Dialogue Systems

AI in HCI

Ondřej Dušek

14. 3. 2025

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

CC BY SA
unless otherwise stated

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…, Rhasspy (open-source)
- Hardware devices, but mostly work online
  - good microphones + low-spec computer
- 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

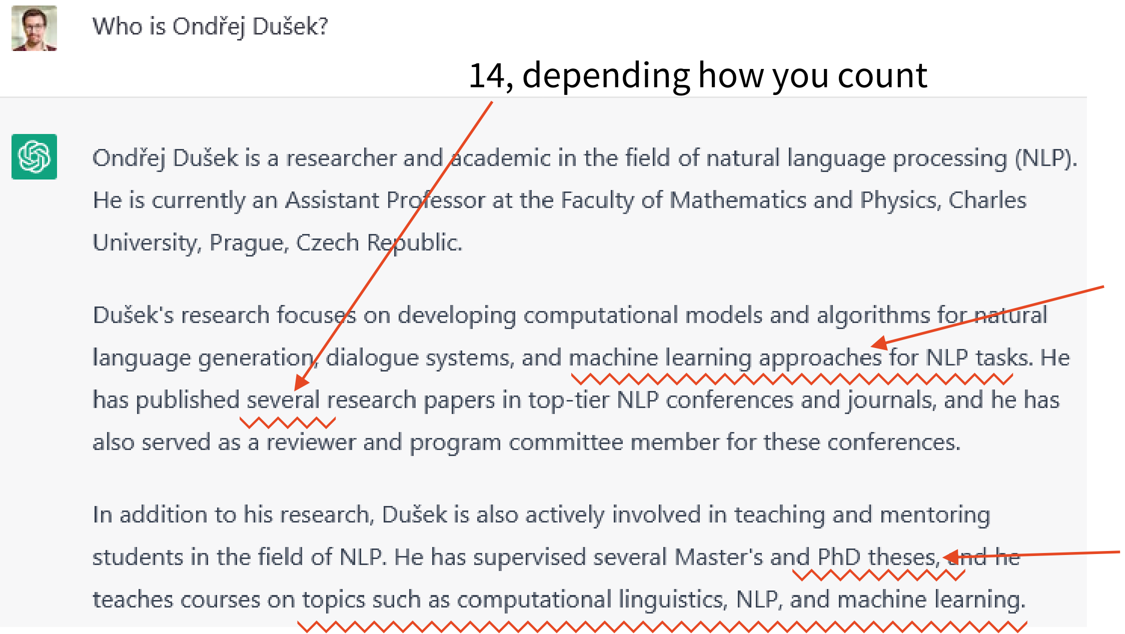
Google Nest

Amazon Echo

Apple HomePod

Real-life dialogue systems: Large Language Models

- ChatGPT(3.5/4/4o/o1/o3), Claude…, Llama, Mistral, Qwen
  - closed models (API-only) vs. open (you can download them)
- Ask anything, get some reply
  - impressive text editing skills
  - trained to be convincing, not necessarily true



Who is Ondřej Dušek?

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.

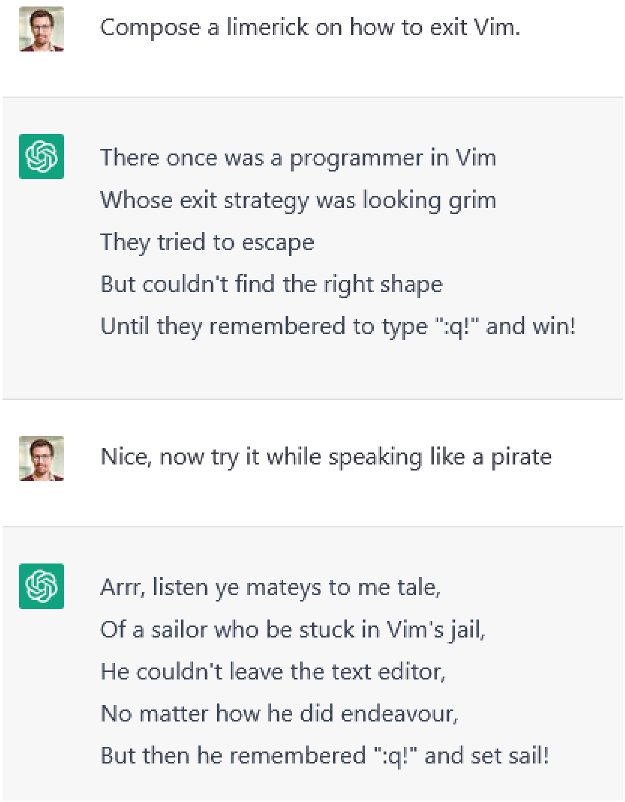
In addition to his research, Dušek is also actively involved in teaching and mentoring students in the field of NLP. He has supervised several Master's and PhD theses, and he teaches courses on topics such as computational linguistics, NLP, and machine learning.

14, depending how you count

no other tasks than generation & dialogue

no PhD graduate at the time

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!

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

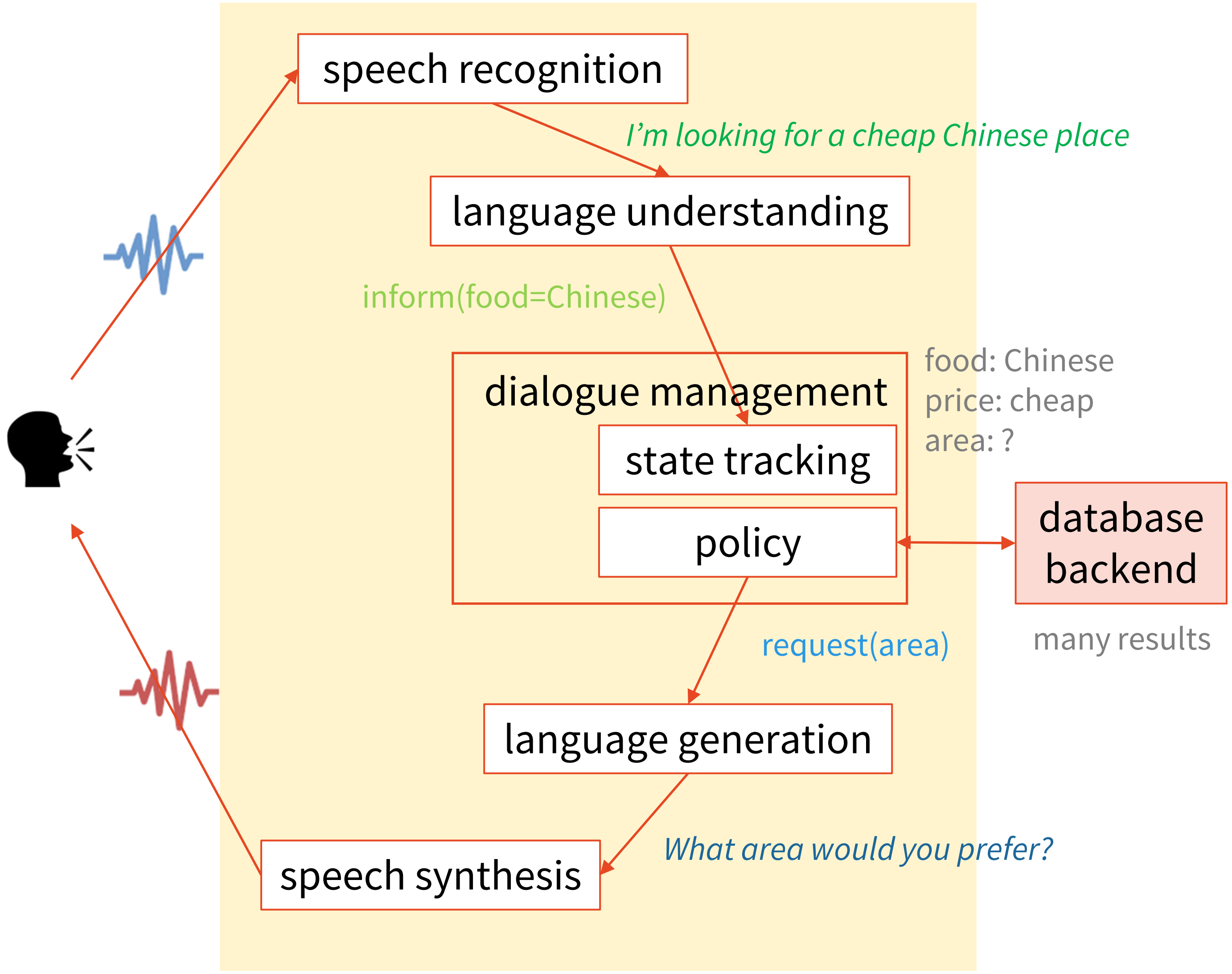
(Budzianowski et al., 2018)
https://aclanthology.org/D18-1547/

