

These slides:

<https://bit.ly/daiz23-od>



Getting Structure in Dialogues with Large Language Models

Ondřej Dušek

Data, AI, Znalosti

9.11.2023

Thanks: Vojtěch Hudeček, Zdeněk Kasner, Ondřej Plátek,
Patrícia Schmidtová, Mateusz Lango, Jonáš Kulhánek, Tomáš Nekvinda, Ioannis Konstas



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

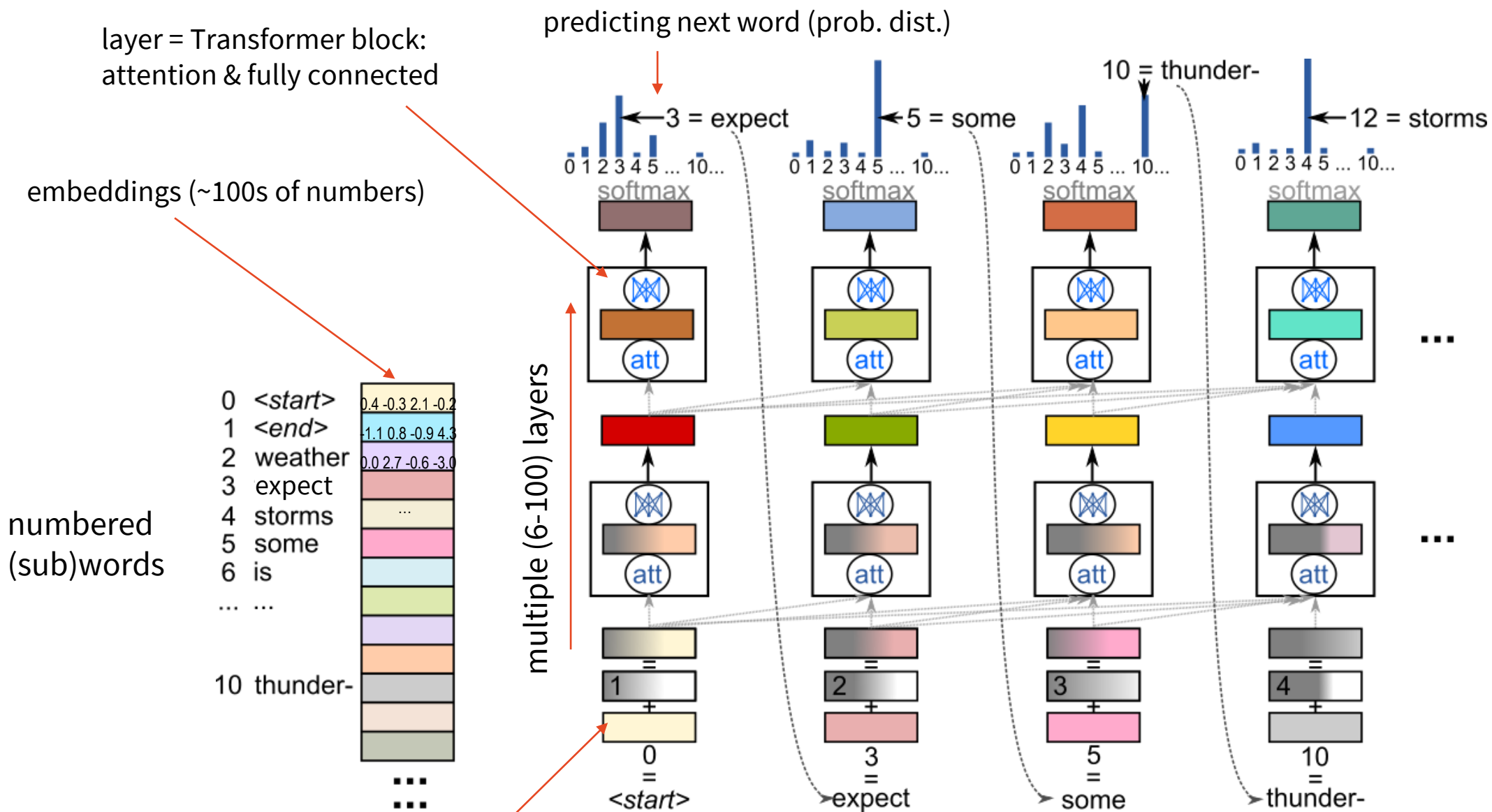


unless otherwise stated

Neural Language Generation

- **End-to-end**
 - feed some input data (linearized), context or prompt
 - neural network handles everything
 - directly generates output text word-by-word, left-to-right
- **Transformer** neural architecture (see→)
- Very **fluent** & convincing outputs
- !Opaque & has **no guarantees on accuracy**
 - used essentially as a black box, internals unknown

