Overview of Natural Language Processing Part II & ACS L90

Lecture 12:Natural Language Generation

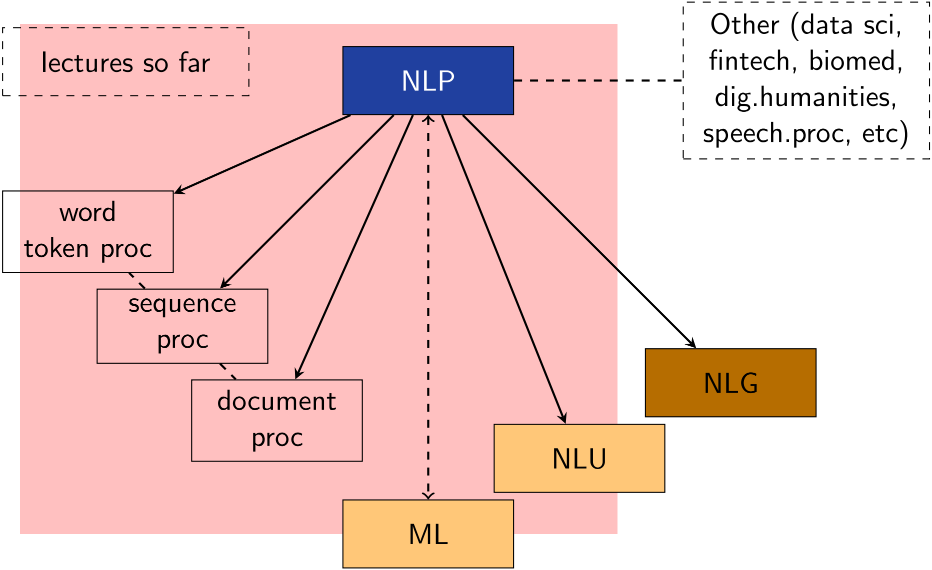
Andrew Caines

some slides by Paula Buttery & Laura Rimell

Department of Computer Science and Technology
University of Cambridge

Michaelmas 2021/22

NLG within NLP (over-simplification)



Part 1: Why generate natural language?

Why NLG?

Applications driven:

- Might be theory driven but usually an application such as:
  - Dialogue systems / chatbots (main focus today)
  - Paraphrasing
  - Summarisation ('abstractive' as opposed to 'extractive')
  - Question answering
  - Multimodal language generation (images, captions)
  - Generation for text or spoken output: speech synthesis
  - Machine translation (spoken)
  - Assistive technology
  - AI authoring (!)

What is NLG?

Producing words in response to some prompt

- A prompt might be:
  - A conversational turn
  - A complex sentence for paraphrasing
  - A news article for summarising
  - A factoid question
  - A *source language* sequence for translation into a *target language* sequence
  - An image for captioning
  - A story beginning: *It was a crisp, Autumn morning in Cambridge...*

How to do NLG?

Broadly speaking, three approaches:

- Classical: hand-written rules, limited domains
  - e.g. the ELIZA chatbot
- Templates: highly practical, reliable, but not highly engaging
  - e.g. ticket booking systems
- Machine learning: sequence to sequence, or incremental generation
  - e.g. Google Translate, sentence completion
  - Industry shift, industry-led? (big-tech / research / industry)

Part 2: Dialogue Systems

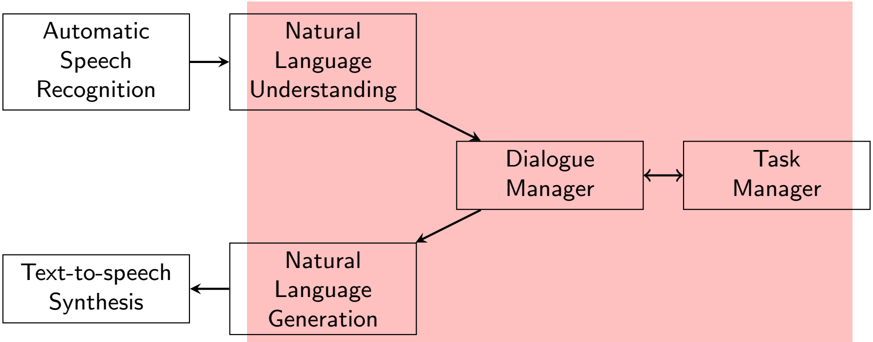
ELIZA, Weizenbaum 1966

- One of the earliest chatbots
- The DOCTOR script imitates responses of a Rogerian psychotherapist
- Capable of a limited form of conversation with a user, by reflecting user turns
- Pattern matching transforms user input to system utterances: identify keywords (which are ranked), decompose the input string and reassemble output string

