Neural Networks for Dialogue Systems

ČSOB/KBC Data Science Bootcamp

Ondřej Dušek

13. 6. 2023

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

Dialogue Systems

- Definition: A *(spoken)* dialogue system is a **computer system designed to interact** with users **in** *(spoken)* **natural language**
  - Wide – covers lots of different cases

Task-oriented

- focused on completing a certain task/tasks
  - booking restaurants/flights, finding bus schedules, smart home…
- most actual DS in the wild & our focus today
- (typically) **single/multi domain** – 1/few topics

Non-task-oriented

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

Task-oriented Dialogue Example

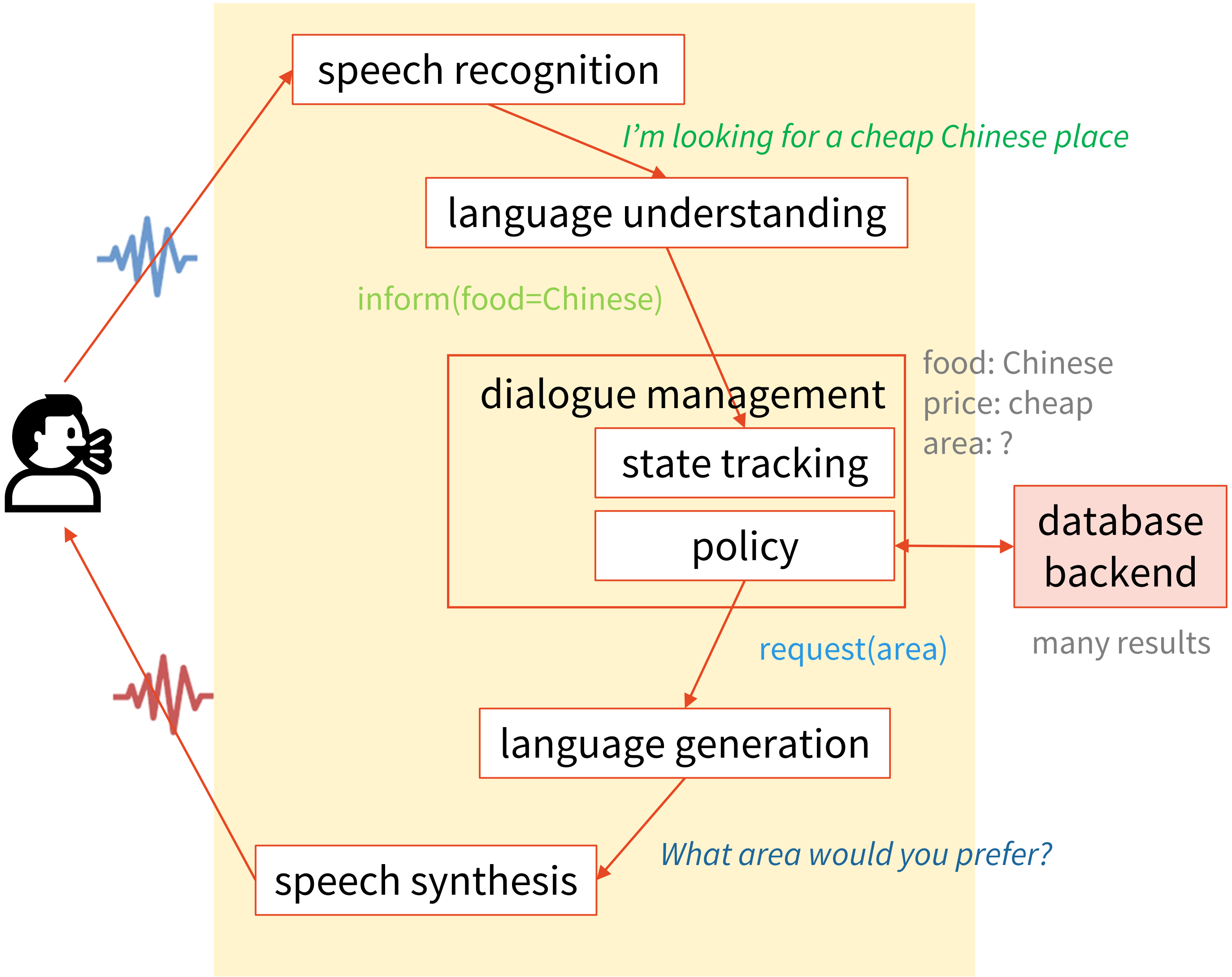
(Budzianowski et al., 2018)
https://aclanthology.org/D18-1547/

- 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?