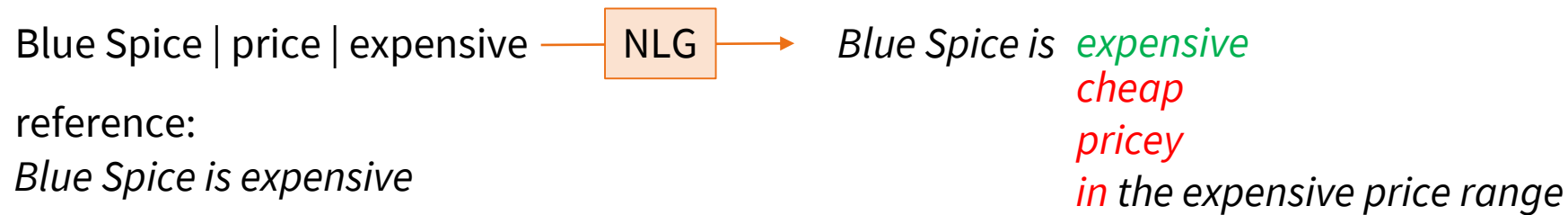
Transformer neural language model



when generating: predicted word fed back in
when training/prompting: ignore predicted, use yours

Training a Neural NLG System

- Reproduce sentences from data
 - replicate exact word at each position
- Fully trained from data
 - initialize model with random parameters
 - input example: didn't hit the right word → update parameters



- Very **low level**, no concept of sentence / text / aim

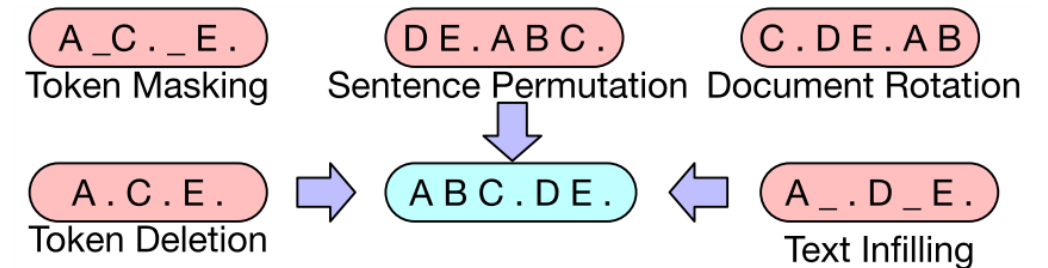
Pretraining & Finetuning

1. **Pretrain** a model on huge data (simple language-based tasks)

- predicting next word
- reconstructing garbled texts

2. **Fine-tune** on your smaller data

- same as training, but starting from a better model
- Models free for download (<https://huggingface.co/>)
 - BERT/RoBERTa, GPT-2, BART, T5...
 - 100k-1B parameters – runs easily on regular GPUs