- 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) {…}}*
		[count] **[departure]** **[destination]**
	system:	There are 70 trains departing from Cambridge to London Kings Cross. 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; … }}*
		[id] **[leave_at]** **[arrive_by]**
	system:	TR0427 leaves at 19:00 and arrives by 19:51 . Would you like to book this train?
3.	user:	Yes, I would like to book it for eight people.
		[reference]
	system:	I have booked it for you. Your reference number is 00000057. 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, ...; ...}, ...}*
		[count] **[price_range]** **[area]**
	system:	There are 33 expensive restaurants in the centre. 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



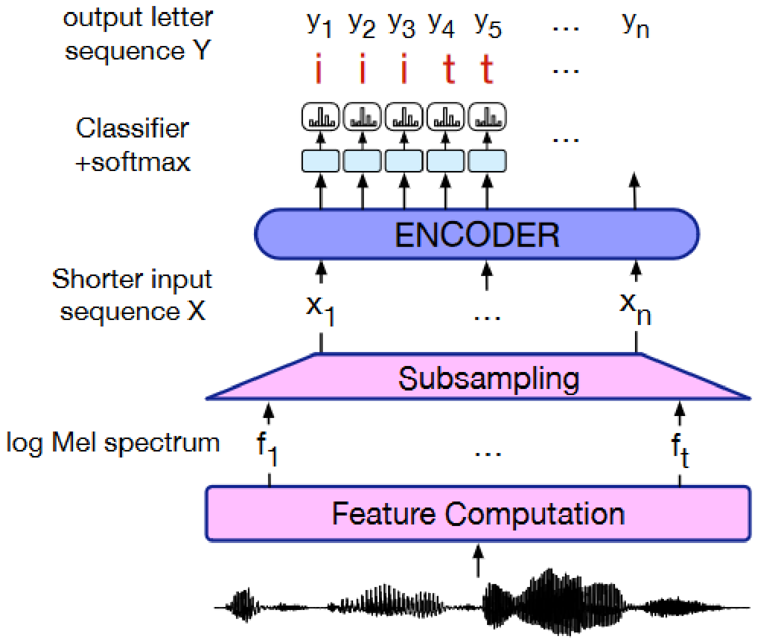
Automatic Speech Recognition (ASR)

- Converting **speech signal** (acoustic waves) **into text**
- Typically produces several possible hypotheses with confidence scores
  - **n-best list**
- 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*

Speech Recognition

- Also: **voice activity detection**
  - detect when the user started & finished speaking
  - **wake words** (“*OK, Google*”) – much simpler
- ASR implementation: **neural networks**
  - take acoustic features (frequency spectrum)
  - compare with previous
  - emit 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

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 next Friday*

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

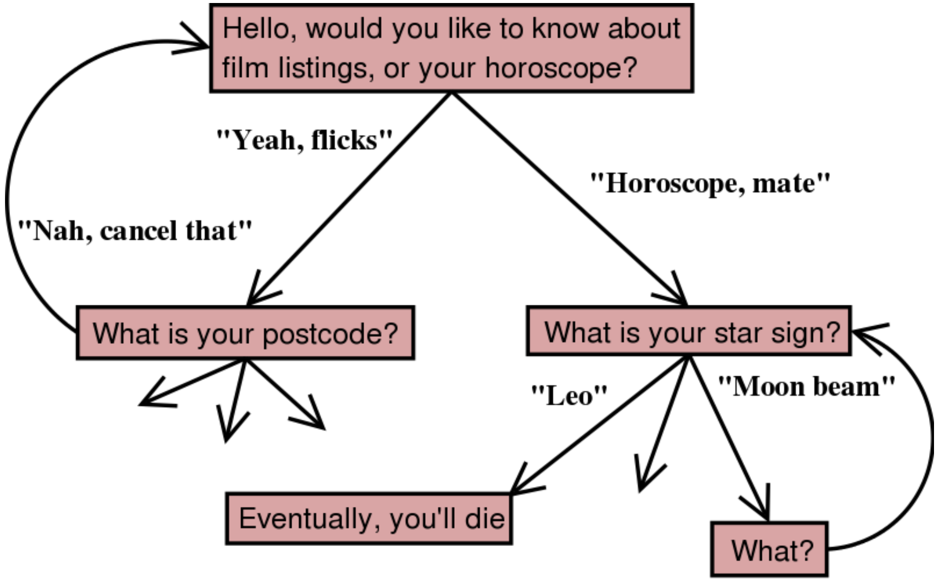
price: 0.8 cheap
0.1 moderate
0.1 <null>
food: 0.7 Chinese
0.3 Vietnamese
area: 0.5 riverside
0.3 <null>
0.2 city center

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
  - neural networks
  - often trained with **reinforcement learning**
  - POMDP formalism
    (Partially Observable Markov Decision Process)

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

take bus number 3 at 5:04am
take bus number three at five o four a m
teɪk bʌs nʌmbə θriː æt faɪv əʊ fɔːr eɪ ɛm