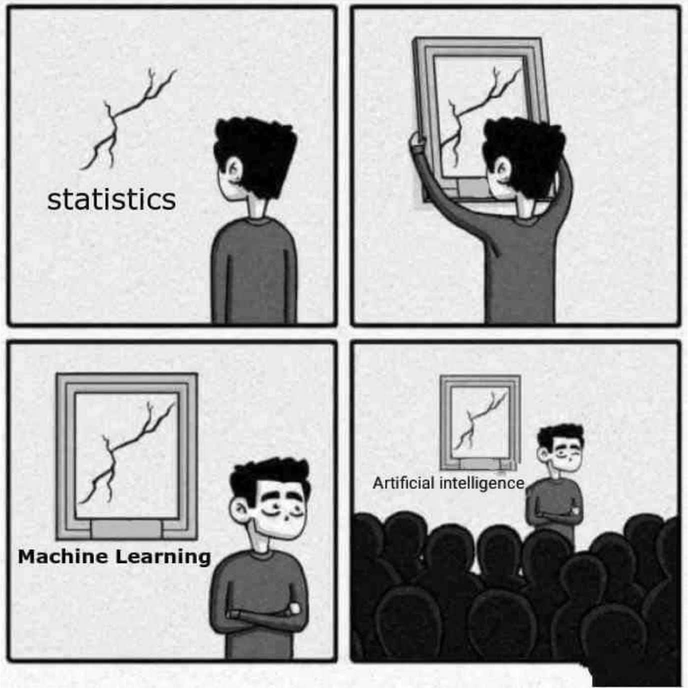
Dialogue Systems Architecture

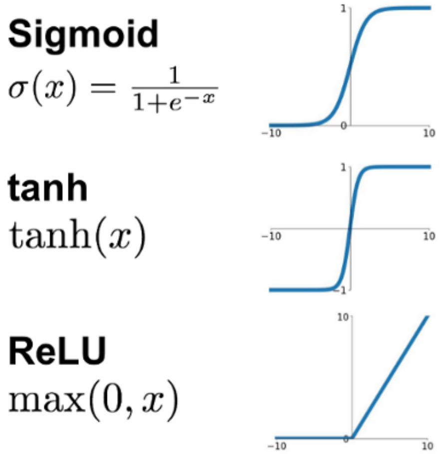
- traditional DS pipeline:
  - ASR: voice → text
  - **NLU: text → meaning**
  - **DM: meaning → reaction**
  - **NLG: reaction → text**
  - TTS: text → voice
- backend
  - needed for anything better than basic chit-chat
- text-based systems: NLU→DM→NLG only



Machine Learning & Neural Nets

- ML is basically function approximation
- function: data (**features**)→ **labels**
- **neural nets**: function = compound non-linear
  - basic building blocks (simple functions)
    - linear
    - nonlinearities: sigmoid, tanh, ReLU
    - softmax: probability estimates
  - stacked into layers
- training = adjusting function parameters to minimize error
  - **supervised learning** = based on data + labels given in advance
    - gradient descent, searching for minimal loss/cost
  - **reinforcement learning** = exploration & rewards given online

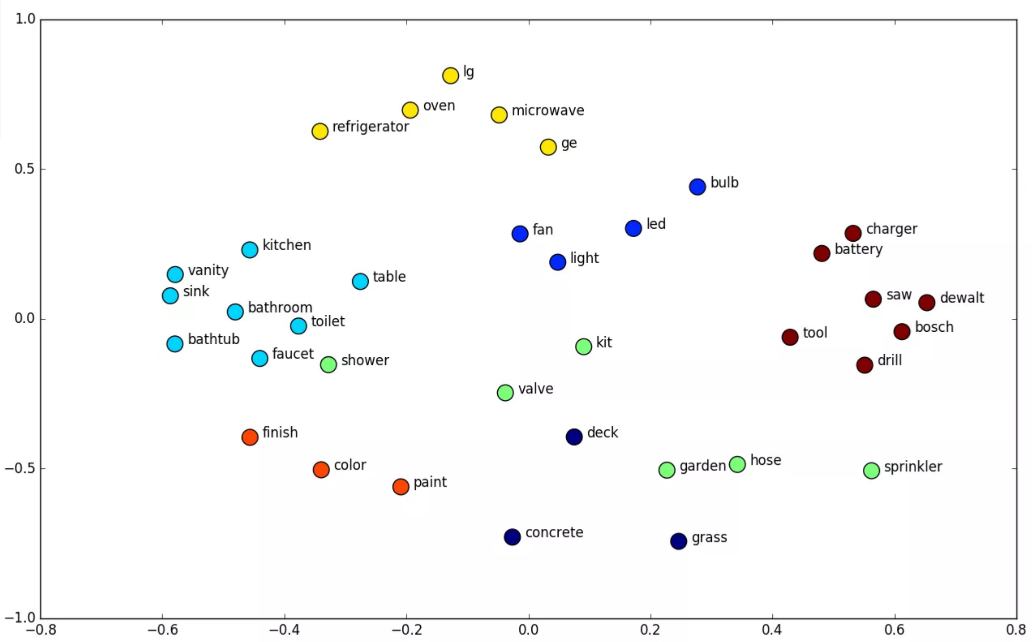




https://towardsdatascience.com/no-machine-learning-is-not-just-glorified-statistics-26d3952234e3
https://medium.com/@shrutijadon10104776/survey-on-activation-functions-for-deep-learning-9689331ba092

