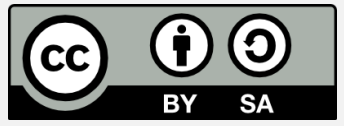
End-to-end Neural Dialogue Systems

Ondřej Dušek

VOCALLS AI Afternoon
19.10.2022

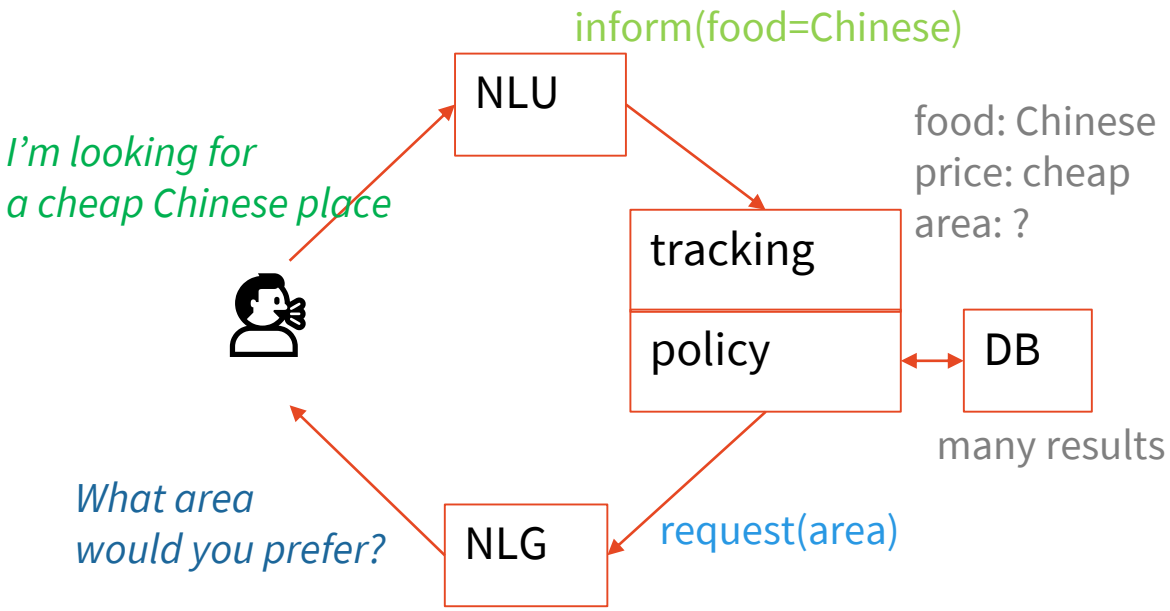
Charles University
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics



Dialogue systems

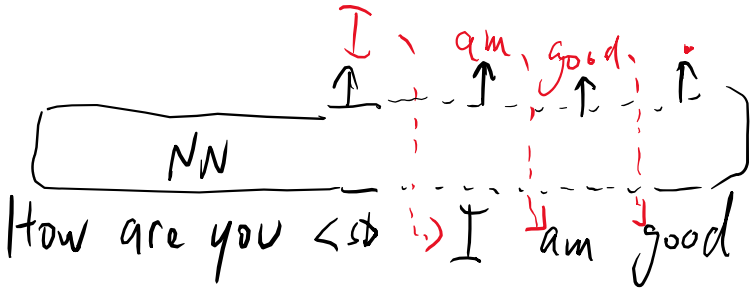
Standard architecture (text-based):

- **Natural language understanding**
  - user utterance → user dialogue act (DA)
- **Dialogue state tracking**
  - update user preferences based on DA
- **Dialogue policy**
  - choose next action based on state
  - system DA
  - consult DB if needed
- **Natural language generation**
  - express system DA as a sentence



End-to-end neural systems

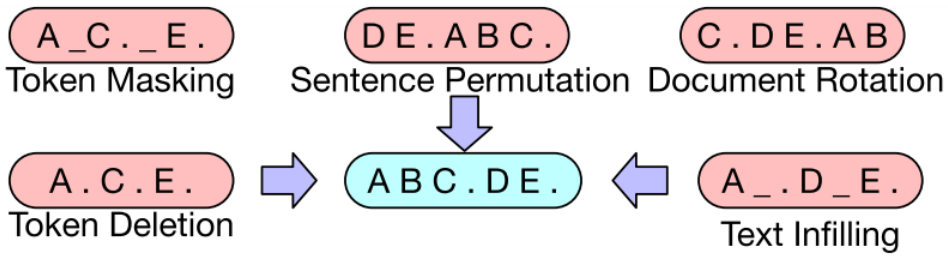
- Standard approach: **Pipeline**
  - each module built separately
  - NLU typically trained (or combined with rules)
  - Tracking could be either
  - Policy, DB, NLG: rules, templates
  - problem: error accumulation, costly to implement
- **End-to-end models**: remove the pipeline
  - NLU/Tracking/Policy/NLG is one neural network
    - generate response word-by-word
  - DB must be external → 2-step operation
  - trained on (many) example dialogues



