Dialogue – the final frontier

- Human-like dialogue with a machine was literally the \textit{first task} proposed in the field of artificial intelligence.
- It remains the most elusive.

- To succeed, our agents must:
  1. Understand the world and task, \textit{and}
  2. Respond realistically and consistently.
Personal assistants

“Hey Siri”  2011
“Hey Cortana”  2014
“Alexa”  2014
“OK Google”  2016
“Hi Bixby”  2017
# Web apps vs. Dialogue Agents

<table>
<thead>
<tr>
<th></th>
<th>Web apps</th>
<th>Dialogue agents</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Situation</strong></td>
<td>Browsing, rarely a specific goal</td>
<td>Searching, with specific goal</td>
</tr>
<tr>
<td><strong>Display</strong></td>
<td>Structured</td>
<td>Semi-structured</td>
</tr>
<tr>
<td><strong>Interface</strong></td>
<td>Click/touch</td>
<td>Language</td>
</tr>
<tr>
<td><strong>Easiness to learn</strong></td>
<td>Some trial &amp; error</td>
<td>No need to learn</td>
</tr>
<tr>
<td><strong>Flexibility</strong></td>
<td>Low, usually deterministic</td>
<td>High (one day)</td>
</tr>
</tbody>
</table>
Building blocks of a dialogue agent

Speech Recognition (ASR) → “What is the closest restaurant?” → Language Understanding (NLU) → Given the current location, I need to (1) find the closest restaurants, (2) display a list, and (3) give an audio response.

Text: “Here are the restaurants I found.” → Language Generation (NLG) → Information Retrieval (IR) → [{“name”: “Subway”, “location”: “195 College St”}, {“name”: “Spicy Mafia”, “location”: “181 College St”}, ]

Optional: renders a clickable list on the phone screen.

• This illustrates one “turn” of dialogue – in multi-turn dialogues, we also need Dialogue State Tracking (DST).
• ASR, NLU, IR, and synthesis are all important components that we’ve already discussed. In this module, we will go over the remaining two components: NLG, DST.
Overview

Each building block is a relatively well-defined NLP task:
• Task setting.
• Approaches to address this task.
• Recent developments & debates about the task.
NATURAL LANGUAGE GENERATION
Generate coherent responses in human language
### Fill the Slots

- **SHOW** → show me | i want | can i see | ...
- **DEPART_TIME_RANGE** → (after | around | before) HOUR | morning | afternoon | evening
- **HOUR** → one | two | three | four... | twelve (AMPM)
- **FLIGHTS** → (a) flight | flights
- **AMPM** → am | pm
- **ORIGIN** → from CITY
- **DESTINATION** → to CITY
- **CITY** → Boston | San Francisco | Denver | Washington

That’s not very scalable, is it?

---

Sequence-to-sequence

• Generating a response can be considered as “translation”.
• Sequence-to-sequence methods in translation may apply as well.
  • E.g., including beam search
End-to-end translation dialogue systems

Extensions exist that add variational encoding or diversity-promoting objective functions to avoid Siri-like repetitiveness.

End-to-end dialogue systems

- **Claim:** “we view our model as a cognitive system, which has to carry out natural language understanding, reasoning, decision making, (sic) and natural language generation”.

- **Objective:** Perplexity (where \( U \) is an utterance)...

\[
\exp \left( - \frac{1}{N_w} \sum_{n=1}^{N} \log P_{\theta} (U_1^n, U_2^n, U_3^n) \right)
\]


- **Overhype** vb. make exaggerated claims about (a product, idea, or event); publicize or promote excessively

CSC401/2511 – Spring 2022
Language degeneration problem

- NLG models like to repeat themselves.
- Here’s an example from GPT-2:

```python
prompt = 'I sometimes get bored.'
input_ids = tokenizer.encode(prompt, return_tensors='pt')
greedy_output = model.generate(input_ids, max_length=50)  # Greedy search
tokenizer.decode(greedy_output[0], skip_special_tokens=True)

'I sometimes get bored. I'm not sure if I'm going to be able to do it. I'm not sure if I'm going to be able to do it. I'm not sure if I'm going to be able to do it.'

beam_output = model.generate(
    input_ids, max_length=50, num_beams=5, early_stopping=True)  # Beam search
tokenizer.decode(beam_output[0], skip_special_tokens=True)

'I sometimes get bored. I don't know what to do. I don't know what to do. I don't know what to do. I don't know what to do. I don't know what to do.'
```
Avoid repeating n-grams