Representing Language: Embeddings

- distributed representation
  - **each word = a vector of floats**
  - basically an easy conversion of 1-hot → numeric
  - a dictionary of trainable features
- part of network parameters – trained
  a) pretraining (optional)
  b) training for the target task
- the network learns which words are used similarly – for the given task
  - they end up having close embedding values
  - different embeddings for different tasks
- embedding size: ~100s-1000
- vocab size: ~50-100k



http://blog.kaggle.com/2016/05/18/home-depot-product-search-relevance-winners-interview-1st-place-alex-andreas-nurlan/

http://ruder.io/word-embeddings-2017/

Subwords

- vocabulary is unlimited, embedding matrix isn't
  - + the bigger the embedding matrix, the slower your models
- Special **out-of-vocabulary token** *<unk>*
  - loses information, we don't want it on the output
- **Subwords:** groups of characters that
  - make shorter sequences than using individual characters
  - cover everything
  - 20-50k subwords for 1 language, ~250k subwords multilingual
- **Byte-pair Encoding** (=one way to get subwords)
  - start from individual characters
  - iteratively merge most frequent bigram, until you get desired # of subwords

f a s t _
f a s t e r _
t a l l _
t a l l e r _

→

fast er _
tall er _
s l o w er _
tall e s t _

(Sennrich et al., 2016)
https://www.aclweb.org/anthology/P16-1162/

(Kudo, 2018)
https://aclanthology.org/P18-1007

Transformer Language Models

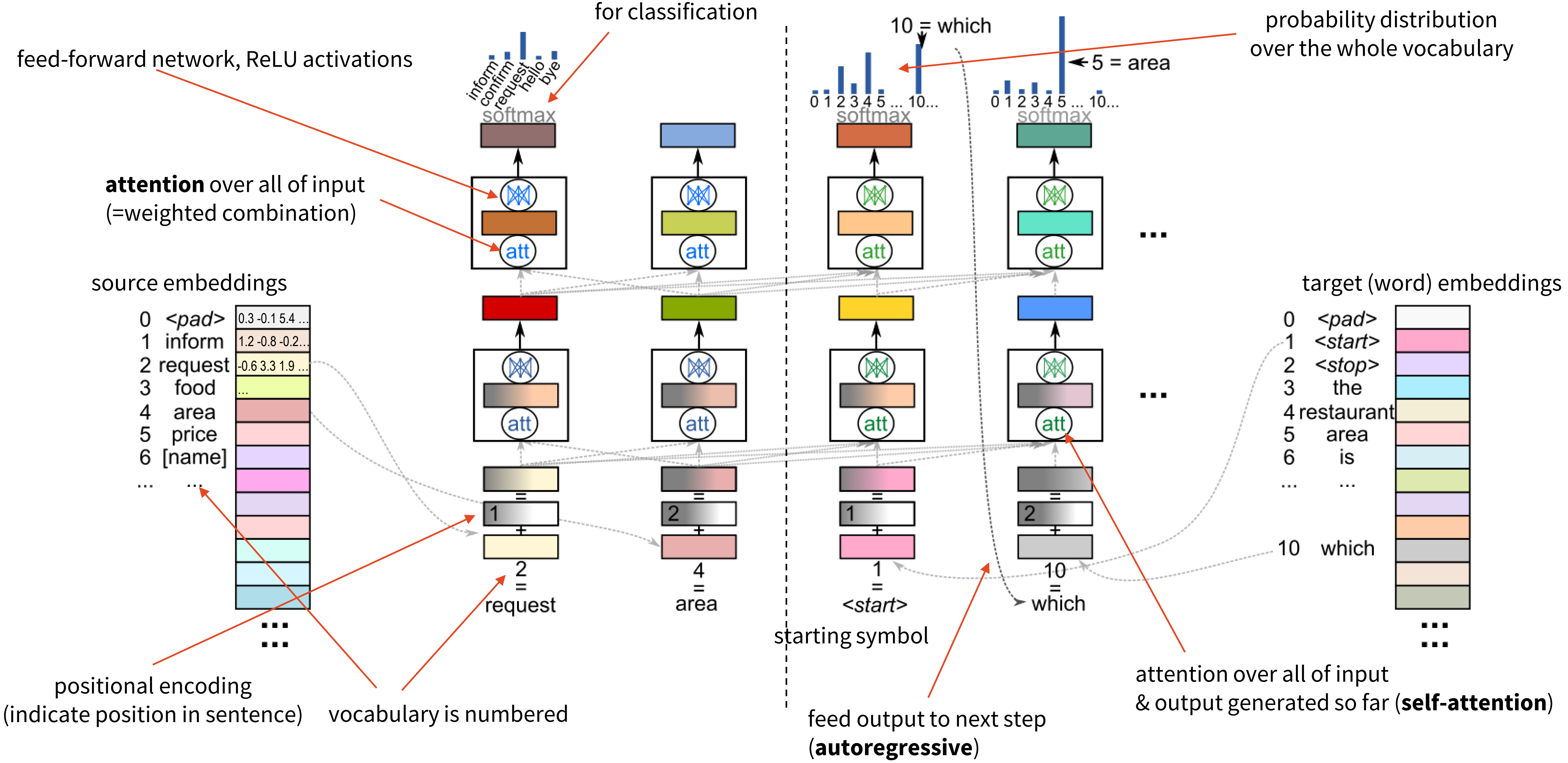
- **Transformer** neural architecture
  - (sub)word representation: **embedding** = vector of numbers
  - **blocks**: **attention** (combining context) + **fully-connected** (abstracting)
    - 6-48 identical layers
  - further task-specific layers on top
  - **predicting next (sub)word** = classification: choosing 1 out of ca. 50k (low level!)
    - autoregressive – word by word, feed output into next step
  - trained from data: initialize randomly & iteratively improve
- Variants:
  - Encoder-only – good for classification/token tagging
  - Decoder-only – good for generation
  - Encoder-Decoder – good for generation, translation

