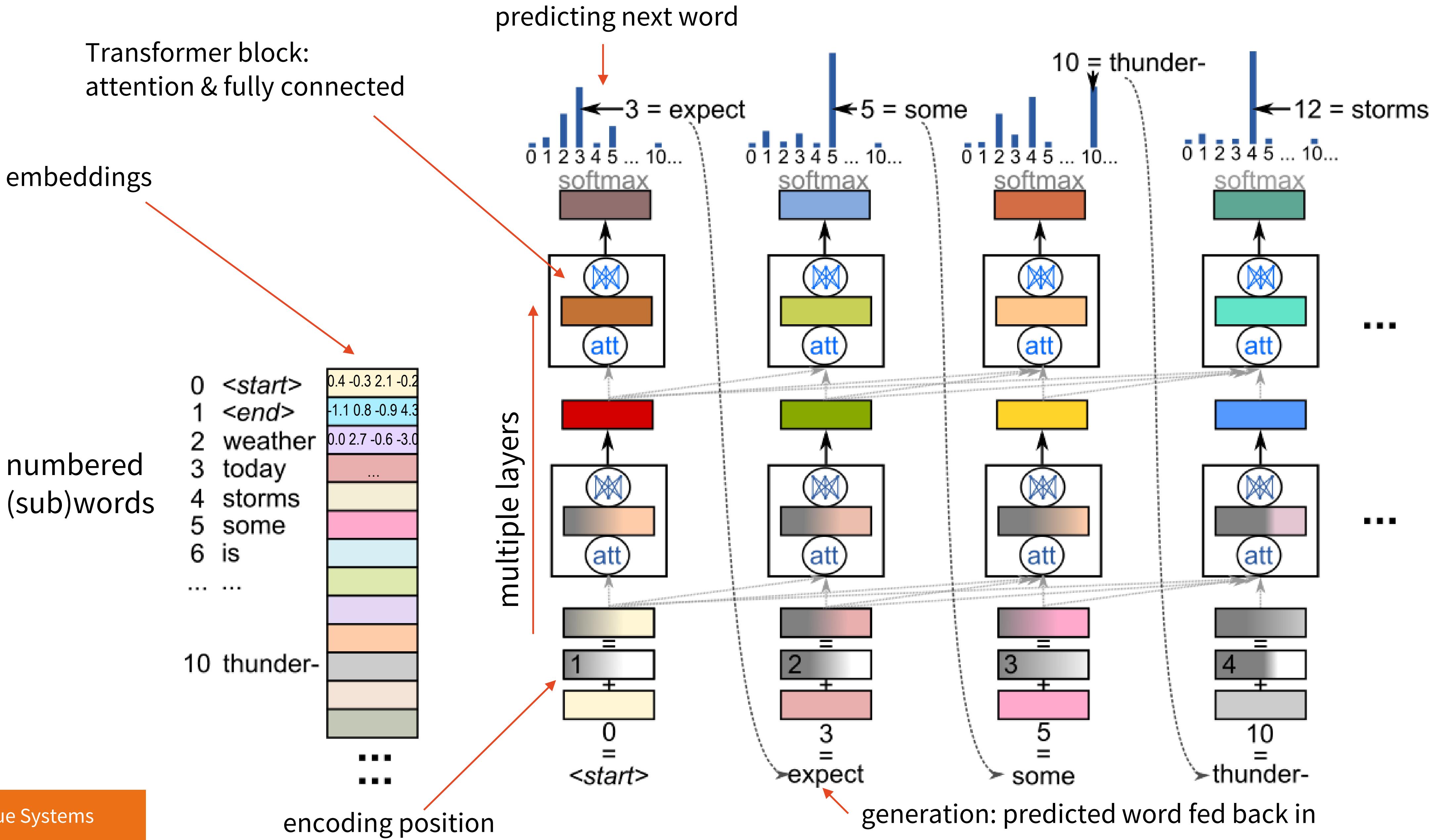
- **Transformer** neural architecture
  - (sub)word representation: **embedding** = vector of numbers
  - **blocks**: **attention** (combining context) + **fully-connected** (abstracting)
  - **predicting next (sub)word** = classification: choosing 1 out of ca. 50k (low level!)
  - trained from data: initialize randomly & iteratively improve
- **Pretrained models**
  - Transformers trained on vast amounts of data
  - Self-supervised training: just naturally occurring text & simple tasks
    - predicting next word
    - predicting masked word
    - fixing corrupt sentences