User:	I am anxious about COVID-19.
ELIZA:	Do you believe it is normal to be anxious about COVID-19?
User:	Yes I do, but not everyone appears to be.
ELIZA:	Are you sure about that?
User:	I think so.
ELIZA:	Do you really think so?
User:	Yes.
ELIZA	I see. Let's explore that a bit more.

* Chat with an ELIZA DOCTOR at https://web.njit.edu/~ronkowit/eliza.html

Dialogue systems in general



- Classic modular approach (each module could be a lecture series of its own)
- Our focus: the dialogue/task manager in order to talk about NLG, a little of the NLU
- NB: recent approaches tend to be all-in-one (end-to-end)

Frame-based dialogue managers

- Task-based: e.g. airplane tickets or cinema screenings
- System needs to fill a number of slots in a frame: from the NLU module
- Required slots must be filled; optional slots should be capable of being filled if a user provides the information
- Required slots are associated with a question (fuzzy boundary with the NLG component)

SLOT	Response	Question
ORIGIN CITY	*required*	From what city are you leaving?
DESTINATION CITY	*required*	Where are you going?
DEPARTURE TIME	*required*	When would you like to leave?
(AIRLINE)	*optional*	...
(PRICE POINT)	*optional*	...

Frame-based dialogue managers

- Frame-based dialogue managers must keep the current state of the dialogue
- When all required information has been obtained from the user, a database query can be made, regardless of number of turns

S: Which city are you leaving from?
U: Baltimore, going to Santa Fe, and I'd like to fly United if possible.
S: When would you like to leave?

SLOT	Response	Question
ORIGIN CITY	**Baltimore**	From what city are you leaving?
DESTINATION CITY	**Santa Fe**	Where are you going?
DEPARTURE TIME	*required*	When would you like to leave?
(AIRLINE)	**United**	...
(PRICE POINT)	*optional*	...

Frame-based dialogue managers

- NLG straightforward so far, now a more advanced problem –
- Confirmation and dealing with errors: explicit confirmation

S: Which city do you want to leave from?
U: Baltimore.
S: **Do you want to leave from Baltimore?**
U: Yes.

U: I'd like to fly from Denver Colorado to New York City on September twenty first in the morning on United Airlines
S: **Let's see then. I have you going from Denver Colorado to New York on September twenty first. Is that correct?**
U: Yes.

- Easy for users to correct misunderstandings – simply answer 'no', and the system must try again to fill the slot(s)
- But the whole dialogue becomes cumbersome

Frame-based dialogue managers

- Implicit confirmation

  U: I want to travel to Berlin.
  S: **When do you want to travel to Berlin?**

  U: Hi I'd like to fly to Seattle Tuesday morning
  S: **Travelling to Seattle on Tuesday, August eleventh in the morning.** *Your full name?*

- Implicit confirmation is more natural and makes a shorter dialogue
- Modern systems decide when to use explicit or implicit confirmation based on confidence level

Frame-based dialogue managers

- Frame-based GUS architecture for task-based dialogues (Bobrow et al, 1977)
- Long-lived and highly-influential: industry standard for $>$30 years
- *The NLG so far*: mostly pre-determined, constrained, non-creative, mainly *prompts* for information plus some opening/closing ‘speech acts’ (J&M 3rd edn §24.1)
- Heavy-lifting: templates and dialogue designers
- Pros: high precision, (hopefully) crafted by experts, sufficient coverage for customer service success/savings
- Cons: time-consuming and expensive, potentially low recall, unlikely to be engaging
- Modern systems inspired by GUS and the dialogue frame: Siri, Alexa, Google Assistant
- Slot-filling via supervised machine learning instead of rules