Neural NLG: Transformer Models

(Vaswani et al., 2017) http://arxiv.org/abs/1706.03762

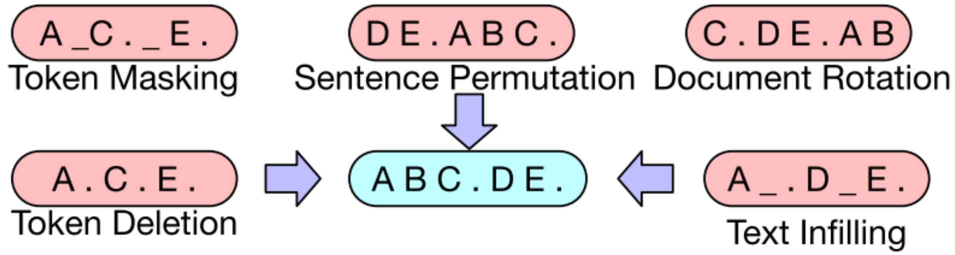
1) **encoder:** encode linearized data

2) **decoder:** decode text word-by-word



Pretrained Language Models

- **Self-supervised pretraining:** supervised, but needs no extra annotation
    - masked word prediction
    - next word prediction
    - fix corrupt sentences
    - predict sentence order
  - huge amounts of data – many GBs of text



- **Finetuning:** just train further, use data for target task
- Lots of models free to download
  - **BERT, RoBERTa(-XLM)** – encoder only
    - lots of language-specific variants (incl. Czert/RobeCzech/small-e-czech)
  - **GPT**(**-2**/**-j**/**-neo**): decoder only, next-word prediction
  - **(m)BART, (m)T5**: encoder-decoder

https://github.com/huggingface/transformers

(Devlin et al., 2019) https://www.aclweb.org/anthology/N19-1423 (Liu et al., 2019) http://arxiv.org/abs/1907.11692
(Radford et al., 2019) https://openai.com/blog/better-language-models/
(Lewis et al., 2020) http://arxiv.org/abs/1910.13461 (Raffel et al., 2019) http://arxiv.org/abs/1910.10683

Natural/Spoken Language understanding (NLU/SLU)

- **Extracting the meaning** from the (now textual) user utterance
- Converting into a structured semantic representation
  - **dialogue acts**:
    - **intent**/act type (*inform, request, confirm*)
    - **slot**/entity/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)

- Specific steps:
  - **named entity resolution** (NER)
    - identifying task-relevant names (*London, Saturday*)
  - **coreference resolution**
    - (“*it*” –> “*the restaurant*”)

NLU Challenges

- non-grammaticality *find something cheap for kids should be allowed*
- disfluencies
  - hesitations – pauses, fillers, repetitions *uhm I want something in the west the west part of town*
  - fragments *uhm I'm looking for a cheap*
  - self-repairs (~6%!) *uhm find something uhm something cheap no I mean moderate*
- ASR errors *I'm looking for a for a chip Chinese rest or rant*
- synonymy *Chinese city centre*
  I've been wondering if you could find me a restaurant that has Chinese food close to the city centre please
- out-of-domain utterances *oh yeah I've heard about that place my son was there last month*

NLU basics

- You can get far with keywords/regexes (for a limited domain)
- **Intent classification**
  - Transformers, PLMs: typically over 1st input element (start-of-sentence token)
- **Slot value detection**
  - classification (binary: *“is slot value X present?”*)
  - **slot tagging** – classify every token
    BIO/IOB scheme: beginning (+slot) – inside (+slot) – outside

I	*need*	*a*	*flight*	*from*	*Boston*	*to*	*New*	*York*	*tomorrow*
O	**O**	**O**	**O**	**O**	**B-dept**	**O**	**B-arr**	**I-arr**	**B-date**

- **Delexicalization**: replacing slot values by placeholders
  - essentially named entity recognition
  - essentially tagging, but typically done by dictionaries