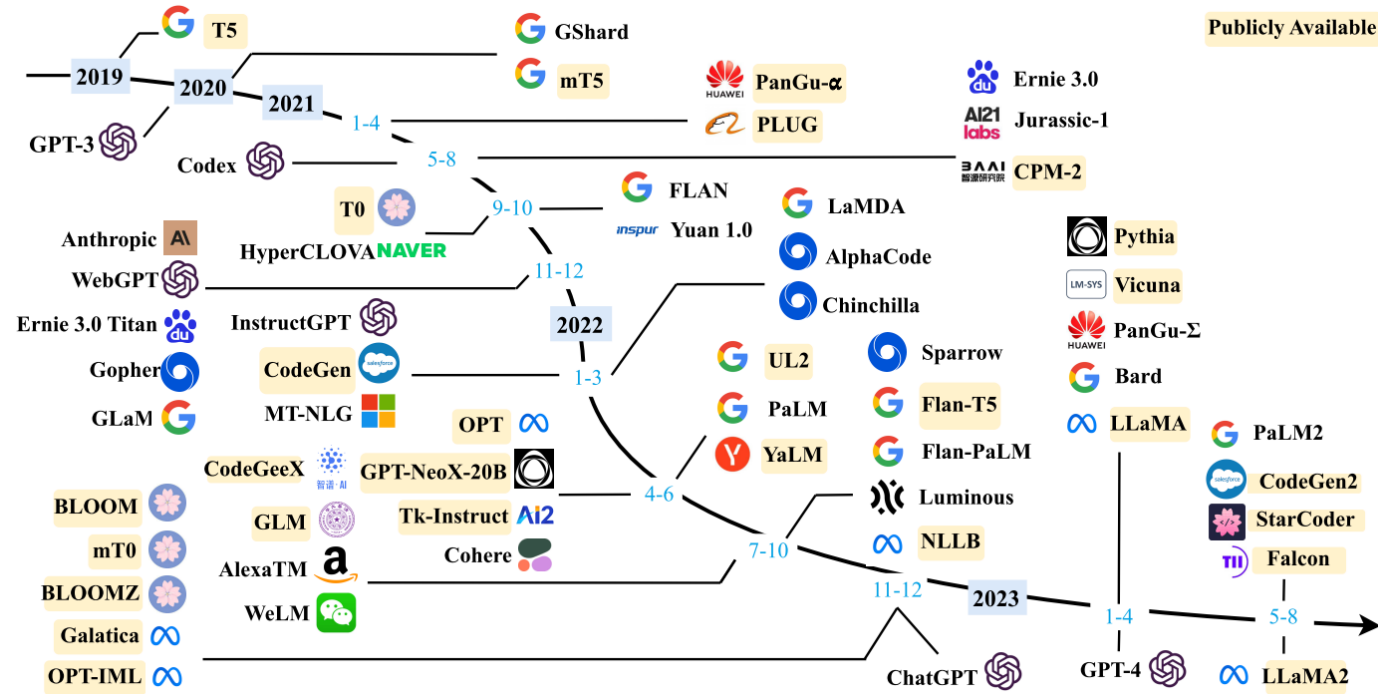
(Lewis et al., 2020)

<https://www.aclweb.org/anthology/2020.acl-main.703>

Large language models (LLMs): Pretrain & Prompt

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

- Today's large models: 10-100B parameters
 - hard(er) to run (OPT, LLaMa, Falcon...)
 - or API only (GPT-3/4, ChatGPT, Bard...)
- architecture mostly the same
- pretrained on more data
- often no need to finetune
- **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.

Sentiment analysis tasks
Coreference resolution tasks
...

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>

LLMs Caveats

- Training scheme ~ Be **convincing** but **not necessarily true**
- !Not reliable for QA: only uses information it memorized, “hallucinates”



Who is Ondřej Dušek?

16, 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.

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 other tasks than generation & dialogue

no PhD graduate yet

I only teach dialogue systems

- Can use information provided in the prompt though (→ →)

- Can we use LMs/LLMs to verbalize simple facts?
 - single subject – relation – object triple (RDF)
 - expressing the relations may be hard

Rel2Text:

- we collected a new dataset to test this
 - current sets were not diverse enough
- 1.5k relations / 4k examples from Wikidata/YAGO/DBPedia
- crowdsourced + manual checks
- It's actually hard for people (our checks removed ~45% data)

relation	possible verbalization
<i>is part of</i>	X is part of Y.
<i>duration</i>	X lasted for Y.
<i>platform</i>	X is available on Y. X runs on Y.
<i>country</i>	X was born in Y. X is located in Y.
<i>parent</i>	X is the parent of Y. Y is the parent of X.
<i>ChEMBL</i>	X has an id Y in the ChEMBL database.