Dialogue-state systems

- Modern dialogue systems: doing dialogue-state or 'belief-state' tracking
- NLU extracting slot fillers from user turns, likely using machine learning and rules in a hybrid fashion
- Dialogue-state tracker maintains the current conversation state, including slot-filler constraints expressed by the user so far
- Dialogue policy component more sophisticated: deciding when to ask for clarification, answer question, make a suggestion, etc
- NLG: take **content plan** (what to say) from policy module, do **sentence realisation** (how to say it)

Dialogue-state systems

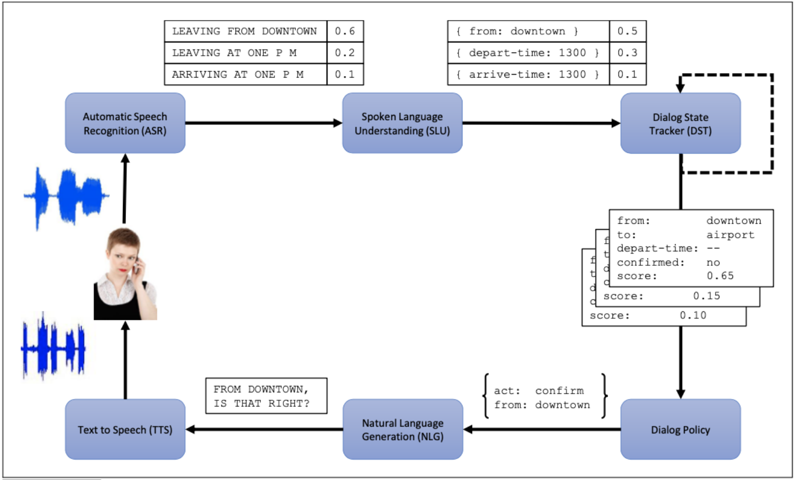


Figure 24.12 Architecture of a dialogue-state system for task-oriented dialogue from Williams et al. (2016a).

Source: J&M 3rd edn chapter 24

Dialogue-state systems: slot-filling NLU

NLU extracting slot fillers from user turns, construed as a BIO sequence labelling task: input encoded as contextual embeddings → feedforward → softmax→ string extraction (& normalisation, e.g. 'San Francisco' to 'SFO')

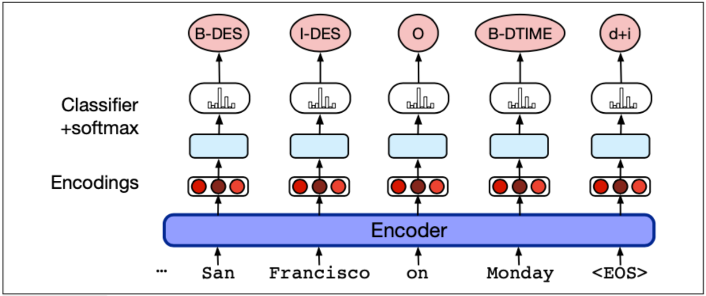


Figure 24.15 A simple architecture for slot filling, mapping the words in the input through contextual embeddings like BERT to an output classifier layer (which can be linear or something more complex), followed by softmax to generate a series of BIO tags (and including a final state consisting of a domain concatenated with an intent).

Source: J&M 3rd edn chapter 24

Dialogue-state systems: slot-filling NLU

- In practice: bootstrap machine learned NLU from GUS-like rule-based system
- Start with rules and an annotated test set for continuous evaluation
- New user inputs paired with labels provided by rule-based system
  $\rightarrow$ training instances for machine learning
- Discover whether the machine learned NLU outperforms rule-based NLU (and replace)
- *Takeaway*: cannot assume ML will be best at the outset, when faced with a 'cold-start' situation

Dialogue-state systems: DST

- Dialogue-state tracker maintains the current conversation state: all slot-filler constraints so far plus the latest **dialogue act** (inspired by 'speech acts' and conversation analysis in Linguistics/Philosophy)
- Supervised classification of dialogue acts based on rep of current input & previous acts
- (nb. dialogue-state tracking challenges & Cambridge-based corpora)