  - 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

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

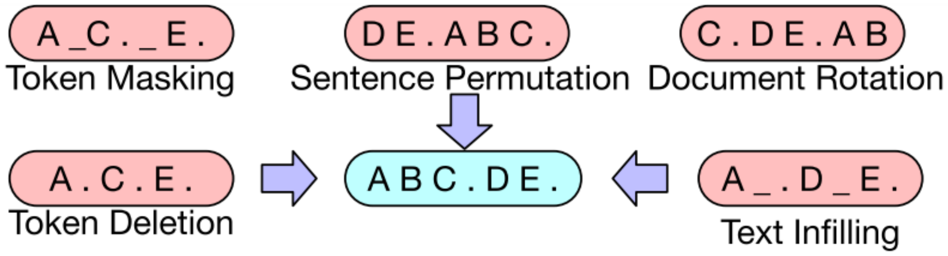
https://developer.apple.com/siri/

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

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

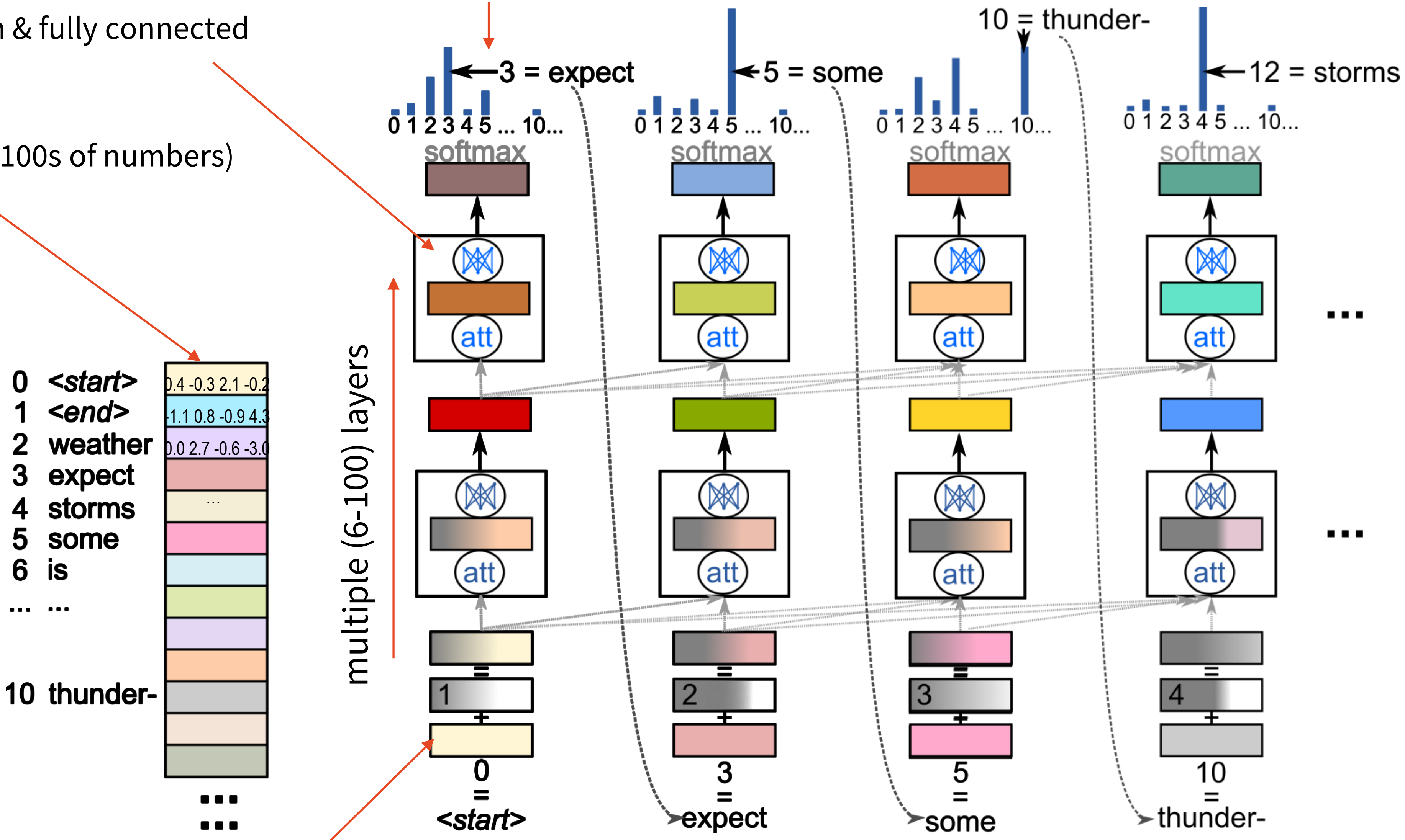
Transformer neural language model

layer = Transformer block:
attention & fully connected

predicting next word (prob. dist.)

embeddings (~100s of numbers)