Neural language models

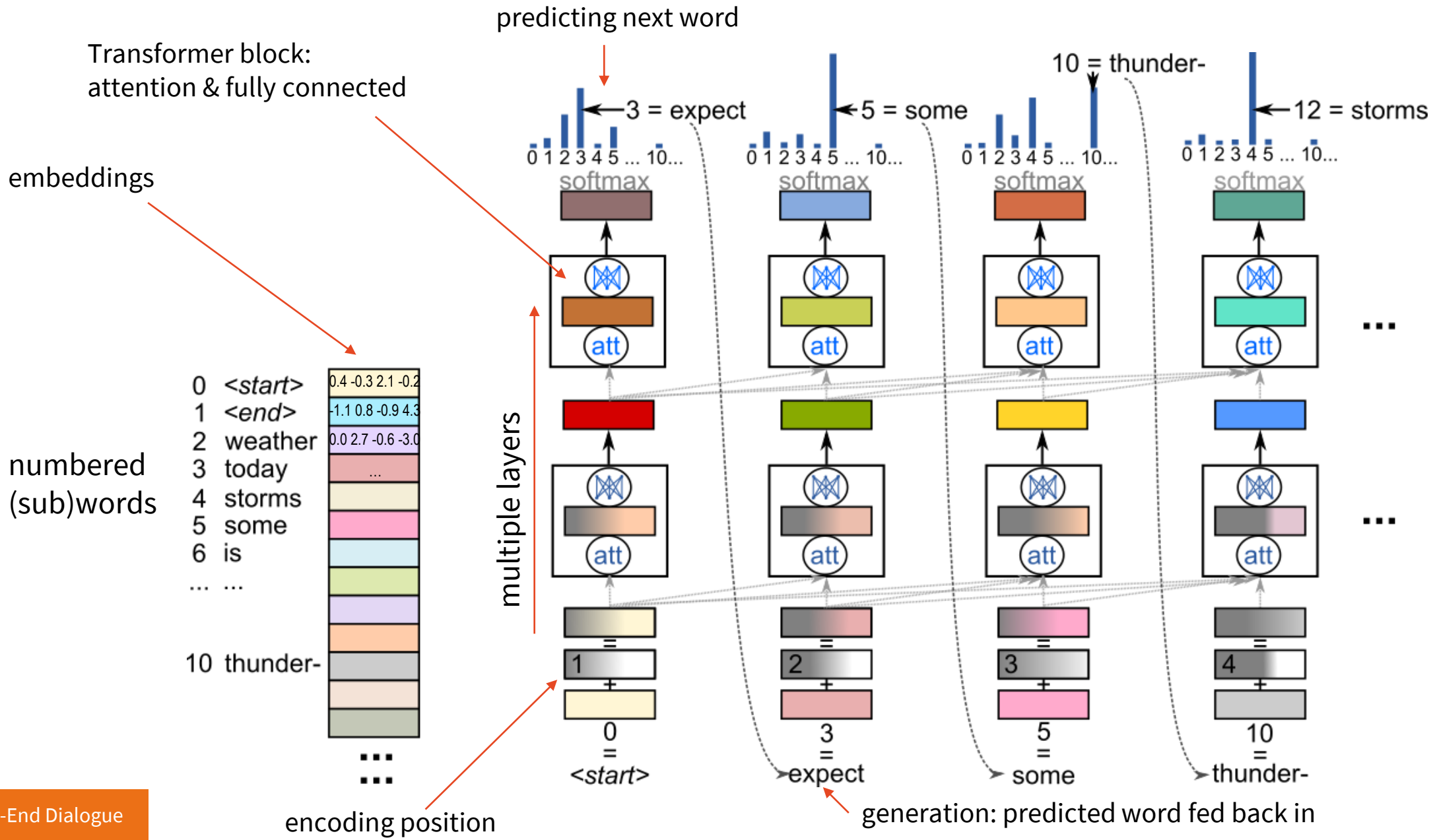
- **Transformer** neural architecture
  - (sub)word representation: **embedding** = vector of numbers
  - **blocks**: **attention** (combining context) + **fully-connected** (abstracting)
  - predicting next (sub)word = classification: choosing 1 out of ca. 50k (probabilistic)
  - trained from data: 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



Multi-domain dialogue (MultiWOZ)

- 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?