User:	I'm looking for a cheaper restaurant
	`inform(price=cheap)`
System:	Sure. What kind - and where?
User:	Thai food, somewhere downtown
	`inform(price=cheap, food=Thai, area=centre)`
System:	The House serves cheap Thai food
User:	Where is it?
	`inform(price=cheap, food=Thai, area=centre); request(address)`
System:	The House is at 106 Regent Street

Source: J&M 3rd edn chapter 24 (from Mrkšić et al, ACL 2017)

Dialogue-state systems: dialogue policy

- Dialogue policy: what action A to take next?
- Could formalise as:

$$\hat{A}_i = \operatorname*{arg\,max}_{A_i \in A} P(A_i | \mathrm{Frame}_i, U_i, A_{i-1})$$

- Where A is a system dialogue act, U is a user dialogue act, and the Frame is the current state of slots and fillers
- Can adopt a supervised learning approach (also: reinforcement learning, rewards for correct actions)

  $\rightarrow$ Content planning: what to say, a dialogue act with attributes (slots and values)

Dialogue-state systems: NLG

- Content planning: what to say, a dialogue act with attributes (slots and values)

  → Sentence realisation: how to say it, trained on many content.plan.rep/sentence pairs from a labelled corpus (usually crowdsourced)
- Improve generalisation by **de-lexicalising** the training examples: replacing slot values with placeholders:

	recommend(restaurant name= Au Midi, neighborhood = midtown, cuisine = french
1	restaurant_name is in neighborhood and serves cuisine food.
2	There is a cuisine restaurant in neighborhood called restaurant_name.

Figure 24.17 Delexicalized sentences that can be used for generating many different relexicalized sentences. From the restaurant recommendation system of Nayak et al. (2017).

Source: J&M 3rd edn chapter 24

Dialogue-state systems: NLG

- From frames to de-lexicalised sentences as sequence-to-sequence problem (seq2seq)
- i.e. encoding a flattened input (dialogue act & arguments) with LSTM, decoding to de-lexicalised string with LSTM
- Re-lexicalised in a post-processing step: use the values from the current frame to fill in the gaps: e.g. *Au Midi has decent service*

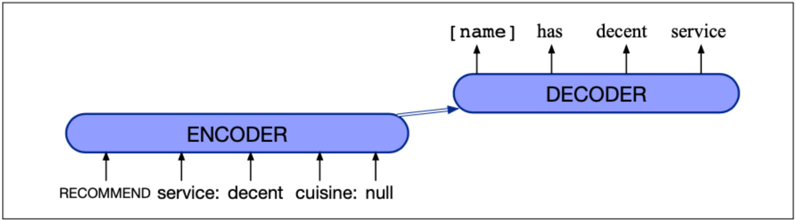


Figure 24.18 An encoder decoder sentence realizer mapping slots/fillers to English.

Source: J&M 3rd edn chapter 24 (from Nayak et al, INTERSPEECH 2017)

Evaluating dialogue systems

- For task-based scenarios:
- Human evaluation via questionnaires: e.g. ease of use, pace, expected behaviour, future use
- Success rate: proportion of correctly filled slots
- Efficiency cost: number of turns, or time taken to achieve task; proportion of user/system turns used for correction
- Automatic metrics: depend on idea of corpus-based 'reference' utterances, but we know that human conversation involves many possible responses at each turn, auto-metrics shown not to correlate well with human judgements (but they are widely used)

Evaluating dialogue systems

- Auto-metrics: BLEU-2 & word embedding average vs humans
  - BLEU: geometric mean of n-gram precision scores (clipped) b/w reference & candidate, $\times$ brevity penalty
  - Embedding ave: cosine similarity between sequence reps of reference and candidate (e.g. mean word embedding of tokens in seq)

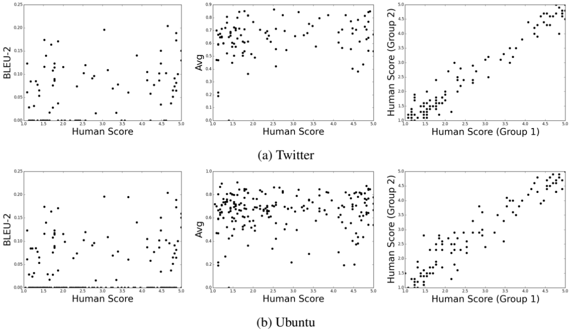


Figure 1: Scatter plots showing the correlation between metrics and human judgements on the Twitter corpus (a) and Ubuntu Dialogue Corpus (b). The plots represent BLEU-2 (left), embedding average (center), and correlation between two randomly selected halves of human respondents (right).