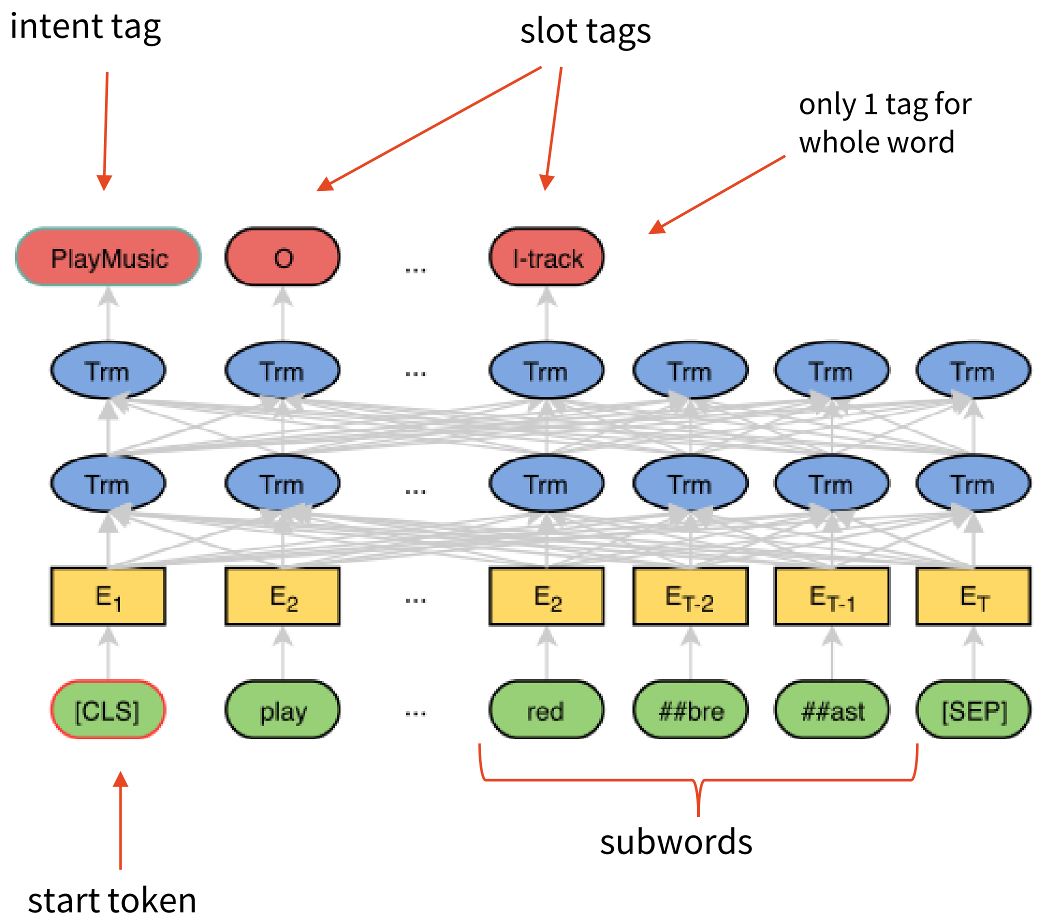
I’m looking for a Japanese restaurant in Notting Hill.
I’m looking for a <food> restaurant in <area>.

I need to leave after 12:00.
I need to leave after <time>.
(= not necessarily 1:1 with slots)

BERT-based NLU

(Chen et al., 2019)
http://arxiv.org/abs/1902.10909

- combined intent-slot
- slot tagging on top of pretrained BERT
  - standard **IOB approach**
  - feed last BERT layers to **softmax over tags**
    - classify only at 1st subword in case of split words (don't want tag changes mid-word)
- special start token tagged with intent
  - again, softmax on top of last BERT layer
- finetune both tasks at once
  - essentially same task,
    just having different labels on the 1st token ☺



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 NLU output
    - more robust, more complex
    - neural nets – often replaces NLU

price: cheap
food: Chinese
area: riverside

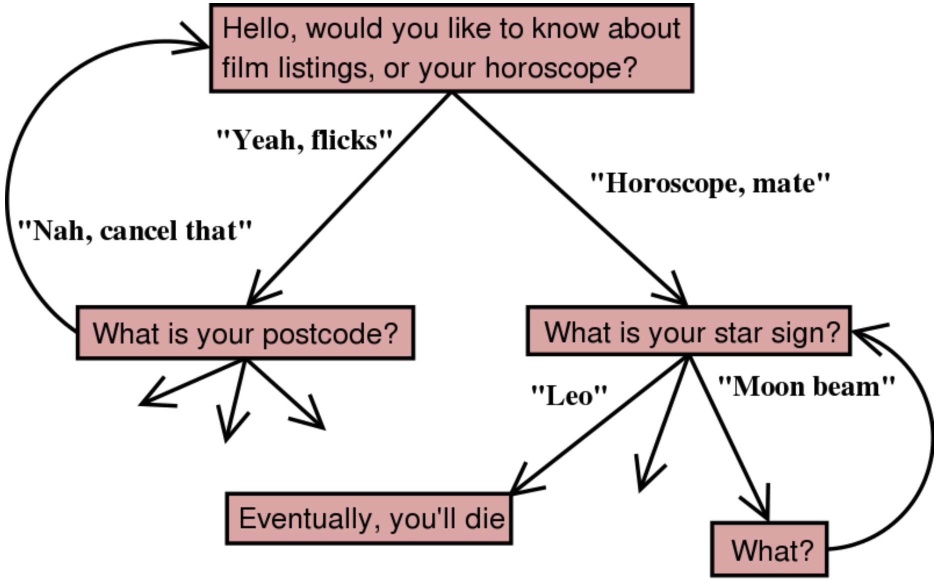
price: 0.8 cheap
0.1 moderate
0.1 <null>
food: 0.7 Chinese
0.3 Vietnamese
area: 0.5 riverside
0.3 <null>
0.2 city center