Evaluating LMs on Rel2Text

- Testing on unseen relations only
- **Finetuning** BART (“old-school” PLM)
 - training on Rel2Text works well
 - WebNLG (old, less relations) OK (esp. on correctness)
 - ~hundreds of examples needed
- **Prompting** ChatGPT
 - requires carefully crafted prompts
 - chattier outputs (~less control)
- Error analysis
 - **Unclear relation labels lead to semantic errors**
 - Still some “unprovoked” semantic errors
 - BART + Rel2Text & ChatGPT produce nicer, less literal verbalizations

Rel2Text data	BLEU	% Log. Entail	PPL↓ (GPT2)
Human	-	-	5.88
Copy baseline	29.04	91.21	7.55
BART/WebNLG	41.99	89.39	5.65
BART/Rel2Text	52.54	91.85	5.89
ChatGPT	38.23	88.58	5.68

~overlap with human
~correctness
~fluency

Task-oriented Dialogue

- **Assistant:** fulfill user requests (book a hotel/restaurant/taxi etc.)
- MultiWOZ: benchmark for multiple connected domains
 - 10k dialogues, extensive annotation (but noisy!)

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

- user:** I am looking for a train from Cambridge to London Kings Cross.
state: *{train {departure = cambridge, destination = london kings cross}}*
DB: *{train (70) {...}}*

system: There are 70 trains departing from Cambridge to London Kings Cross. What day would you like to travel?
- 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; ...}}*

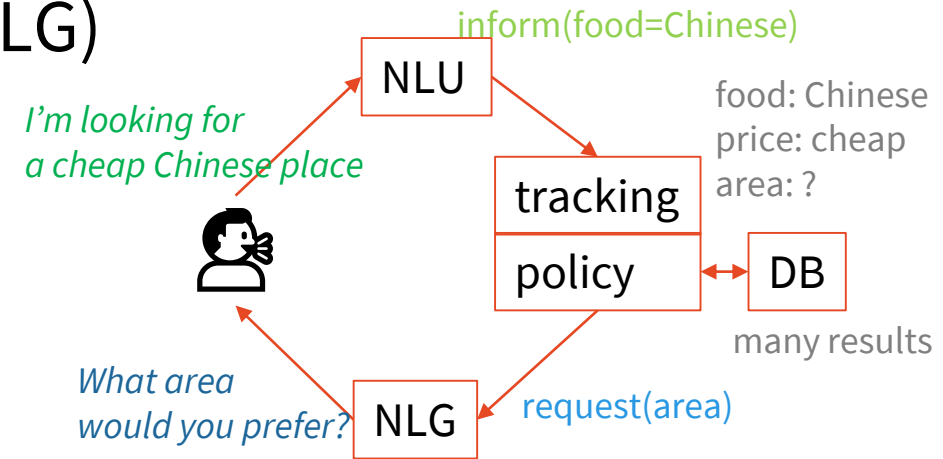
system: TR0427 leaves at 19:00 and arrives by 19:51. Would you like to book this train?
- user:** Yes, I would like to book it for eight people.

system: I have booked it for you. Your reference number is 00000057. Is there anything else I can help you with?
- 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 expensive restaurants in the centre. Is there a particular type of food you would like?

End-to-end Neural Dialogue

- Traditional: separate components (NLU→DM→NLG)
 - trained separately, possibly optimization by RL
- **End-to-end models**
 - single neural LM for NLU&DM&NLG
 - word-by-word response generation