Liu et al, EMNLP 2016: How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation

Evaluating dialogue systems

- Auto-metric: **adversarial evaluation**, inspired by Turing test
- Train a classifier to distinguish between human and machine-generated turns
- The more successful the machine at fooling the classifier, the better it is
- Adversarial approach can also be used in reinforcement learning setting: training a *generator* and a *discriminator* with the output of the latter providing rewards for the former, pushing it towards more human-like dialogue

  Li et al, EMNLP 2017, Adversarial Learning for Neural Dialogue Generation

Open-domain chatbots

- How about dialogue systems for non-task-oriented chit-chat? (open-domain)
- Several approaches taken
- Retrieval approaches: some dialogue history as input, select next utterance from a large candidate set

  e.g. 'Poly-encoder' transformer model trained on 800M sentences from Reddit; fine-tuned on PersonaChat (being assigned some personality traits and chatting), Wizard of Wikipedia (discussing topics in depth), & Empathetic Dialogues (demonstrating empathy and talking about emotional situations); Humeau et al, ICLR 2020; Smith et al, ACL 2020.

Open-domain chatbots

- Generation approaches: seq2seq models trained with cross-entropy to generate responses word-by-word conditioned on dialogue context
  e.g. hierarchical recurrent encoder-decoder (Serban et al, AAAI 2016) with (1) **encoder**, (2) **context**, (3) **decoder** RNNs:

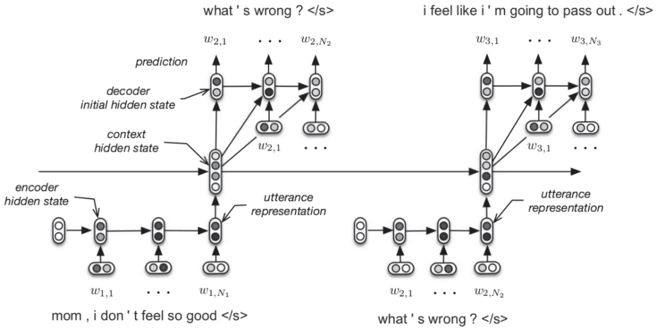


Figure 1: The computational graph of the HRED architecture for a dialogue composed of three turns. Each utterance is encoded into a dense vector and then mapped into the dialogue context, which is used to decode (generate) the tokens in the next utterance. The encoder RNN encodes the tokens appearing within the utterance, and the context RNN encodes the temporal structure of the utterances appearing so far in the dialogue, allowing information and gradients to flow over longer time spans. The decoder predicts one token at a time using a RNN. Adapted from Sordoni et al. (2015a).

Open-domain chatbots

- Generation approaches: as **sequence completion**

  i.e. given a **prefix** of tokens $\mathbf{x}_{1:k}$, we will use language model p_θ to conditionally decode a **continuation** $\mathbf{x}_{k+1:N}$, giving the **completion** $(x_1, \ldots, x_k, \hat{x}_{k+1}, \ldots, \hat{x}_N)$
- Generally applicable to question answering, story generation, cloze tests (gap filling) or sentence completion
- And dialogue modelling where $\mathbf{x}_{1:k}$ is (some of) a dialogue history, continuation is the next utterance
- Deterministic or stochastic decoding methods to generate continuations step-by-step: e.g. **greedy search** (highest prob token at each step); **beam search** maintains fixed-size set of partially-decoded sequences and scores each token in vocab appended to each one; or **sample** n most probable tokens