    …
  - Lot of them released online, plug-and-play



https://github.com/huggingface/transformers

Transformer neural language model



End-to-end Neural Dialogue with GPT-2

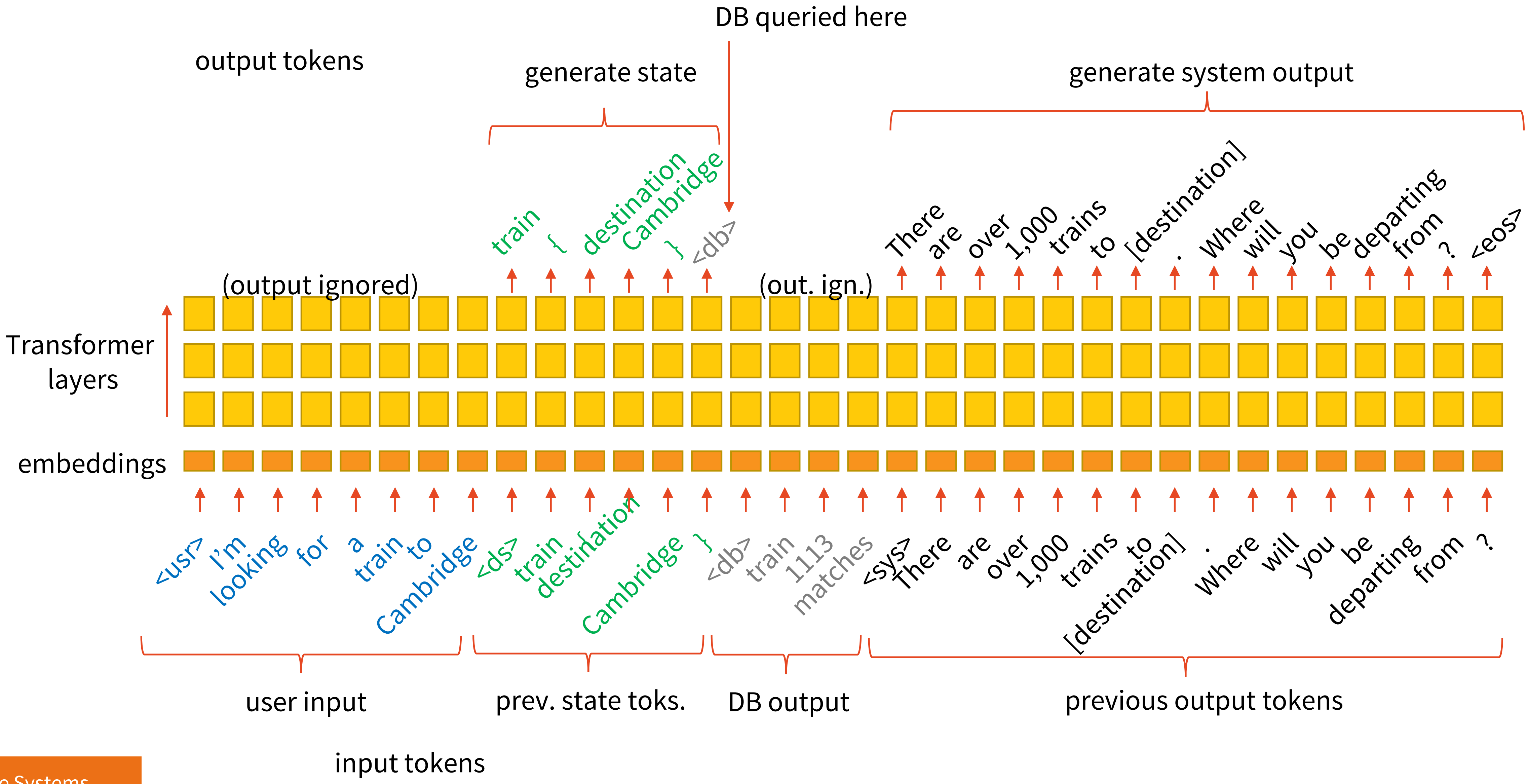
- **GPT-2**: one of most popular pretrained language models
  - Transformer (100M-1.5B params)
  - pretrained on next-word prediction
  - 8M docs, 40GB data from the web