End-to-end Neural Dialogue with GPT-2

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

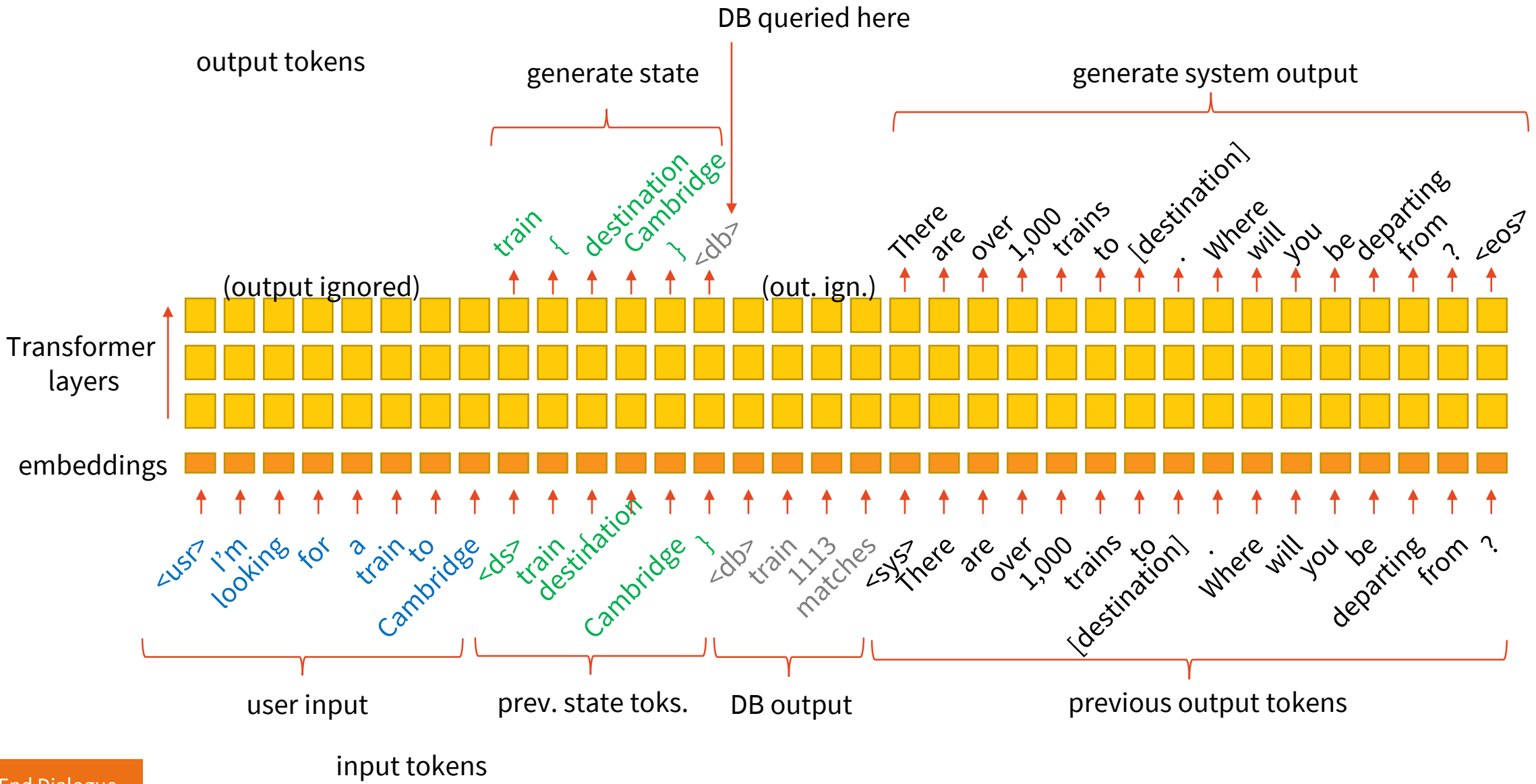
https://huggingface.co/gpt2

- **Dialogue**: GPT-2 finetuned (=further trained) on dialogue data (MultiWOZ)
- Single neural network, fits Transformer operation

https://github.com/ufal/augpt

- 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

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

Thanks

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

Credits:

Vojtěch Hudeček

Jonáš Kulhánek

Tomáš Nekvinda

Our work: https://github.com/ufal/augpt

These slides: https://bit.ly/e2e-ds-vocalls