Open-domain chatbots

- Known problems:
  - Generative models tend to produce dull and repetitive responses – more likely to output frequent words, less likely to produce rare words compared to humans
  - Might hallucinate knowledge: confidently state incorrect information
  - & no access to external knowledge beyond model parameters: e.g. a pre-2020 dataset unlikely to feature COVID-19
  - Can be forgetful or self-contradictory: 'goldfish memory' problem
  - Can be biased, offensive, inappropriate

Open-domain chatbots

- Retrieve-and-refine: classic dialogue retrieval but not shown to user, instead appended to context turn and 'more vibrant' response generated based off this input-candidate string
- e.g. open-source chatbot by Facebook Research: BlenderBot (Roller et al, EACL 2021)
- v1: pre-trained on Reddit, fine-tuned on PersonaChat, Empathetic Dialogues, Wizard of Wikipedia, Blended Skill Talk (how to combine the previous skills – divulging personal information, showing empathy, displaying knowledge)
- New in v2: external information from internet searches, generating relevant queries given conversation context
- And a long-term memory to enable coherence across multiple conversation sessions: Multi-Session Chat dataset provides supervision as to what to store, and good dialogue turns given those memories
- More info and pre-trained model: `https://parl.ai/projects/blenderbot2/`

Open-domain chatbots

- Back to the known problems:
  - Generative models tend to produce dull and repetitive responses – more likely to output frequent words, less likely to produce rare words compared to humans
    $\rightarrow$ unlikelihood loss: a set of negative candidates for each step, mix likelihood & unlikelihood losses, *likelihood pushes up the probability of tokens in ground-truth response to context x, unlikelihood pushes down the probability of tokens in negative candidate responses to x* (Welleck et al, ICLR 2020)
    $\rightarrow$ also the idea of controlled generation with pre-defined control codes to narrow the domain: e.g. CTRL chatbot by Salesforce Research (Keskar et al, 2019)

Open-domain chatbots

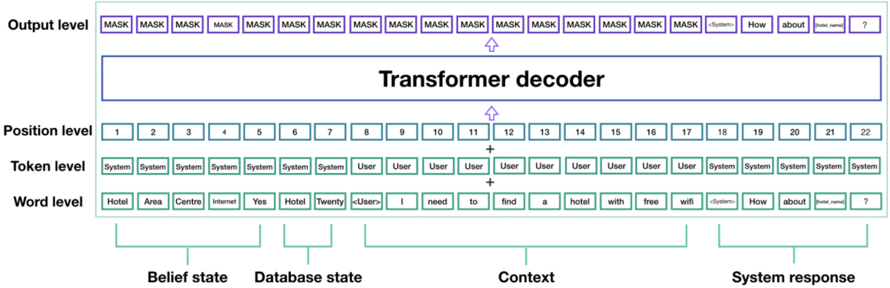
- Back to the known problems:
  - Might hallucinate knowledge: confidently state incorrect information
  - & no access to external knowledge beyond model parameters: e.g. a pre-2020 dataset unlikely to feature COVID-19
    → BlenderBot: Komeili et al, 'Internet-Augmented Dialogue Generation'
  - Can be forgetful or self-contradictory: 'goldfish memory' problem
    → unlikelihood training with labels from natural language inference {`entailment, neutral, contradiction`} crowdsourced on PersonaChat: persona statements & dialogue turns (Li et al, ACL 2020)
    → BlenderBot: Xu et al, 'Beyond Goldfish Memory: Long-Term Open-Domain Conversation'

Open-domain chatbots

- Back to the known problems:
  - Can be biased, offensive, inappropriate
    $\rightarrow$ dealing with gender bias: combination of counterfactual data augmentation, targeted data collection, bias controlled training (Dinan et al, EMNLP 2020)
    $\rightarrow$ detecting 'unsafe' dialogue turns with human-in-the-loop 'build it, break it, fix it' strategy: {train offence-detection model, 'beat the system' with crowdworkers, fix by re-training}, repeat (Dinan et al, EMNLP 2019)
    $\rightarrow$ or through 'negative training' to fine-tune against undesirable behaviour (He & Glass, ACL 2020)