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

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/

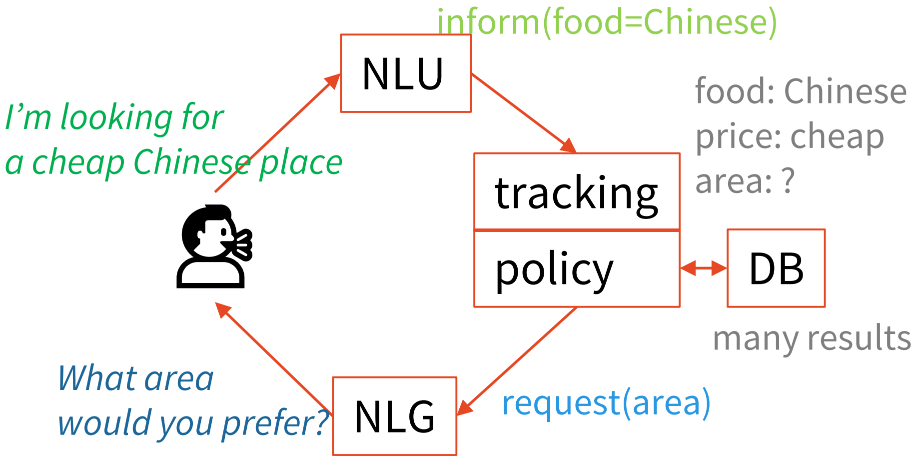
https://www.ibm.com/products/watson-assistant

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

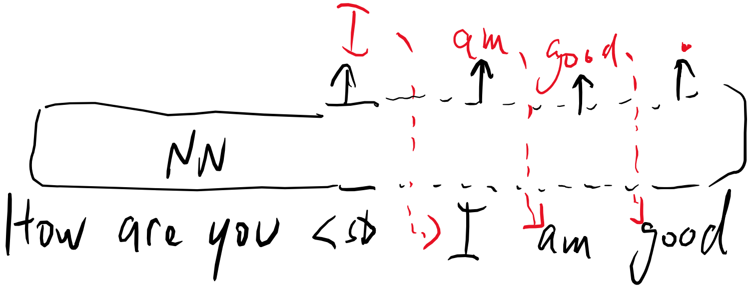
https://developer.amazon.com/alexa-skills-kit

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



End-to-end Neural Dialogue with GPT-2

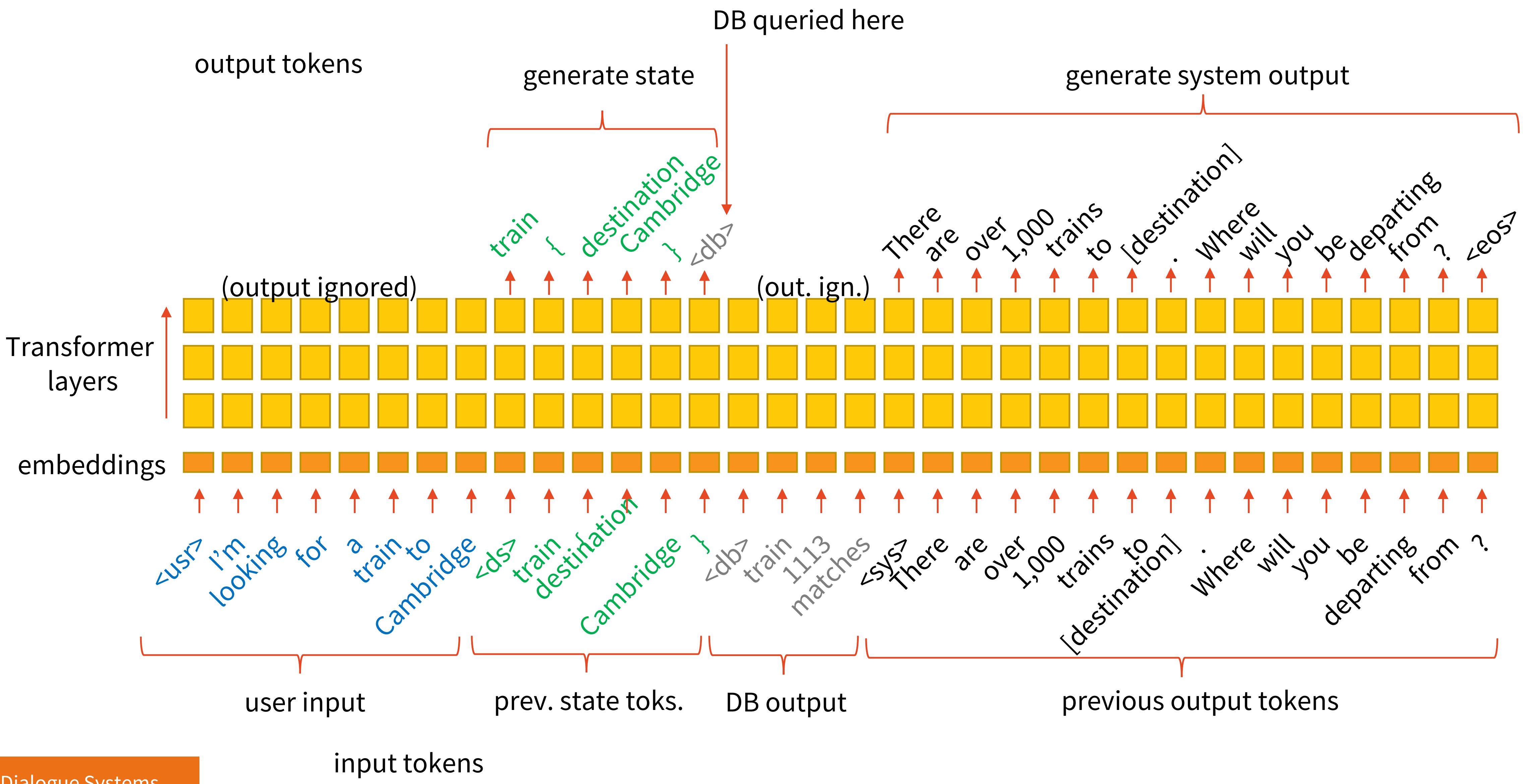
- **GPT-2**: one of most popular pretrained language models
  - Transformer (100M-1.5B params)
  - pretrained on next-word prediction
  - 8M docs, 40GB data from the web