(Radford et al., 2019)
https://openai.com/blog/better-language-models/
https://huggingface.co/gpt2

- **Dialogue**: GPT-2 finetuned (=further trained) on dialogue data (MultiWOZ)
  - task: next word prediction again (low level!)
- Multi-step, all word-by-word:
  1. feed in dialogue context (← ignore generation outputs for this bit)
  2. generate dialogue state (as text)
  3. query DB
  4. feed in DB results as text (← ignore outputs)
  5. generate response

(Kulhánek et al., 2021)
http://arxiv.org/abs/2102.05126
https://github.com/ufal/augpt

End-to-end Neural Dialogue with GPT-2



Problems & solutions

- **Needs a lot of data** & annotation (1000s of dialogues)
  - costly, may be noisy
  - → transfer learning, data augmentation
- **Hallucinates** sometimes
  - may generate factually incorrect outputs, hard to control
  - → data cleaning, consistency training (corrupt data & train to detect this)
- **Repetitive**/dull outputs
  - settles for the most frequent output
  - → sampling (randomness)
- Still a long way to go
  - ~70% correct/successful dialogues
  - still needs a lot of data

Large language models (LLMs): Pretrain & prompt

- Larger than GPT2 size: 10-100B parameters
  - hard to run in-house (OPT, BLOOM, LlaMa) or not free (GPT-3, ChatGPT, LaMDa)
  - ~10B can run (not train) on regular GPUs with 8-bit mode https://huggingface.co/blog/hf-bitsandbytes-integration
- architecture mostly the same
  - typically decoder-only
- **prompting:** feed in context / few examples / ask question, get reply
  - typically no need to finetune
  - finetuning can potentially help,
    but it's expensive & has less effect than in smaller LMs
- better training (ChatGPT): reinforcement learning with **human feedback**
  - overall human ratings of text – not just next word
- same problems as finetuning – hard to control

ChatGPT

- Impressive text editing & composition skills
- Still hallucinates a lot
  - only uses information it memorized

Who is Ondřej Dušek?

14, depending how you count

Ondřej Dušek is a researcher and academic in the field of natural language processing (NLP). He is currently an Assistant Professor at the Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic.

Dušek's research focuses on developing computational models and algorithms for natural language generation, dialogue systems, and machine learning approaches for NLP tasks. He has published several research papers in top-tier NLP conferences and journals, and he has also served as a reviewer and program committee member for these conferences.

no other tasks than generation & dialogue

In addition to his research, Dušek is also actively involved in teaching and mentoring students in the field of NLP. He has supervised several Master's and PhD theses, and he teaches courses on topics such as computational linguistics, NLP, and machine learning.

no PhD graduate yet

I only teach dialogue systems

Compose a limerick on how to exit Vim.

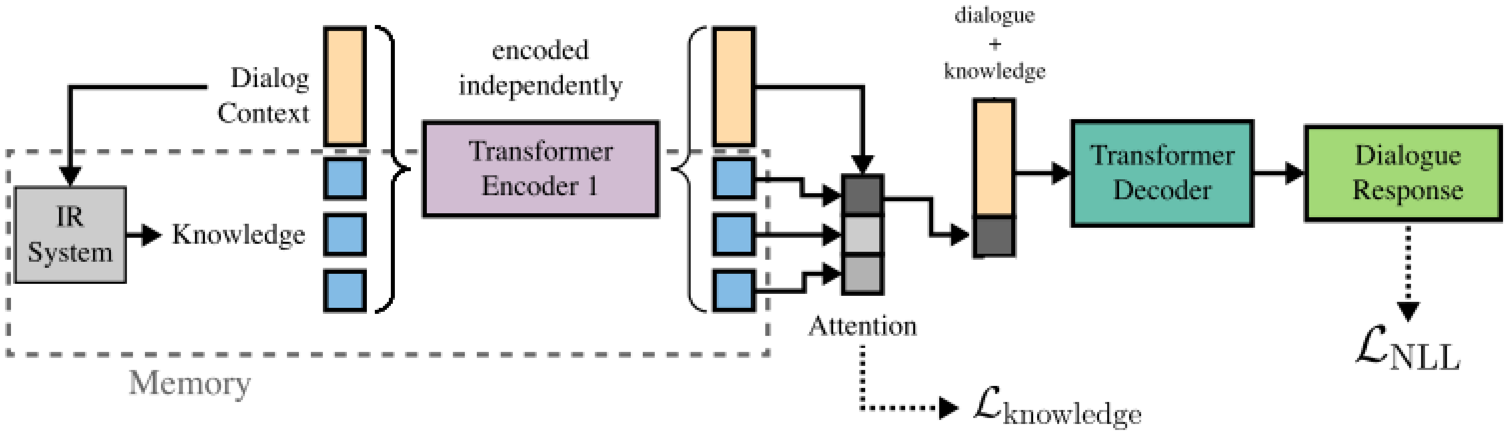
There once was a programmer in Vim
Whose exit strategy was looking grim
They tried to escape
But couldn't find the right shape
Until they remembered to type ":q!" and win!