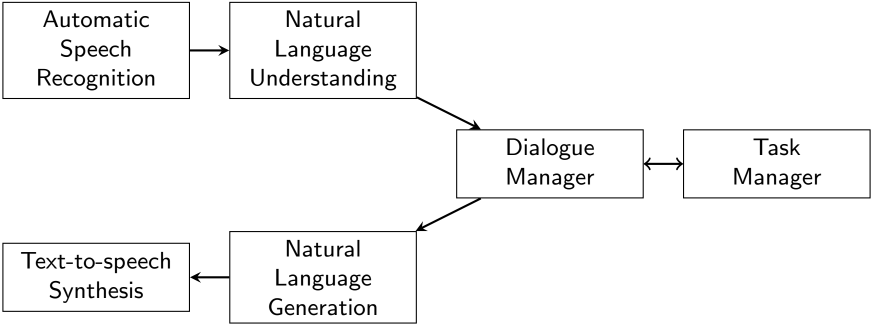
Task-based chatbots

- Back to task-based dialogue:
- Adoption of generation as completion of prefixes, where prefixes are dialogue-state representations



Source: Budzianowski & Vulić, WNGT 2019, *Hello, It's GPT-2 – How Can I Help You?* Towards the Use of Pretrained Language Models for Task-Oriented Dialogue Systems

Task-based chatbots



- Natural language input: joint dialogue management & NLU embeddings (belief state, database state, context)
- System & User tokens
  $\rightarrow$ learn a system response

Part 3: Concerns about NLG

Concerns about NLG

- "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?" Bender et al, FAccT 2021
- encoded bias, changing social views, size $\neq$ diversity, risks and harms in practice, environmental & financial cost
- *Training data extraction attacks*: obtaining verbatim sequences from GPT-2's training data, incl personal info (Carlini et al, 2021)
- Lack of disclosure about training (method, code, *data*): GPT-3 bigger and better than GPT-2 by many measures, including reciting radical content (McGuffie & Newhouse 2020)
- "Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data", Bender & Koller, ACL 2020: "we argue that a system trained only on form has *a priori* no way to learn meaning"
- No common-sense, lack of consistency, little real-world grounding, pragmatics, appropriateness – unlike human learners

Concerns about NLG

- Large MLMs led to performance improvements but: Frustratingly Simple Pretraining Alternatives to Masked Language Modeling; Does Pretraining for Summarization Require Knowledge Transfer?
- Could try to generate from deeper meaning representations: e.g. 2020-21 NLP/L90 lecture by Weiwei Sun on summarisation from AMR (Hardy & Vlachos, EMNLP 2018)
- Ehud Reiter's blog: many posts about NLG, including 'Academic NLG should not fixate on end-to-end neural' – points to potential of modular pipeline ML architectures (they work well: Castro Ferreira et al, EMNLP 2019)
- Ethics of NLG 'in the wild': long-held concerns, e.g. ELIZA; industry-led training, guardian role for researchers?
- *Reminder of the positives*: question-answering, fact-checking, home assistance, language learning, summarisation and knowledge distillation, chit-chat, customer service, clinical/personal settings, *and more*

Part 4: Other NLG tasks

A very brief summary

- Machine translation: a rich history, J&M 3rd edn ch.10
- Abstractive summarisation: Weiwei's lecture 12 last year, on Moodle
- Creative writing, e.g. a poetry assignment: 'Write a poem from the point of view of a cloud looking down on two warring cities' –

  I think I'll start to rain,
  Because I don't think I can stand the pain,
  Of seeing you two,
  Fighting like you do. (By GPT-3, reported in Nature)

- Image captioning

The man at bat readies to swing at the pitch while the umpire looks on.

- n.b. Speech not only text (e.g. for assistive tech in the home)

Further reading

- Lecture notes 19/20, lecture 11:
  https://www.cl.cam.ac.uk/teaching/2122/NLP/materials.html
- Chapters 10 'Machine Translation', 23 'Question Answering', 24 'Chatbots & Dialogue Systems' by Jurafsky & Martin *Speech and Language Processing* 3rd edition (in prep).
  https://web.stanford.edu/~jurafsky/slp3/
- Weizenbaum, 1966, 'ELIZA', in *Communications of the ACM*
- ReproGen: Shared Task on Reproducibility of Human Evaluations in NLG, INLG 2021
- 'Ghost Fictions' by Cambridge Digital Humanities