(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

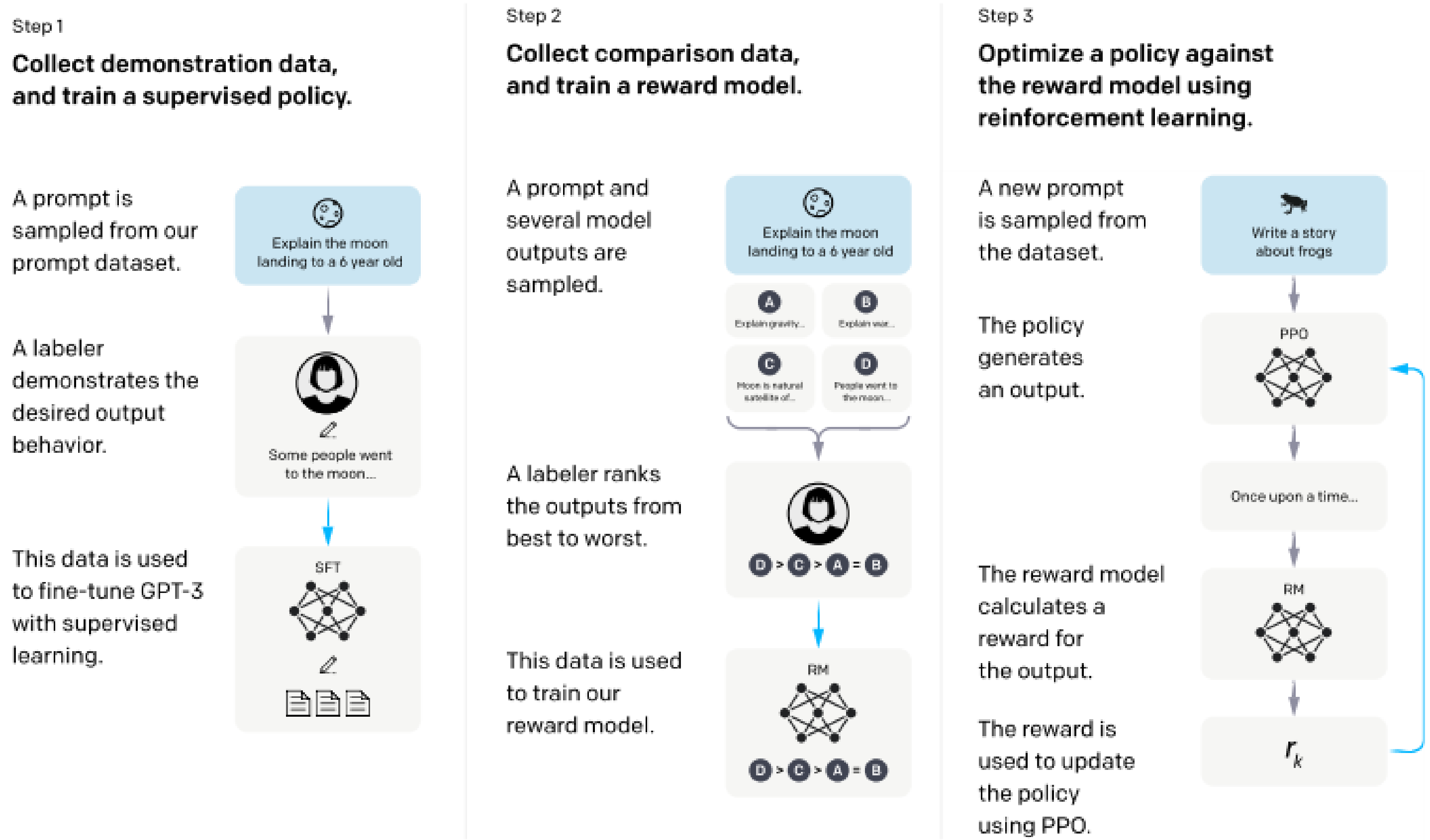
- Larger than GPT2 size: 10-100B parameters
  - hard to run in-house (**OPT**, **BLOOM**, **LlaMa**) or not free (**GPT-3/4**, **ChatGPT**, **LaMDa**…)
  - ~10B can run (not train) on regular GPUs with 8-bit mode https://huggingface.co/blog/hf-bitsandbytes-integration
- Transformer architecture, again
  - 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
- same problems as finetuning – hard to control