Nice, now try it while speaking like a pirate

Arrr, listen ye mateys to me tale,
Of a sailor who be stuck in Vim's jail,
He couldn't leave the text editor,
No matter how he did endeavour,
But then he remembered ":q!" and set sail!

Retrieval-augmented Generation

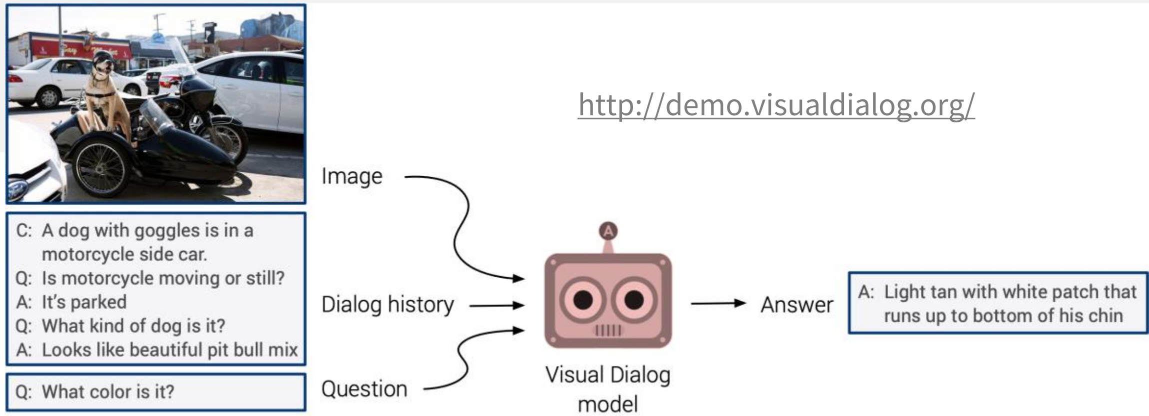
- Search = grounding in facts
  - e.g. using Wikipedia
- 2-step approach:
  1) **Retrieve** a candidate
     - search, relevant to input
  2) **Edit** it to match context
     - generate, condition on candidate
- Models trained to (partially) copy from facts
  - explicitly: classify – copy vs. generate (old)
  - implicitly: shape of data (new)
- Tradeoff: right amount of copying
  - Don’t ignore the retrieved
  - Don’t copy it verbatim