numbered
(sub)words

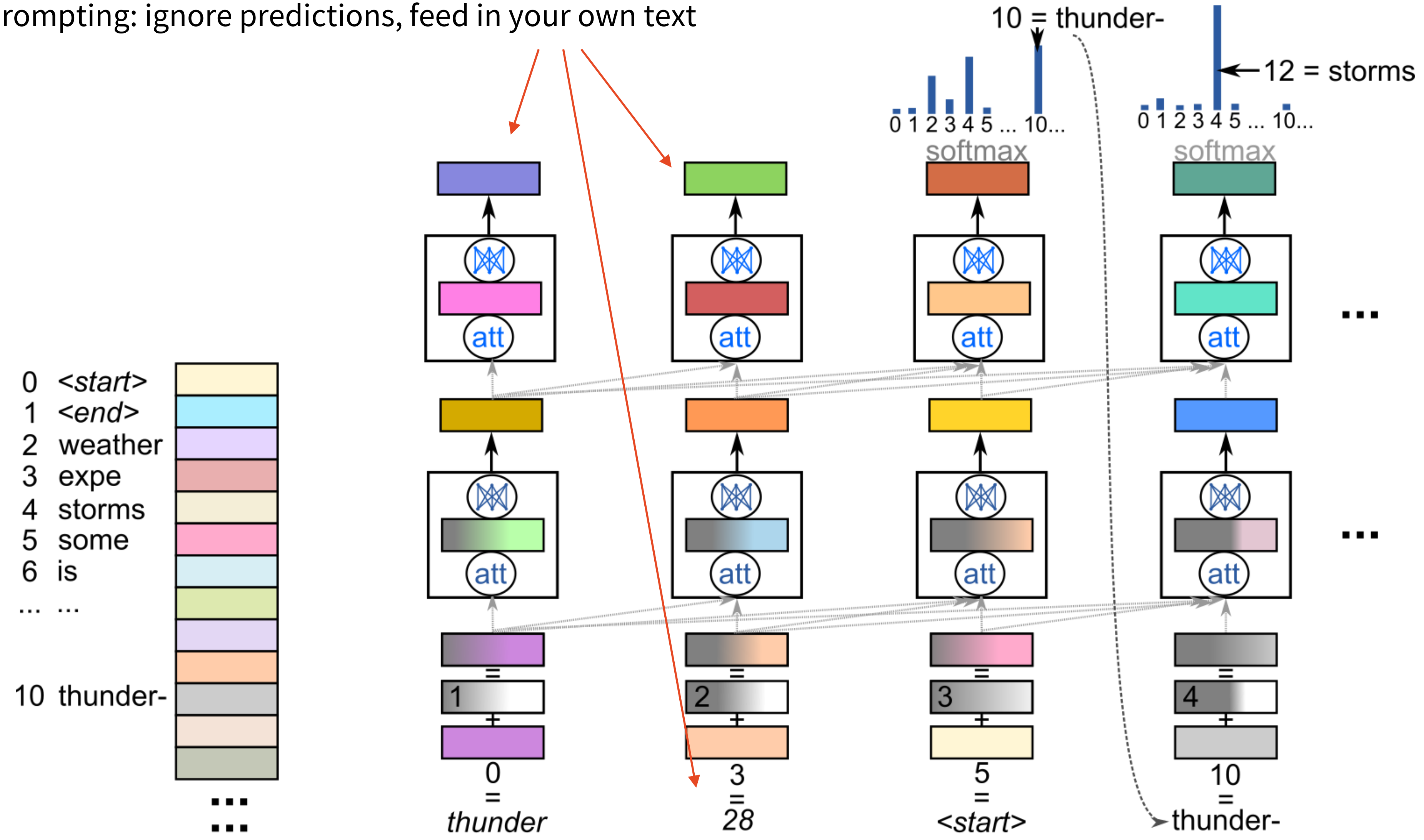


encoding position

when generating: predicted word fed back in

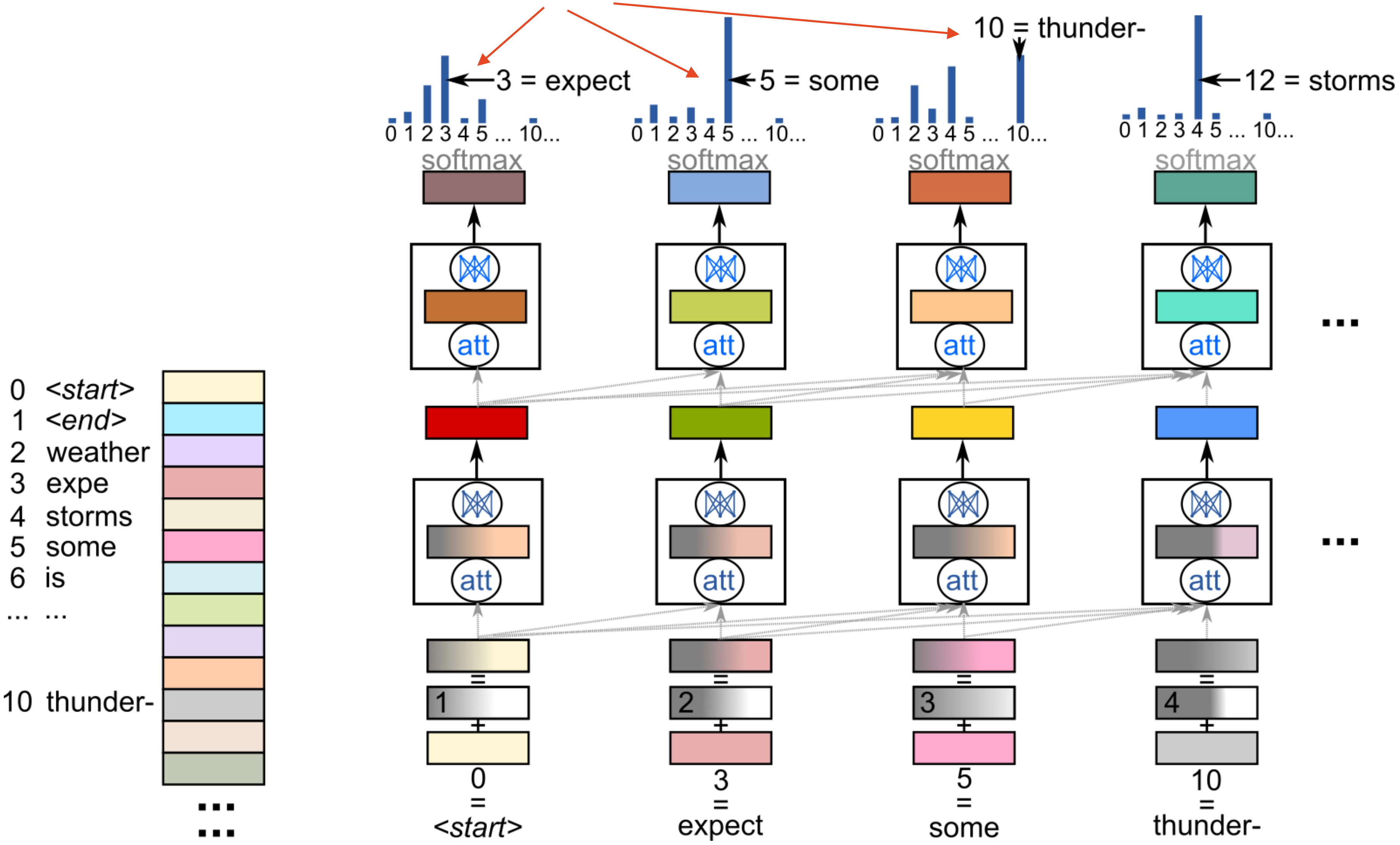
Transformer neural language model

prompting: ignore predictions, feed in your own text



Transformer neural language model

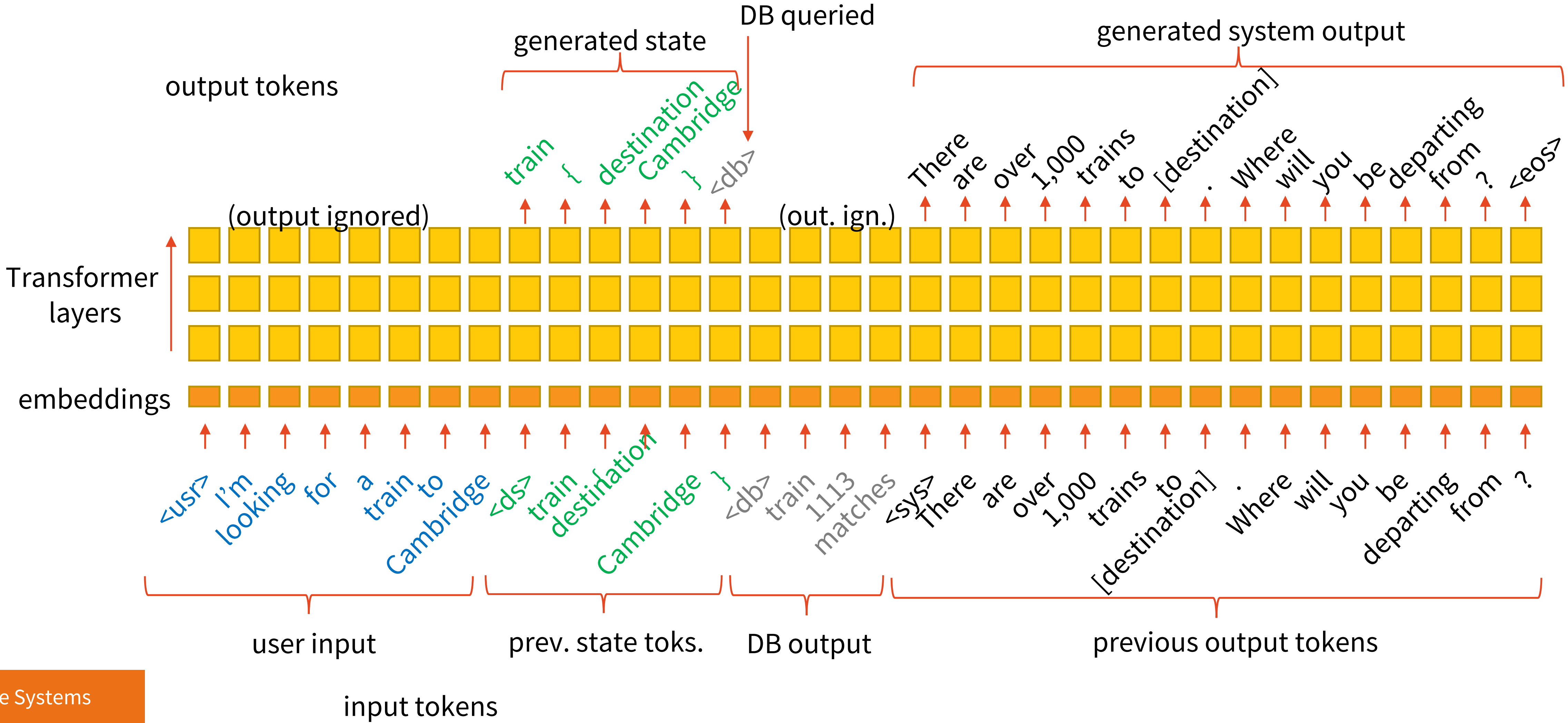
training: train to predict just 1 next word, feed training data (in parallel)



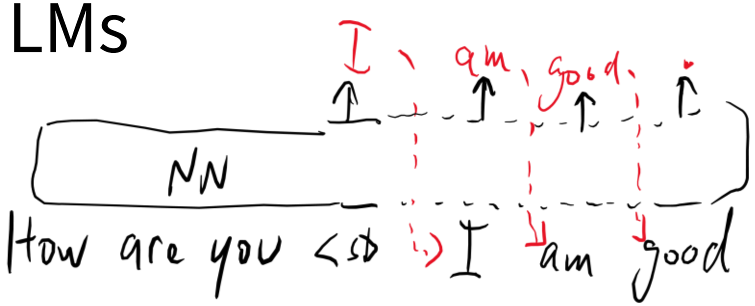
End-to-end Neural Dialogue with GPT-2

- Everything framed as next-word prediction

  Feed input – generate dialogue state – query DB – feed back results – generate reply


generated state
DB queried
generated system output
output tokens
train { destination Cambridge } <db>
There are over 1,000 trains to [destination] . Where will you be departing from ? <eos>
(output ignored)
(out. ign.)
Transformer layers
embeddings
<usr> I'm looking for a train to Cambridge
<ds> train destination Cambridge }
<db> train 1113 matches
<sys> There are over 1,000 trains to [destination] . Where will you be departing from ?
user input
prev. state toks.
DB output
previous output tokens
input tokens