• This is hacky, but it works!

```python
beam_output = model.generate(
    input_ids, max_length=50, num_beams=5,
    no_repeat_ngram_size=2, early_stopping=True)  # Beam search + no repeat
tokenizer.decode(beam_output[0], skip_special_tokens=True)

'I sometimes get bored. I don't know what to do about it.\n\n"I'm not sure if I'm going to go back to school or not, but I think it's time for me to get back on my feet."

beam_output = model.generate(
    input_ids, max_length=50, num_beams=5,
    no_repeat_ngram_size=3, early_stopping=True)  # Beam search + no repeat
tokenizer.decode(beam_output[0], skip_special_tokens=True)

'I sometimes get bored. I don't know what to do with myself.\n\n"I don't want to go to the gym. I want to do something else."\n\nHe added: "I'm not going to do anything else."
```
Why do these models repeat?

• An intuition: NLG models have limited memories.
  • What happened, e.g., >3 steps away is forgotten.
  • The last 3 steps only provide a finite number of “patterns”.
  • Therefore, once entering a cycle in NLG, it is hard to get out.

• How to address this problem?
  • Use larger hidden vectors, attention connections, specially-designed network structures, etc.
  • Introduce some randomness.
Let’s get random by sampling

- Randomly sample token according to the distribution of tokens.
  \[ x_t \sim P(x_t = w | x_{1..(t-1)}) \]

Image source: Antoine Bosselut’s tutorial at ACL 2020 “Decoding from Neural Text Generation Models” Link
Scale the temperature

- The distribution might get too random...
- A solution: tune the temperature in softmax.

\[
P(x_t = w | x_{1..(t-1)}) = \frac{e^{o_n}}{\sum_{m=1}^{M} e^{o_m}}
\]

\[
P(x_t = w | x_{1..(t-1)}) = \frac{e^{o_n/\tau}}{\sum_{m=1}^{M} e^{o_m/\tau}}
\]

\(\tau = 2\)  \(\tau = 1\)  \(\tau = 0.5\)
Sample from only top $k$

- We don’t need *all* tokens in the vocabulary in this step.
- Those with small probabilities should have no chance at all; only consider top $k$ candidates.

Image source: Antoine Bosselut’s tutorial at ACL 2020 “Decoding from Neural Text Generation Models” [Link](#)
Top-\(p\) ("nucleus") sampling

- Sample from subset of vocabulary ("nucleus"), where probability mass is concentrated
- Sample from those candidates with \(p > p_*\)
  - where \(p_*\) is a hyper-parameter.

Generation vs copying

• Sometimes, we want to **quote** from the input as part of a response.
• Sometimes, we want to **synthesize** the response.
• Pointer-Generator: let seq2seq models **learn to choose** from the two modes!
  • Compute a probability $p_{gen}$ based on the decoder state and decoder inputs. Overall:
    $$P(w) = p_{gen} P_{generate}(w) + (1 - p_{gen}) P_{copy}(w)$$
  • $P_{generate}$ follows seq2seq computation.
  • $P_{copy} \approx$ frequency of $w$ in source document.

Evaluation criteria

• Some metrics to quantify the quality of generated sentences:
  • If you have a target sentence:
    • N-gram overlap: **BLEU, ROUGE, METEOR, ...**
    • Distance based: **Levenshtein, ...**
  • To measure the diversity:
    • **Self-BLEU**: repetitiveness with oneself.
    • **Type-token ratio** (TTR): vocabulary richness.
  • There are many other evaluation criteria!

Type-Token Ratio (TTR)

- \[ TTR = \frac{N.\text{unique tokens}}{N.\text{tokens}} \]
- More repetition -> lower TTR
- TTR measures the lexical richness.

