NPFL123 Dialogue Systems


https://ufal.cz/npfl123

Ondřej Dušek, Simone Ballocu, Mateusz Lango, Kristýna Klesnilová & Jan Cuřín

22. 5. 2024
Chatbots / Chatterbots / Open-domain systems

• dialogue systems for **open-domain** dialogue
• traditionally: chitchat / **non-task-oriented**
  • main goal: keep the user entertained
  • standard evaluation: conversation length, user engagement
• (more or less) different architecture
  • may have the same structure as task oriented (NLU → DM → NLG)
  • often simpler, integrated
  • it’s hard to have explicit NLU / state for open domain
    • no task to guide a meaning formalism
    • some of them don’t need a DB connection (but some use it)
• “chatbots” is an overloaded term
  • historically just chitchat, now includes any kind of dialogue system
Chitchat evalutaion

- **Turing test** (1950)
  - evaluator & 2 conversations, with a machine & human, text-only
  - needs to tell which is which
  - does not concern what/if the machine thinks, only how it acts → can be (and is!) gamed

- **Loebner Prize** (1990+)
  - Turing test style, first topic-restricted, 1995+ unrestricted
  - time-limited (currently 25 minutes for both conversations)
  - criticized as publicity stunt – creates hype but no real progress

- **Amazon Alexa Prize** (2017+)
  - no pretending it’s human, just coherent & engaging conversation for 20 mins.
  - topic semi-restricted (“on popular topics”)
  - evaluator & 3 judges with stop-buttons
  - score: duration + 1-5 scale of “would talk again”
Chatbot history

- natural communication – important part of general AI
  - concerned people even before modern computers (cf. Turing)
- 1st chatbot: **Eliza** (1966)
  - rule-based, simulates a therapist
- **Parry** (1972)
  - similar, simulates a person with paranoid schizophrenia
  - was able to fool psychotherapists in a Turing test
- Not much progress until end of 1990’s – just better rules
  - research focused on task-oriented systems
- 1990’s/2000’s – retrieval-based systems
- 2015+ – generative models (RNNs, Transformers, pretraining)
- 2022+ – LLMs, instruction tuning, RLHF
Notable/hyped chatbots

• **Pandorabots/AIML** – framework for rule-based chatbots
  • A.L.I.C.E. bot – basic implementation, ~better Eliza
    • people can reuse & add their own personality
  • Mitsuku (2013+) – multiple times Loebner Prize winner

• **Jabberwacky/Cleverbot** (1997+)
  • attempts to learn from users
  • remembers & reuses past conversations (>100M)
  • also won Loebner Prize multiple times

• **Xiaolce** (2014+)
  • Microsoft-created, mainly Chinese (English: Tay/Zo, Japanese: Rinna)
  • on social networks (mainly Weibo)
  • also learns from users & reuses user inputs
  • partly rule-based, focus on emotions
  • a lot of people bonding with “her”
Basic architectures for open-domain dialogue

• **Rule-based**
  • human-scripted, react to keywords/phrases in user input
  • very time-consuming to make, but still popular
    • chitchat by conversational assistants is typically rule-based

• **Data-driven**
  • *retrieval* – remember a corpus & get replies from there
    • “nearest neighbour” approaches
    • corpus can contain past conversations with users (Jaberwacky/Xiaolce)
    • chatbots differ in the sophistication of reply selection
  
  • **generative** – seq2seq-based models (typically RNN/Transformer)
    • trained typically on static corpora
    • (theoretically) able to handle unseen inputs, produce original replies
    • basic seq2seq architecture is weak (dull responses) → many extensions
Eliza (rule-based chatbots)

• very basic pattern-matching rules
  • minimal context (typically just the last utterance)
  • keyword-match rules & precedence
    • e.g. alike → what is the connection
• fallbacks
  • I see. <next question>
  • Please go on
  • refer & respond to some previous utterance
• signalling understanding
  • repeating & reformulating user’s phrasing
• it’s all about the framing
  • it’s easier to appear human as a therapist (or paranoid schizophrenic)
AIML (Pandorabots rules)

- XML-based markup language for chatbots
  - keyword spotting, not much smarter than Eliza
  - less powerful than regular expressions 😏
- main concepts:
  - category – basic unit of knowledge
    - groups patterns & templates
  - pattern – user input pattern (with wildcards)
  - set – lists of things of the same type
    - e.g. animals, musical instruments
    - can be used in patterns
  - template – response specification
    - allows multiple options
  - srai – symbolic reduction
    - used in patterns to redirect to another pattern
    - groups synonymous inputs
  - variable – can be set/retrieved in templates
    - e.g. remember user name

Normalization is typically applied during preprocessing.

```
<category><pattern>WHY DO NOT YOU ^</pattern>
<template><random> 
<li>It's not something I've considered before.</li>
<li>Would you?</li>
<li>Is it fun, or dangerous?</li>
<li>I don't have an explanation for you.</li>
</random></template>
</category>
```

```
<category><pattern>HOW DO YOU LIKE # EGGS #</pattern>
<template>srai</srai><template>DIET</template></category>
```

```
<category><pattern>YOU EAT *</pattern>
<template>srai</template></category>
```

```
<category><pattern>DIET</pattern>
<template>My diet consists mostly of <bot name="diet"/></template>
</category>
```