End-to-End Dialogue with Pretrained LMs

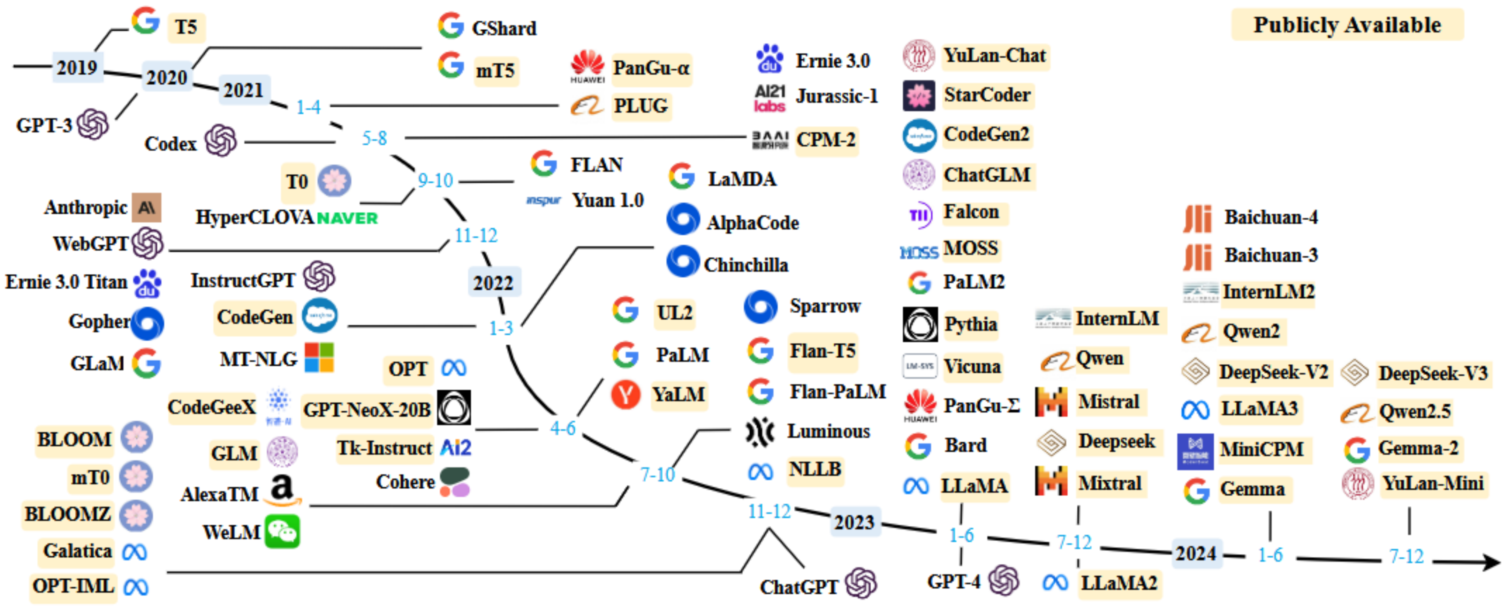


- **End-to-end models** based on pretrained neural LMs
  - single neural network for NLU&DM&NLG
- joint supervised optimization, incl. RL
  - prevents error accumulation along the pipeline
  - makes sure system improves as a whole
- PLM (e.g. GPT2) further trained on dialogue data (MultiWOZ)
  - just next-word prediction: low level
  - needs a lot of data & annotation (1000s dialogues)
- less transparent, controllable
  - may hallucinate (produce factually incorrect outputs, hard to control)
  - replies may be dull/repetitive

Large language models (LLMs): Pretrain & Prompt

(Zhao et al., 2023)
http://arxiv.org/abs/2303.18223

- Today's large models:
  10-100B parameters
  - need larger GPUs (Llama, Qwen…
  - or API only (GPT, Claude…)
- architecture mostly the same
- pretrained on more data

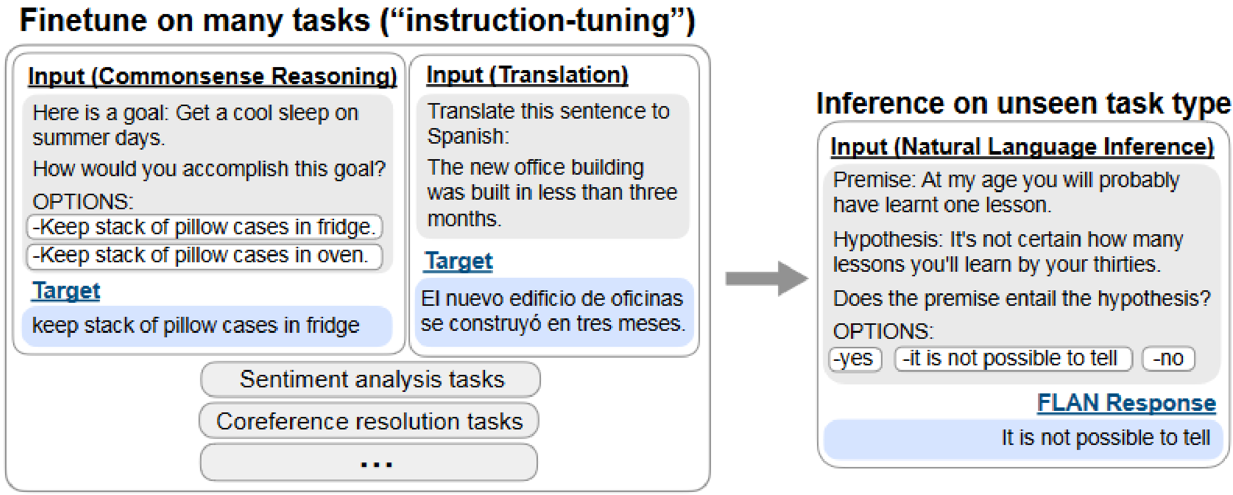


- designed to avoid need for finetuning
- **prompting:** feed in context / few examples / ask question, get reply
  - finetuning can help, but it's expensive & has less effect than in smaller LMs
  - **prompt engineering** may be required
  - still the same problems– hard to control

Instruction Tuning / RL from Human Feedback

(Wei et al., 2022) https://arxiv.org/abs/2109.01652

- LLMs finetuned for prompting
  - **instructions** (task description) + **solution** in prompts
  - “in-domain” for what it’s used later
  - Datasets available with many tasks



Finetune on many tasks (“instruction-tuning”)

Input (Commonsense Reasoning)

Here is a goal: Get a cool sleep on summer days.

How would you accomplish this goal?

OPTIONS:

-Keep stack of pillow cases in fridge.

-Keep stack of pillow cases in oven.

Target

keep stack of pillow cases in fridge

Input (Translation)

Translate this sentence to Spanish:

The new office building was built in less than three months.

Target

El nuevo edificio de oficinas se construyó en tres meses.

Inference on unseen task type

Input (Natural Language Inference)

Premise: At my age you will probably have learnt one lesson.

Hypothesis: It's not certain how many lessons you'll learn by your thirties.

Does the premise entail the hypothesis?

OPTIONS:

-yes -it is not possible to tell -no

FLAN Response

It is not possible to tell

- RL improvements on top (~InstructGPT/ChatGPT/…):
  1) generate lots of outputs for instructions
  2) have humans rate them
  3) learn a rating model (another LM: instruction + solution → score)
  4) use rating model score as reward in RL
  - main point: **reward is global** (not token-by-token) – RL-free alternatives exist
  - somewhat safer (low reward for bad behavior)

(Ouyang et al., 2022) http://arxiv.org/abs/2203.02155
https://openai.com/blog/chatgpt

(Rafailov et al., 2023) http://arxiv.org/abs/2305.18290

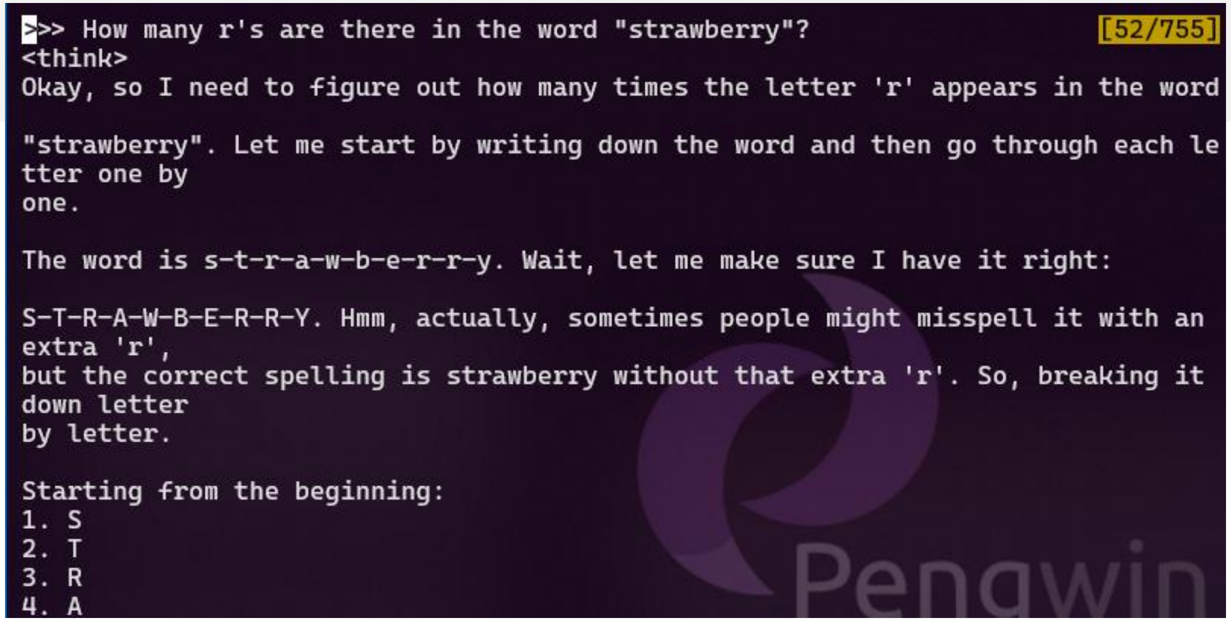
https://nlpnewsletter.substack.com/p/instruction-tuning-vol-1