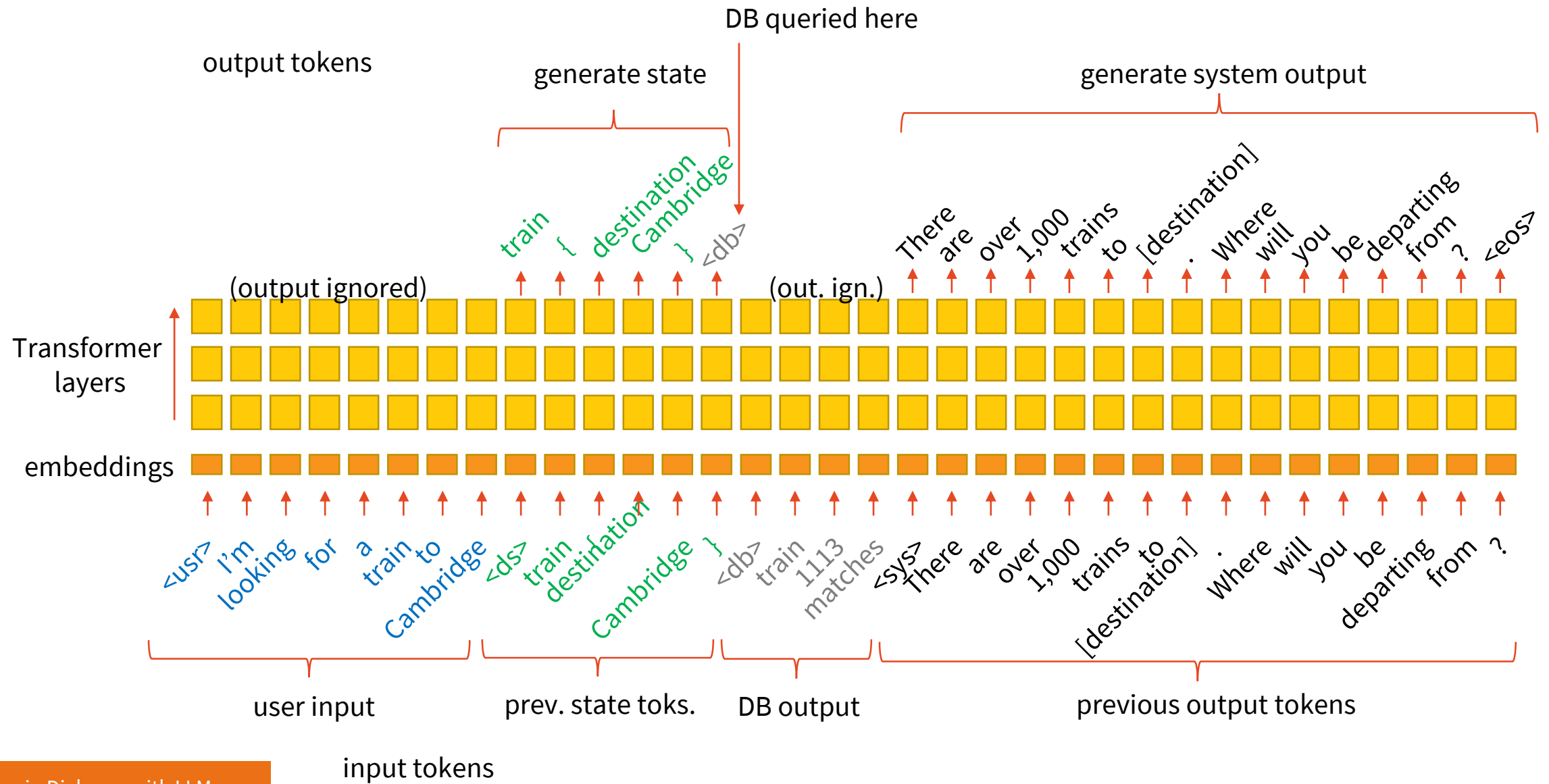


AuGPT: finetuned GPT-2 LM (~100M params)

- Multi-step, all word-by-word:
 1. feed in dialogue context
 2. generate dialogue state (as text)
 3. query DB, feed in DB results as text
 4. generate response