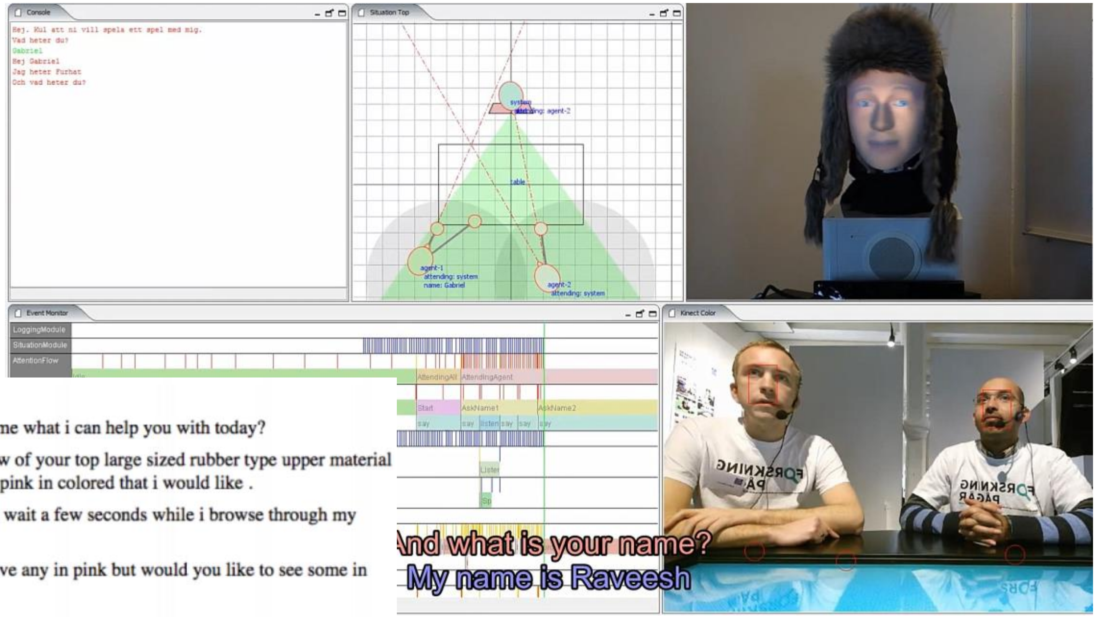
(Pandey et al., 2018) https://aclanthology.org/P18-1123/
(Weston et al., 2018) https://aclanthology.org/W18-5713/
(Dinan et al., 2019) https://arxiv.org/abs/1811.01241
(Xu et al., 2021) http://arxiv.org/abs/2107.07567
(Roller et al., 2021) https://aclanthology.org/2021.eacl-main.24

Multimodal/Visual Dialogue

- adding other modalities
- specific components
  - parallel to NLU
    - vision – image classification networks
    - face identification/tracking
  - parallel to NLG
    - mimics/gesture generation
    - gaze
    - image retrieval
  - vision – typically CNN
    - often off-the-shelf stuff
  - specific classifiers/rules



http://demo.visualdialog.org/



https://youtu.be/5fhjuGu3d0I?t=137

https://vimeo.com/248025147

(Agarwal et al., 2018)
http://aclweb.org/anthology/W18-6514

Further Research Areas

- Multi/open domains
  - reusability, domain transfer
  - training from little data
  - pretraining with “generic” data
  - connecting task-oriented systems and chatbots
- Context dependency
  - understand/reply in context (grounding, speaker alignment)
- Incrementality
  - don’t wait for the whole sentence to start processing
  - not much stuff going on at the moment, but would help
- Evaluation
  - checking if the system does well is actually non-trivial

Summary

- Dialogue is far from solved, but useful systems exist
  - task-oriented vs. non-task-oriented
  - closed vs. open domain
- Standard practice: (**ASR →**) **NLU → DM → NLG** (**→ TTS**) components
  - implementation varies
  - rules/machine learning
  - there are standard toolkits
- Research **End-to-end models** join the components in a single neural net
  - finetuned/prompted language models
  - Transformer architecture
  - experimental, hard to control
- Multimodal systems: adding off-the-shelf I/O components

Thanks

Contact me: odusek@ufal.mff.cuni.cz
http://ufal.cz/ondrej-dusek

Get the slides here:

http://ufal. cz/ondrej-dusek/bibliography (under “Talks”, later today)

References/Inspiration/Further:

- Pierre Lison (Oslo University): https://www.uio.no/studier/emner/matnat/ifi/INF5820/h14/timeplan/index.html
- Oliver Lemon & Verena Rieser (Heriot-Watt University): https://sites.google.com/site/olemon/conversational-agents
- Filip Jurčíček (Charles University): https://ufal.mff.cuni.cz/~jurcicek/NPFL099-SDS-2014LS/
- Milica Gašić (University of Cambridge): http://mi.eng.cam.ac.uk/~mg436/teaching.html
- David DeVault & David Traum (Uni. of Southern California): http://projects.ict.usc.edu/nld/cs599s13/schedule.php
- Luděk Bártek (Masaryk University Brno): https://is.muni.cz/el/1433/jaro2018/PA156/um/
- Gina-Anne Levow (University of Washington): https://courses.washington.edu/ling575/