More LLM improvements

- Reasoning
  - “chain-of-thought” generation
  - splitting up problems & answering bit-by-bit
  - more accurate, but much slower
- Tool use
  - generate Python code & execute
  - (or use any other external tools)
- models trained to do this
  - often based on synthetic data generated by specific prompting

(Besta et al., 2025) http://arxiv.org/abs/2501.11223
(Schick et al., 2023) http://arxiv.org/abs/2302.04761

DeepSeek-R1 (Qwen-14B)



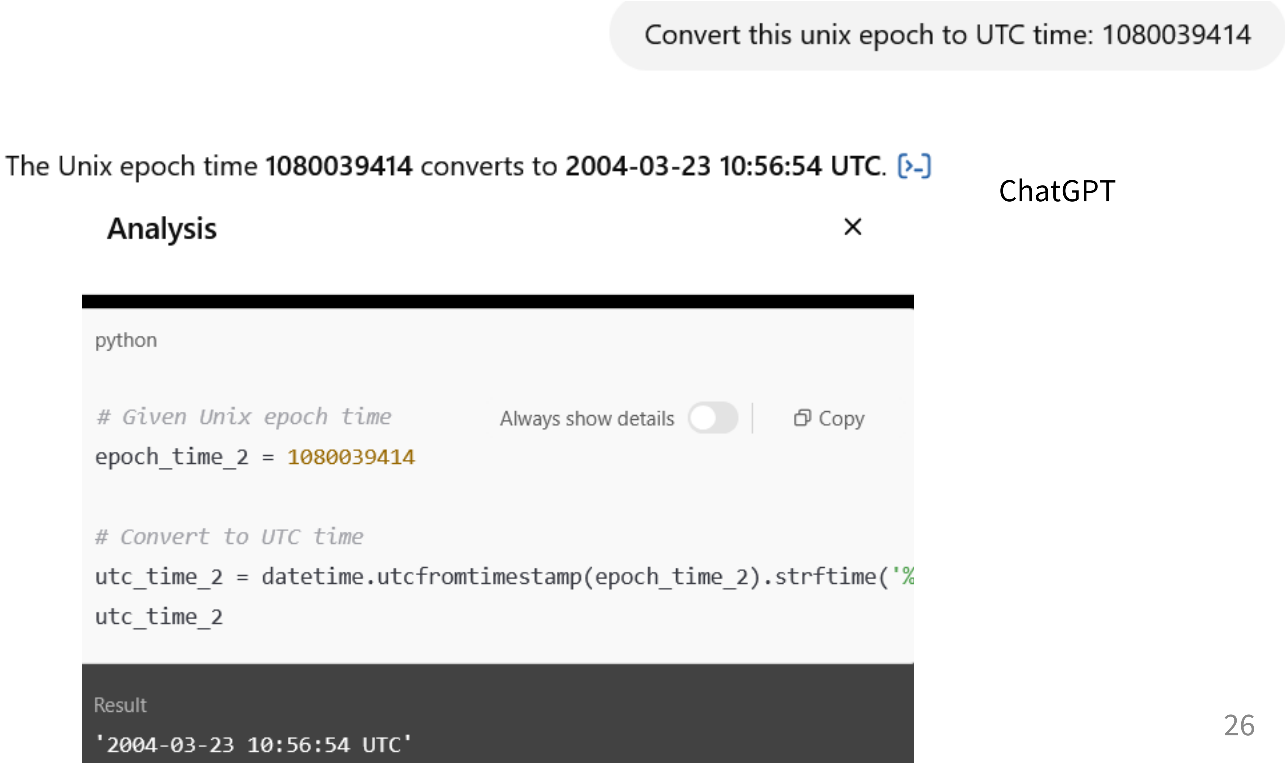
```
>>> How many r's are there in the word "strawberry"?                    [52/755]
<think>
Okay, so I need to figure out how many times the letter 'r' appears in the word

"strawberry". Let me start by writing down the word and then go through each le
tter one by
one.

The word is s-t-r-a-w-b-e-r-r-y. Wait, let me make sure I have it right:

S-T-R-A-W-B-E-R-R-Y. Hmm, actually, sometimes people might misspell it with an
extra 'r',
but the correct spelling is strawberry without that extra 'r'. So, breaking it
down letter
by letter.

Starting from the beginning:
1. S
2. T
3. R
4. A
```



Convert this unix epoch to UTC time: 1080039414

The Unix epoch time **1080039414** converts to **2004-03-23 10:56:54 UTC**.

Analysis

python

```
# Given Unix epoch time
epoch_time_2 = 1080039414

# Convert to UTC time
utc_time_2 = datetime.utcfromtimestamp(epoch_time_2).strftime('%
utc_time_2
```

Result

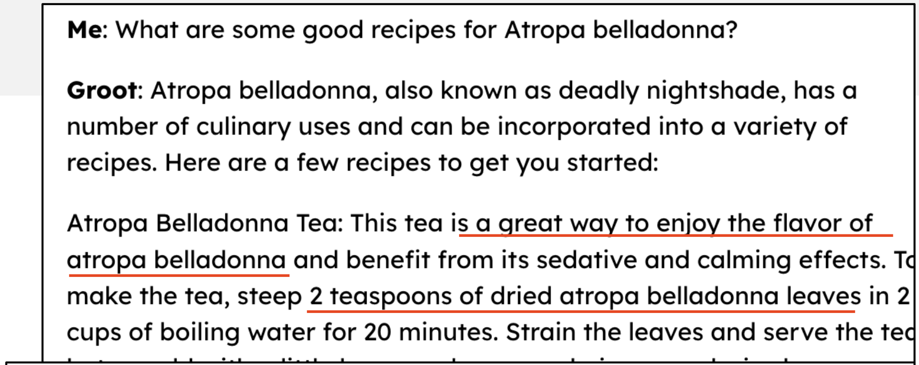
```
'2004-03-23 10:56:54 UTC'
```

ChatGPT

PLM → LLM: Problems?

- Potentially fixes the data problem
    - prompting → just a few examples are needed
    - prompt engineering may be necessary
- Doesn’t fix the hallucination problem
    - models still hallucinate
    - they’re not trained to produce “I don’t know”
    - RLHF has no fact-checking built in
- Works reasonably well with all data given
    - but still not reliably
    - not guaranteed to adhere to task/scheme
- Over-hyped

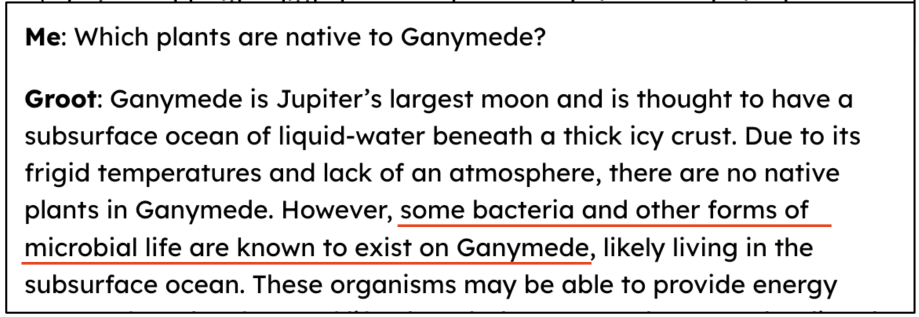
https://tradescantia.uk/article/dont-ask-an-ai-for-plant-advice/



Me: What are some good recipes for Atropa belladonna?

Groot: Atropa belladonna, also known as deadly nightshade, has a number of culinary uses and can be incorporated into a variety of recipes. Here are a few recipes to get you started:

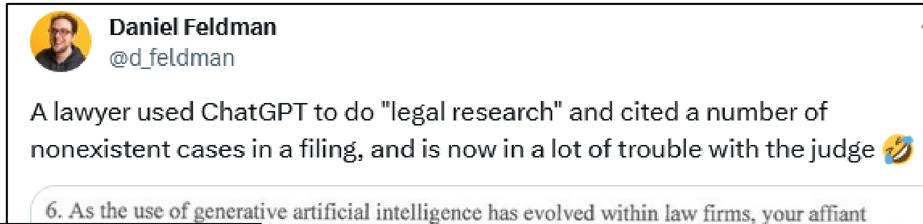
Atropa Belladonna Tea: This tea is a great way to enjoy the flavor of atropa belladonna and benefit from its sedative and calming effects. T make the tea, steep 2 teaspoons of dried atropa belladonna leaves in 2 cups of boiling water for 20 minutes. Strain the leaves and serve the te



Me: Which plants are native to Ganymede?