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

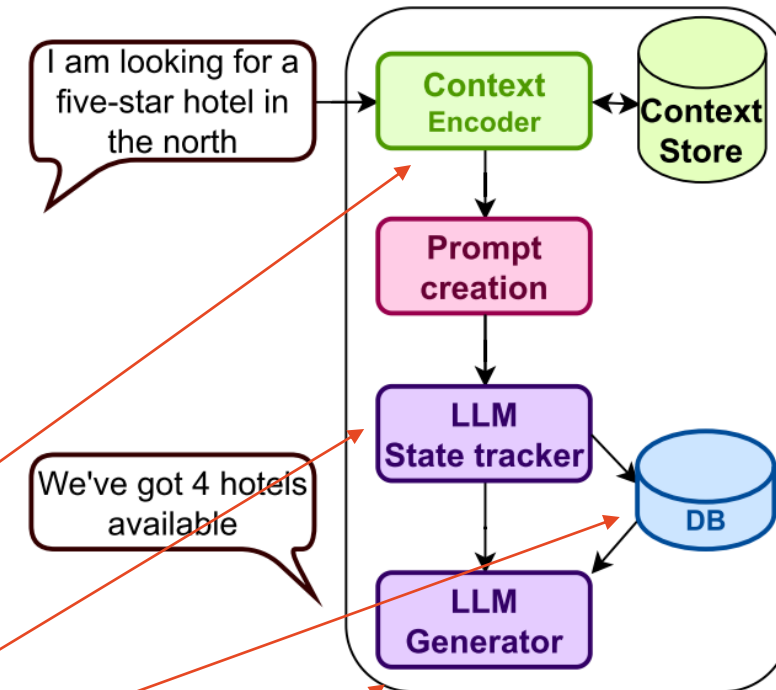
End-to-end Neural Dialogue with GPT-2



Performance

- Dialogue success (=user gets what they wanted)
 - 1-step (corpus-based): 67%
 - crowdsourced human eval: 82% perceived, 62% w/DB
 - expert eval – if you try hard: 87%
- **Hallucinates** sometimes
 - may generate factually incorrect outputs, hard to control
 - → data cleaning, consistency checks
- **Needs a lot of data** & annotation (MultiWOZ = 10k)
 - costly, may be noisy
 - → transfer learning, data augmentation
 - ... or **LLM prompting?**

- How good are LLMs if we require structure?
 - slots / DB are given
 - can we get rid of finetuning / training?
- **Prompting / in-context learning**
- Multiple LMs: ChatGPT, Tk-Instruct, Alpaca...
 - 7-20B params
- Few-shot/zero-shot – little to no “training” data
 - ~ potential for wide usage
- Approach: still the same idea: **context** → **state** → **DB** → **response**



LLM Prompts

- Need to detect active domain
 - additional LLM step
- Examples for prompts (few shot)
 - retrieve similar from context store (domain-specific, FAISS)
 - only a few per domain used

instruction

domain description

examples
(optional)

dial. history

user input

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 ---

...

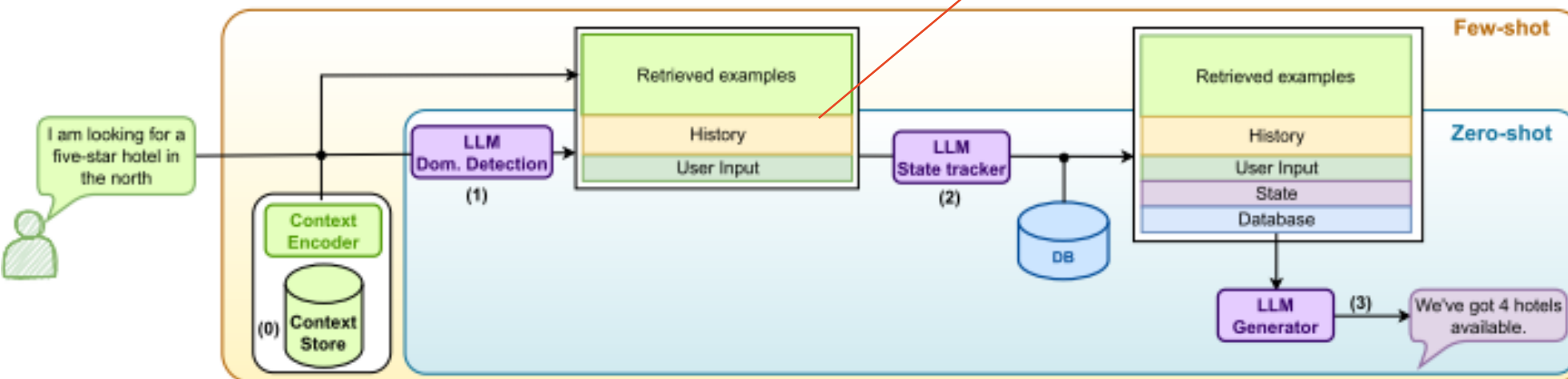
Assistant: “Hello, how can I help you?”

...