Instruction Tuning & RLHF

(Ouyang et al., 2022) http://arxiv.org/abs/2203.02155
https://openai.com/blog/chatgpt/ , https://www.youtube.com/watch?v=0A8ljAkdFtg
https://crfm.stanford.edu/2023/03/13/alpaca.html , https://open-assistant.io/

- Improved (pre)training of LLMs (InstructGPT/ChatGPT, Alpaca, OpenAssistant…)
  - better replies, not as low-level as next-word prediction
  - can be safer
  - more efficient (smaller, stronger models)

1) finetuning
   - standard, but better data – instructions & solutions
2) evaluation/ranker model
   - lots of rankings
   - used to train RL reward model
3) RL with reward model



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.

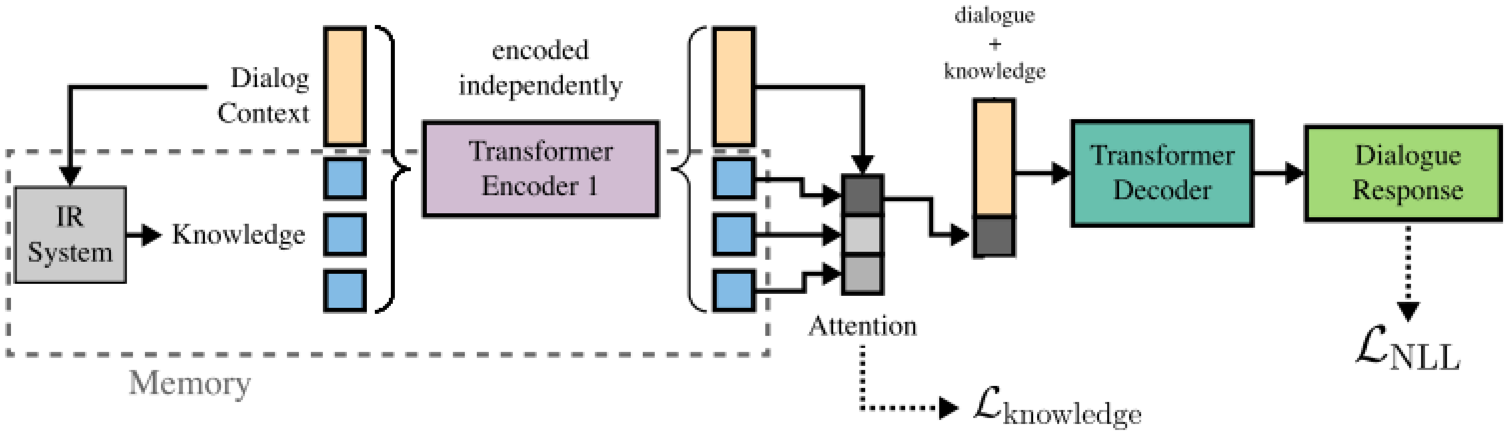
There once was a programmer in Vim
Whose exit strategy was looking grim
They tried to escape
But couldn't find the right shape
Until they remembered to type ":q!" and win!

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

Summary

- SotA = Transformer architecture & pretrained language models
  - self-supervised pretraining, finetuned for task
- Component models include NNs
  - NLU = encoder models & classification / token tagging
  - DM = state tracking ~ rules or NLU-like, policy ~ rules / RL
  - NLG = (encoder-)decoder models
- End-to-end models
  - 2-step: decode state, decode reply, query DB in between
- LLMs = same but larger
  - allows prompting
  - instruction finetuning – better training

Thanks

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

Practicals:

http://bit.ly/atis-nlu-empty

http://bit.ly/atis-nlu-filled

Slides:

http://bit.ly/nn-sds-slides

References/Inspiration/Further:

- Jurafsky & Martin: Speech & Language processing. 3rd ed. draft 2021 (https://web.stanford.edu/~jurafsky/slp3/)
- McTear: Conversational AI. Morgan & Claypool 2021 (https://doi.org/10.2200/S01060ED1V01Y202010HLT048)
- Gao et al.: Neural Approaches to Conversational AI, 2019 (http://arxiv.org/abs/1809.08267)
- 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
- 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
- Gina-Anne Levow (University of Washington): https://courses.washington.edu/ling575/