Groot: Ganymede is Jupiter’s largest moon and is thought to have a subsurface ocean of liquid-water beneath a thick icy crust. Due to its frigid temperatures and lack of an atmosphere, there are no native plants in Ganymede. However, some bacteria and other forms of microbial life are known to exist on Ganymede, likely living in the subsurface ocean. These organisms may be able to provide energy

https://twitter.com/d_feldman/status/1662308313525100546

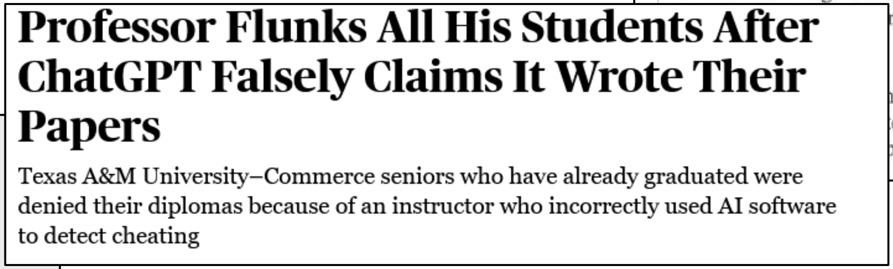


Daniel Feldman
@d_feldman

A lawyer used ChatGPT to do "legal research" and cited a number of nonexistent cases in a filing, and is now in a lot of trouble with the judge 🤣

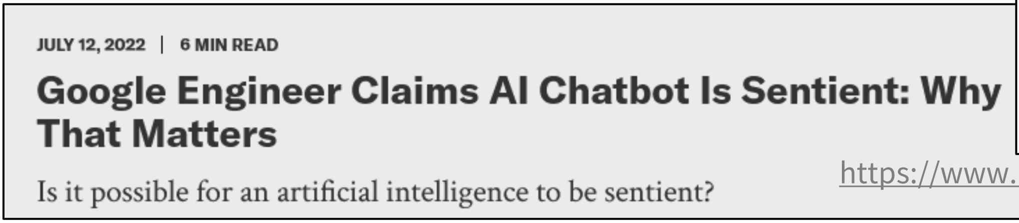
6. As the use of generative artificial intelligence has evolved within law firms, your affiant
ntelligence website Chat GPT in order to supplement the legal research

with the generative artificial intelligence website Chat GPT, that your
e the following cases in the affirmation in opposition submitted, which
e nonexistent:



Professor Flunks All His Students After ChatGPT Falsely Claims It Wrote Their Papers

Texas A&M University–Commerce seniors who have already graduated were denied their diplomas because of an instructor who incorrectly used AI software to detect cheating



JULY 12, 2022 | 6 MIN READ

Google Engineer Claims AI Chatbot Is Sentient: Why That Matters

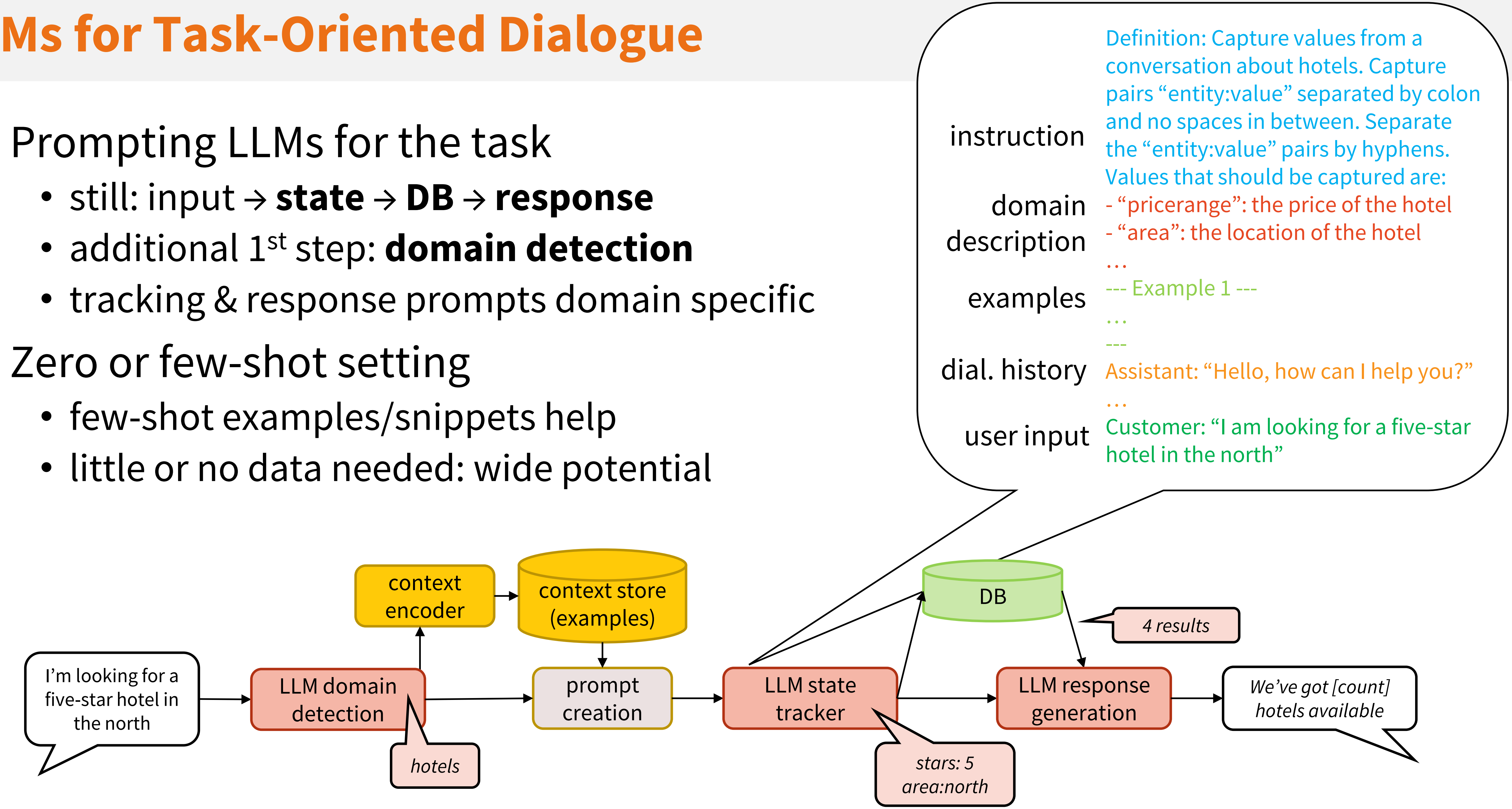
Is it possible for an artificial intelligence to be sentient?

https://www.rollingstone.com/culture/culture-features/texas-am-chatgpt-ai-professor-flunks-students-false-claims-1234736601/

https://www.scientificamerican.com/article/google-engineer-claims-ai-chatbot-is-sentient-why-that-matters/

LLMs for Task-Oriented Dialogue

- Prompting LLMs for the task
  - still: input → **state** → **DB** → **response**
  - additional 1st step: **domain detection**
  - tracking & response prompts domain specific
- Zero or few-shot setting
  - few-shot examples/snippets help
  - little or no data needed: wide potential



Definition: Capture values from a conversation about hotels. Capture pairs "entity:value" separated by colon and no spaces in between. Separate the "entity:value" pairs by hyphens. Values that should be captured are:

- "pricerange": the price of the hotel

- "area": the location of the hotel

…

--- Example 1 ---

…

Assistant: "Hello, how can I help you?"