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

LLM

stars: 4 – area: north



Results

- Domain detection ~ 70%+
 - Alpaca & TklInstruct OK, ChatGPT almost perfect
- Belief state – pretty bad
 - much worse than SotA
 - examples help (ChatGPT best: ~40-50% F1, others 7-33%), 10 ex./domain enough
 - we may have been too strict, though

- Responses:
OKish

Dialogue Success		ChatGPT	TklInstruct
1-step (corpus)	predicted state	44%	19%
	gold state	68%	46%
expert eval (end-to-end, with recoveries)		76%	64%

- More potential with better prompt engineering

- Evaluating NLG is hard, metrics are inaccurate, humans are expensive
- Can we use LMs to evaluate instead?
- Free chat (non-task-oriented)
- Checking **appropriateness, relevance, diversity** of responses on **1-5** scale

Chat Turns	Appr	Rel	Div
A do you have any pets?	5	-	4
B I am retired so I love to travel so pets would slow me down	4	4	4
A I understand that my idea of traveling is a hot hot bubble bath	3	2	4
B Yes <u>I have dogs and cats</u> I like to take them with me on trips	2	2	4

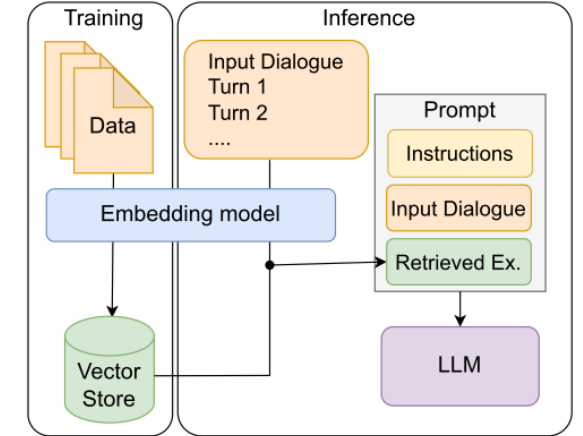
Evaluating Chat

Approach

- Same as previous: LLM prompting
 - few-shot examples in DB
 - LLM asked to provide a score given response in context
- Alternative: LLMs as embeddings & regression on top
 - finetuned on few-shot data
- Checking correlation with humans

Results

- LLM prompting better than prev. SotA (with ChatGPT, Llama2 close, others fail)
- Prompt formulation matters, examples useful
 - but maybe static examples are enough



Appr	Rel	Div	∅
49%	36%	45%	42%

Conclusions

- LLMs are powerful & can work well...
 - if you provide data on the input
 - if you optimize your prompts
- So far, ChatGPT/GPT4 are better than open LLMs
 - new LLMs coming up every week (Llama2, Falcon, Mistral, ...)
 - OpenAI closed models may have seen some of the data (~is it zero-shot?)

Future work

- look into data leakage effect
- more transparency ~ prompting, interpretable latents
- constraining – alignments, decoding-time “critic”

(Lango & Dušek, 2023)

<https://arxiv.org/abs/2310.16964>

Thanks

Contacts:

Ondřej Dušek

odusek@ufal.mff.cuni.cz

<https://tuetschek.github.io>

[@tuetschek](#)

Links

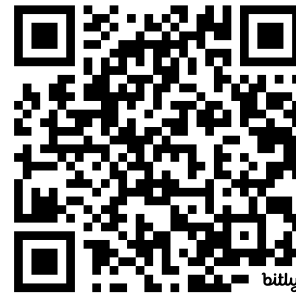
These slides: <https://bit.ly/daiz23-od>

Relations: <https://aclanthology.org/2023.eacl-main.176>

Dialogue: <http://arxiv.org/abs/2102.05126>
<https://aclanthology.org/2023.sigdial-1.21>

Evaluation: <https://aclanthology.org/2023.dstc-1.14>

Thanks:



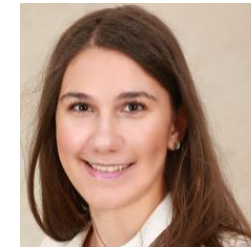
Zdeněk Kasner



Vojtěch Hudeček



Ondřej Plátek



Patrícia Schmidtová



Mateusz Lango



Jonáš Kulháněk



Tomáš Nekvinda



Ioannis Konstas