"what are thoughts well what are thoughts is a good question"

Number of types (unique words) = 8
Number of total words = 11
Type-token ratio = 8 / 11 = 72.7%
PUTTING THE PIECES TOGETHER

Putting it together, for responding realistically and consistently
Dialogue Acts

• Everything in a discourse is a kind of action being performed by the speaker or writer.
• In speech, these are referred to as speech acts.
• In dialogue, these are referred to as dialogue acts.
• Dialogue is more complicated than mere speech, e.g.:
  • The hearer can ground on the speaker’s utterances (i.e., acknowledge, and make it clear that the speaker understands).
  • There are some actions to correct the misunderstandings of the other speaker.
Dialogue Acts

• Here is a hypothetical conversation between a human and a smart assistant. Each utterance is labeled by a dialogue act:

“Where is the closest Subway?” [seek_information]
“The closest subway is the Queen’s Park subway station.” [information]
“No. I mean the Subway restaurant.” [correction]
“I see. Here is the information about this Subway.” [acknowledgement + information]
“Can you make an order?” [action_instruction]
“Ok.” [agree]
Dialogue Acts Classification

• Detecting the dialogue act is a popular NLP task.
• This is usually handled as a sequential tagging task.
• A typical dataset is Switchboard Dialogue Act (SwDA) Corpus.
• Here are some example annotations:

<table>
<thead>
<tr>
<th>Name</th>
<th>Tag</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statement-non-opinion</td>
<td>sd</td>
<td>Me, I’m in the legal department.</td>
</tr>
<tr>
<td>Statement-opinion</td>
<td>sv</td>
<td>I think it’s great.</td>
</tr>
<tr>
<td>Agree / Accept</td>
<td>aa</td>
<td>That’s exactly it.</td>
</tr>
<tr>
<td>Conventional-closing</td>
<td>fc</td>
<td>Well, it’s been nice talking to you.</td>
</tr>
</tbody>
</table>

The Switchboard Dialog Act Corpus website: [http://compprag.christopherpotts.net/swda.html](http://compprag.christopherpotts.net/swda.html)
Let me Bing that for you

(a) 2014

(b)

Let me actually answer that for you

What (might have) happened?

Chatbots should track the states

• When interacting with chatbots, there can be multiple turns.
• Dialogue responses should consider both the context and the inquiry.

“Where is the closest Subway?”
“The closest subway is the Queen’s Park subway station.”
“No. I mean the Subway restaurant.”
“I see. Here is the information about this Subway.”
“Can you make an order?”
(A) “Where do you want to make this order?”
(B) “Ok. What would you like to have?”
States of (dis-)belief

- Map utterances to dialogue acts and beliefs about the world.
- Maintain (and update*) those beliefs.

* Humans can barely do this.


https://dialogflow.com/docs/intro
### Core dialog acts

<table>
<thead>
<tr>
<th>Core dialog acts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Info-request</td>
<td>Speaker wants information from addressee</td>
</tr>
<tr>
<td>Action-request</td>
<td>Speaker wants addressee to perform an action</td>
</tr>
<tr>
<td>Yes-answer</td>
<td>Affirmative answer</td>
</tr>
<tr>
<td>No-answer</td>
<td>Negative answer</td>
</tr>
<tr>
<td>Answer</td>
<td>Other kinds of answer</td>
</tr>
<tr>
<td>Offer</td>
<td>Speaker offers or commits to perform an action</td>
</tr>
<tr>
<td>ReportOnAction</td>
<td>Speaker notifies an action is being/has been performed</td>
</tr>
<tr>
<td>Inform</td>
<td>Speaker provides addressee with information not explicitly required (via an Info-request)</td>
</tr>
</tbody>
</table>

### Conventional dialog acts

<table>
<thead>
<tr>
<th>Conventional dialog acts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greet</td>
<td>Conversation opening</td>
</tr>
<tr>
<td>Quit</td>
<td>Conversation closing</td>
</tr>
<tr>
<td>Apology</td>
<td>Apology</td>
</tr>
<tr>
<td>Thank</td>
<td>Thanking (and down-playing)</td>
</tr>
</tbody>
</table>