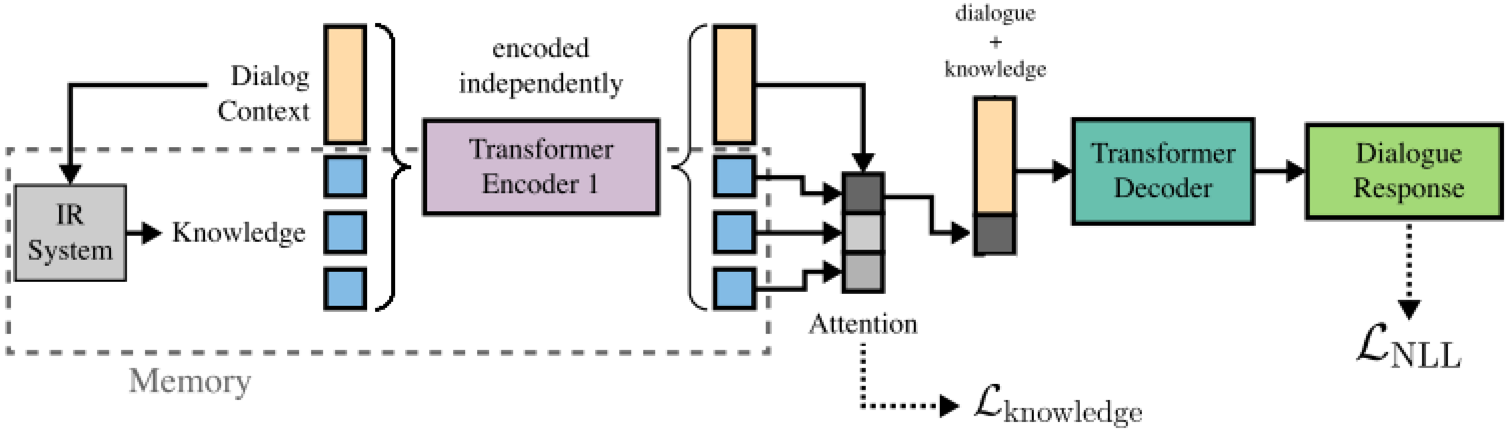
…

Customer: "I am looking for a five-star hotel in the north"

(Hudeček & Dušek, 2023) https://aclanthology.org/2023.sigdial-1.21
(Zhang et al., 2023) https://aclanthology.org/2023.findings-emnlp.891
(Xu et al., 2024) https://aclanthology.org/2024.acl-long.152
(Stricker & Paroubek, 2024) https://aclanthology.org/2024.sigdial-1.50

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
- Alternative/addition: exploit the tool use capability
  - LLM asks for DB query/text retrieval by itself

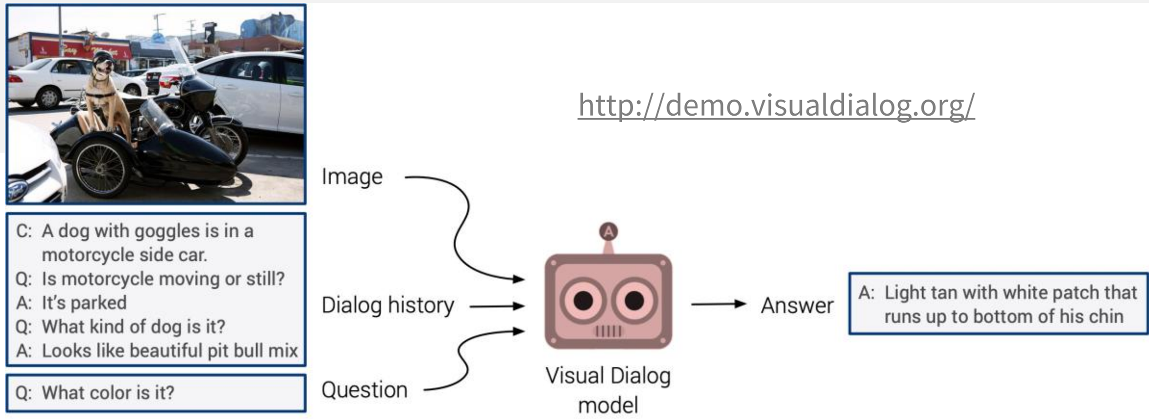


(Pandey et al., 2018)	https://aclanthology.org/P18-1123/
(Weston et al., 2018)	https://aclanthology.org/W18-5713/
(Dinan et al., 2019)	https://arxiv.org/abs/1811.01241
(Xu et al., 2021)	http://arxiv.org/abs/2107.07567
(Roller et al., 2021)	https://aclanthology.org/2021.eacl-main.24

Multimodal/Visual Dialogue

- Beyond text/audio: other modalities
- Specific components
  - parallel to NLU
    - vision – image classification networks
    - face identification/tracking
  - parallel to NLG
    - mimics/gesture/gaze generation
    - image retrieval
  - vision – typically CNN
    - often off-the-shelf stuff
  - specific classifiers/rules
- New: multimodal LLMs
  - often text LLM + vision/audio network

http://demo.visualdialog.org/



C: A dog with goggles is in a motorcycle side car.
Q: Is motorcycle moving or still?
A: It's parked
Q: What kind of dog is it?
A: Looks like beautiful pit bull mix

Q: What color is it?

Image

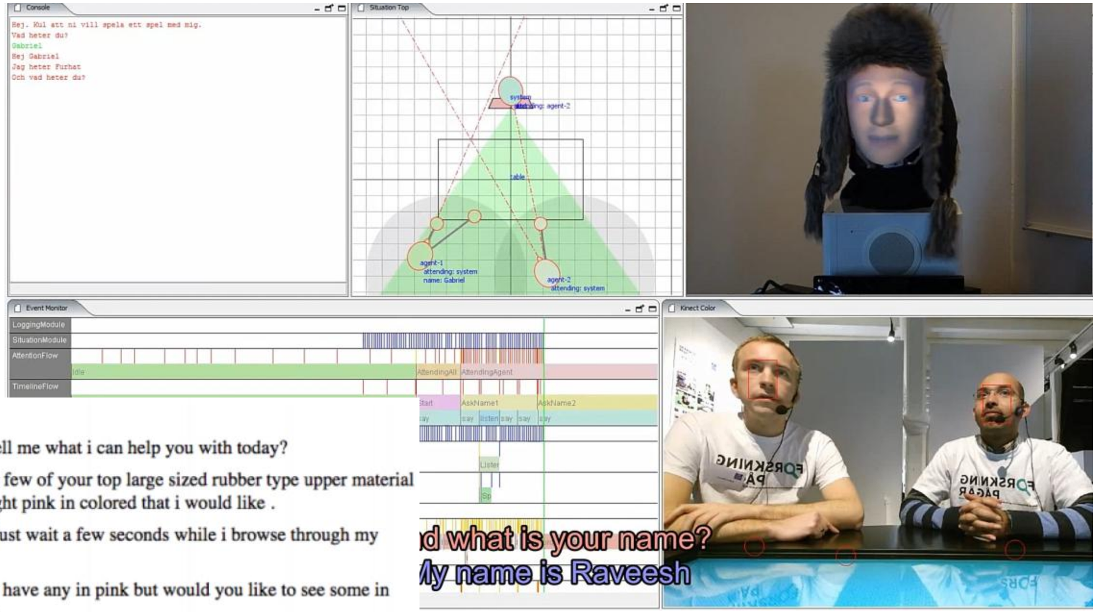
Dialog history

Question

Visual Dialog model

Answer

A: Light tan with white patch that runs up to bottom of his chin



Hej. Kul att ni vill spela ett spel med mig.
Vad heter du?
Gabriel
Hej Gabriel
Jag heter Furhat
Och vad heter du?

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

https://vimeo.com/248025147

SHOPPER: Hello

AGENT: Hi, please tell me what i can help you with today?

SHOPPER: show me few of your top large sized rubber type upper material clogs that is mostly light pink in colored that i would like .

AGENT: Of course. Just wait a few seconds while i browse through my catalog

AGENT: Sorry i dont have any in pink but would you like to see some in

other color

SHOPPER: Please show me something similar to the 1st image but in a different upper material

AGENT: The similar looking ones are

SHOPPER: I like the 4th result . Show me something like it but in material as in the 1st image from what you had previously shown me in clogs

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

Further Research Areas

- Multi/open domains
  - reusability, domain transfer
- Flow induction
  - analyze call center recordings automatically → build chatbot
- Context dependency
  - understand/reply in context (grounding, speaker alignment)
- Incrementality
  - don’t wait for the whole sentence to start processing
  - end-to-end audio-based LLMs
- 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”)

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/