### Feedback/turn management dialog acts

<table>
<thead>
<tr>
<th>Feedback/turn management dialog acts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clarif-request</td>
<td>Speaker asks addressee for confirmation/repetition of previous utterance for clarification.</td>
</tr>
<tr>
<td>Ack</td>
<td>Speaker expresses agreement with previous utterance, or provides feedback to signal understanding of what the addressee said</td>
</tr>
<tr>
<td>Filler</td>
<td>Utterance whose main goal is to manage conversational time (i.e. speaker taking time while keeping the turn)</td>
</tr>
</tbody>
</table>

### Non-interpretable/non-classifiable dialog acts

<table>
<thead>
<tr>
<th>Non-interpretable/non-classifiable dialog acts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Other</td>
<td>Default tag for non-interpretable and non-classifiable utterances</td>
</tr>
</tbody>
</table>
State of (dis-)belief

- Use reinforcement learning to make these explicit.

CSC401/2511 – Spring 2022

Belief, $b_t$: intent(open, podBay.doors)

Observation, $o_t$

Action, $a_t$: I’m afraid I can’t do that.

Very negative reward $r_k$ associated with the door being open

Policy $\pi(b) = a$

Return $R_t = \sum_{k=t}^{T} \gamma^k r_k$

Value $V^\pi(b) = E[R_t | b_t = b]$

Q $Q^\pi(b, a) = E[R_t | b_t = b, a_t = a]$
Aside – RL in dialogue

Aside – RL in dialogue

Aside – RL in dialogue

• Challenge 1: data is limited in a particular domain
  Solution 1: learn a distributed architecture with Gaussian priors

• Challenge 2: Estimates of $Q$ aren’t shared across different domains
  Solution 2: Use a Bayesian ‘committee machine’

Aside – RL in dialogue

- ACER learns an ‘off policy’ gradient $\nabla J$ and modified loss $\nabla L$.
- Avoid bias through replaying experience

The off-policy version of the Policy Gradient Theorem [30] is used to derive the gradients $\nabla_\omega J(\omega) \approx g(\omega)$:

$$g(\omega) = \sum_{b \in B} d^\mu(b) \sum_{a \in A} \nabla_\omega \pi(a|b) Q_\pi(b, a)$$ (1)

$$\nabla L(\theta) = \nabla_\theta (Q^{ret} - Q_\theta(b, a))^2$$

$$Q^{ret} = Q(b, a) + \mathbb{E}_\mu \left[ \sum_{t \geq 0} \gamma^t \left( \prod_{s=1}^t \lambda \min(1, \rho(a_s|b_s)) \right) (r_t + \gamma V(b_{t+1}) - Q(b_t, a_t)) \right]$$

From Milica Gašić, Cambridge

PyDial toolkit

- **PyDial** (pydial.org) is an open-source Python toolkit for building dialogue systems. PyDial has 3 key components:
  - **Agent** module.
  - **User Simulation** module
    - used in, e.g., RL-based algorithms
  - **Evaluation** module.

Corpora for dialogue

<table>
<thead>
<tr>
<th>Metric</th>
<th>DSTC2</th>
<th>SFX</th>
<th>WOZ2.0</th>
<th>FRAMES</th>
<th>KVRET</th>
<th>M2M</th>
<th>MultiWOZ</th>
</tr>
</thead>
<tbody>
<tr>
<td># Dialogues</td>
<td>1,612</td>
<td>1,006</td>
<td>600</td>
<td>1,369</td>
<td>2,425</td>
<td>1,500</td>
<td>8,438</td>
</tr>
<tr>
<td>Total # turns</td>
<td>23,354</td>
<td>12,396</td>
<td>4,472</td>
<td>19,986</td>
<td>12,732</td>
<td>14,796</td>
<td>115,424</td>
</tr>
<tr>
<td>Total # tokens</td>
<td>199,431</td>
<td>108,975</td>
<td>50,264</td>
<td>251,867</td>
<td>102,077</td>
<td>121,977</td>
<td>1,520,970</td>
</tr>
<tr>
<td>Avg. turns per dialogue</td>
<td>14.49</td>
<td>12.32</td>
<td>7.45</td>
<td>14.60</td>
<td>5.25</td>
<td>9.86</td>
<td>13.68</td>
</tr>
<tr>
<td>Avg. tokens per turn</td>
<td>8.54</td>
<td>8.79</td>
<td>11.24</td>
<td>12.60</td>
<td>8.02</td>
<td>8.24</td>
<td>13.18</td>
</tr>
<tr>
<td>Total unique tokens</td>
<td>986</td>
<td>1,473</td>
<td>2,142</td>
<td>12,043</td>
<td>2,842</td>
<td>1,008</td>
<td>24,071</td>
</tr>
<tr>
<td># Slots</td>
<td>8</td>
<td>14</td>
<td>4</td>
<td>61</td>
<td>13</td>
<td>14</td>
<td>25</td>
</tr>
<tr>
<td># Values</td>
<td>212</td>
<td>1847</td>
<td>99</td>
<td>3871</td>
<td>1363</td>
<td>138</td>
<td>4510</td>
</tr>
</tbody>
</table>

Table 1: Comparison of our corpus to similar data sets. Numbers in bold indicate best value for the respective metric. The numbers are provided for the training part of data except for FRAMES data-set were such division was not defined.

- **Ubuntu dialogue corpus** and **AMI Meeting corpus** are also popular.

The DSTC challenges

• DSTC challenge is held (almost) annually since 2012.
  • DSTC 1-6: “Dialogue State Tracking Challenges”
  • DSTC 7-present: “Dialogue Systems Technology Challenges”
• What “dialogue state” exactly means depends on the problem settings. For example:
  • In DSTC1 (dialogue about bus timetable information): The dialogue system infers the bus that the user wants to take.
  • In DSTC2 (dialogue about restaurant search): The dialogue state includes the slot/value attributes of the user goal, their search method, etc.
Recent DSTC challenges

DSTC10 @ AAAI-22 contains some more complex challenges. Here are the 3 tasks in Track 1 (Internet meme and open-domain dialogue)

1. Text response modeling. Generate coherent and natural text response based on chat history containing memes and texts.
2. Meme Retrieval. Based on the multimodal chat history, select a suitable meme to respond.
3. Meme Emotion Classification. Based on the multimodal chat history, predict the user’s sentiment.
Recent DSTC challenges

Here are the 2 tasks in DSTC10 Track 2:

1. **Dialogue State Tracking.** Fill each slot with the estimated string.

2. **Conversation modeling** given knowledge access:
   - 2.1: Binary classification. Decide if continue the dialogue or trigger the “knowledge access”
   - 2.2: Select from knowledge sources.
   - 2.3: Given the conversation context and the “knowledge snippet”, generate a system response.
EVALUATING DIALOGUE AGENTS
Lessons from HitchBot?

People (sometimes) like cute things that are smaller than they are.
Participant evaluation

• Human chats with model for 6 turns and rates 8 dimensions:
  • Avoiding repetition
  • Interestingness
  • Making sense
  • Fluency
  • Listening
  • Inquisitiveness
  • Humanness
  • Engagingness

“How much did you enjoy talking to this user?”

“How often did this user say something which didn’t make sense?”

Observer evaluation

Annotators look at two conversations and decide which is better, in terms of:

• **Engaging**: who would you prefer to talk to for a long conversation?
• **Interesting**
• **Humanness**
• **Knowledgeable**: If you had to say that one speaker is more knowledgeable and one is more ignorant, who is more knowledgeable?
Evaluation for task-based systems

• Task-based systems are evaluated by task success.
• For a slot machine: we can also use slot error rate.

“Make an appointment with Frank at 10:30 in LM 162”

<table>
<thead>
<tr>
<th>Slot</th>
<th>Filler</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person</td>
<td>Frank</td>
</tr>
<tr>
<td>Time</td>
<td>11:30 AM</td>
</tr>
<tr>
<td>Location</td>
<td>LM 162</td>
</tr>
</tbody>
</table>

Slot error rate: 1/3

Task success: In the end, was the correct meeting added to the calendar?
Evaluation for task-based systems

• A more fine-grained method to evaluate task-based dialogue systems is user satisfaction survey.

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTS Performance</td>
<td>Was the system easy to understand?</td>
</tr>
<tr>
<td>ASR Performance</td>
<td>Did the system understand what you said?</td>
</tr>
<tr>
<td>Task Ease</td>
<td>Was it easy to find the message/flight/train you wanted?</td>
</tr>
<tr>
<td>Interaction Pace</td>
<td>Was the pace of interaction appropriate?</td>
</tr>
<tr>
<td>User Expertise</td>
<td>Did you know what you could say at each point?</td>
</tr>
<tr>
<td>System response</td>
<td>How often was the system slow in replying you?</td>
</tr>
<tr>
<td>Expected Behavior</td>
<td>Did the system work the way you expected it to?</td>
</tr>
<tr>
<td>Future Use</td>
<td>Do you want to use the system in the future?</td>
</tr>
</tbody>
</table>

Automatic evaluation

Automatic evaluation methods (e.g., BLEU) are usually *not* used to evaluate chatbots.

- They correlate poorly with human judgements.
- There are multiple correct ways for a dialogue to proceed, but BLEU just “checks the ground truth”.

One current research direction is **adversarial evaluation**.

- This is inspired by the Turing Test.
- Train a “Turing-like” classifier to distinguish human responses vs. machine responses.
- Successful dialogue systems are good at fooling the evaluator.
Evaluating end-to-end dialogue

<table>
<thead>
<tr>
<th></th>
<th>Ubuntu Dialogue Corpus</th>
<th>Twitter Corpus</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Embedding Averaging</td>
<td></td>
</tr>
<tr>
<td>R-TFIDF</td>
<td>0.536 ± 0.003</td>
<td>0.483 ± 0.002</td>
</tr>
<tr>
<td>C-TFIDF</td>
<td>0.571 ± 0.003</td>
<td>0.531 ± 0.002</td>
</tr>
<tr>
<td>DE</td>
<td><strong>0.650 ± 0.003</strong></td>
<td><strong>0.597 ± 0.002</strong></td>
</tr>
<tr>
<td>LSTM</td>
<td>0.130 ± 0.003</td>
<td>0.593 ± 0.002</td>
</tr>
<tr>
<td>HRED</td>
<td>0.580 ± 0.003</td>
<td>0.599 ± 0.002</td>
</tr>
<tr>
<td></td>
<td>Greedy Matching</td>
<td></td>
</tr>
<tr>
<td>R-TFIDF</td>
<td>0.370 ± 0.002</td>
<td>0.356 ± 0.001</td>
</tr>
<tr>
<td>C-TFIDF</td>
<td>0.373 ± 0.002</td>
<td>0.362 ± 0.001</td>
</tr>
<tr>
<td>DE</td>
<td>0.413 ± 0.002</td>
<td>0.384 ± 0.001</td>
</tr>
<tr>
<td>LSTM</td>
<td>0.097 ± 0.003</td>
<td>0.439 ± 0.002</td>
</tr>
<tr>
<td>HRED</td>
<td><strong>0.418 ± 0.003</strong></td>
<td><strong>0.439 ± 0.002</strong></td>
</tr>
<tr>
<td></td>
<td>Vector Extrema</td>
<td></td>
</tr>
<tr>
<td>R-TFIDF</td>
<td>0.342 ± 0.002</td>
<td>0.353 ± 0.001</td>
</tr>
<tr>
<td>C-TFIDF</td>
<td>0.353 ± 0.002</td>
<td>0.353 ± 0.001</td>
</tr>
<tr>
<td>DE</td>
<td><strong>0.376 ± 0.001</strong></td>
<td><strong>0.365 ± 0.001</strong></td>
</tr>
<tr>
<td>LSTM</td>
<td>0.089 ± 0.002</td>
<td></td>
</tr>
<tr>
<td>HRED</td>
<td><strong>0.384 ± 0.002</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Models evaluated using the vector-based evaluation metrics, with 95% confidence intervals.

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).

Security and privacy issues

- DNNs can generate toxic content.
  - Apparently, GPT3 etc., have ‘radicalized’ knowledge about extremist, racism, etc.
  - The knowledge might come from the training data.
- How to avoid them?
  - Filtering does not solve everything.
  - We need to be careful about the training data!
  - Domain-specific experts are needed.

Dialogue summary

- Discourse & pragmatics provide huge research potential.
  - (i.e., many problems are unsolved)
- Developing dialogue technology requires integrating almost all NLP technologies.
- Dialogue technology also has wide application potential.