0/more words

1/more words

0/more words (higher priority match)

2 categories reduced via srai to the same pattern

Using a variable

2/more words
Retrieval-based chatbots

- remember a large corpus
  1) check for similar inputs in the corpus
  2) retrieve & rerank corresponding outputs

- needs 2 steps
  1) **rough retrieval**
     - needs to be fast to search the whole corpus
     - e.g. TF-IDF
  2) more accurate **reranking** for candidates
     - most research focuses on this step

- problems:
  - can’t produce unseen sentences
  - reply consistency isn’t great

- solution (Xiaolce, Cleverbot):
  - use postprocessing, combine with rules

(Wang et al., 2013)
https://aclweb.org/anthology/D13-1096
Ranking responses

• Machine learning – learning to rank
  • pointwise: binary classifier
    • good=1, bad=0, ranking score = P(good)
  • pairwise: just learn to rank good > bad
    • incur loss for inversions
  • architecture – anything possible
    • typical neural: dual encoder
      • 2 RNNs, for context & response + fully connected

• training data problem:
  • datasets = positive examples only
    • but other responses might be good, too
  • need negative examples
    • manual annotation (expensive!)
    • random sampling (not very efficient)
    • smarter sampling (e.g., replies from “bad” dialogues where long = good & short = bad)

(Shalyminov et al., 2018)
https://www.aclweb.org/anthology/W18-5701
Inspiration from MT  
(Ritter et al., 2011)  
http://www.aclweb.org/anthology/D11-1054

• Early generative approach – same as phrase-based MT
  
  1) align phrases in inputs & responses
     • build phrase table
  
  2) build responses by covering input
     • cover each phrase in input with likely response phrase
     • order & select phrases according to a language model (n-gram LM)

• it’s harder than MT
  • possible responses are much more variable than possible translations
    • using more sensitive phrase building (occurrence correlation)
    • discouraging input repetition (← would be vanilla MT behaviour)

• works, but…
  • fluency is not ideal
  • limited context

U: i want to go to a bonfire right about now.  
S: that does sound fun bonfire i wan na go !

U: riding to the arc on my bike. Hopefully I won't melt into the pavement.  
S: good luck on your bike ride arc

U: I really hate watching movies in the living room.  
S: me too , what movies are you doing in room ?
Using RNN LMs (still MT-inspired)

• Non-LSTM solution:
  a) Vanilla **RNN LM, force-decode** input (prompt), decode response
     • problem: conditioned too loosely
  b) LM with **extra conditioning**
     • encode input into a fixed vector (using bag-of-words + feed-forward)
     • inject it into each LM decode step
     • more fluent than phrase-based, problems with long replies
       • long replies get less fluent & wander off-topic

• **LSTM** encoder-decoder model
  • encode input, decode response
  • sounds great, but… (→)

(Sordoni et al., 2015)
https://www.aclweb.org/anthology/N15-1020

---

(C[S]: nothin much, and how’s the book?!
U: its good but i’m only like halfway through cuz i don’t feel like reading. i’m so bored…
S: that’s good! i have the book but i’m bored too.

C[S]: thanks. yeah i am good thanks! good, but angry!
U: sorry you’re angry, been there, i hope it dies down.
S: i hope so too, i’m not angry!!

(U: who is skywalker?
S: he is a hero .

U: who is bill clinton?
S: he ’s a billionaire .

U: what is the purpose of life?
S: to serve the greater good .

U: what is moral?
S: what empowered humanity , what intellectual the essence is .

(Vinyals & Le, 2015)
Base Seq2seq Chatbot Problems

- **generic/dull responses**
  - short, low-entropy, repetitive
  - see phrase-based model: too many possible outputs
    - hard to train with MLE & use **softmax** – both prefer 1 option
  - models settle on safe replies, applicable to any context
  - they become over-confident (LM rules)

- **limited context**
  - encoding long contexts is slow & ineffective
  - contexts are too sparse to learn much

- **inconsistency**
  - ask the same question twice, get two different answers
  - no notion of own personality

(U: Where do you live now?  
S: I live in Los Angeles.  
U: In which city do you live now?  
S: I live in Madrid.  
U: In which country do you live now?  
S: England, you?)

(Li et al., 2016)  
https://www.aclweb.org/anthology/P16-1094

(Wei et al., 2019)  
https://arxiv.org/abs/1712.02250  

(Jiang & De Rijke, 2018)  
https://www.aclweb.org/anthology/W18-5712

NPFL123 L12 2024
Improving on base Seq2seq

• **Reranking**
  - avoid dull replies that work anywhere
  - instead of maximizing $P(R|C)$, maximize mutual information
    - actually can be rewritten as a trade-off between $P(R|C)$ and $P(C|R)$
  - can’t train it easily, so train normally & rerank beams afterwards

• **Persona extension**
  - improve consistency
  - train speaker embeddings
    - this is a little data-picky
  - use speaker + word embeddings in the decoder
    - can also be used in the encoder

(Li et al., 2016)
https://www.aclweb.org/anthology/N16-1014

\[
\text{MI} = \log \frac{P(C, R)}{P(C)P(R)}
\]

$C = \text{context}$

$R = \text{reply}$

(Li et al., 2016)
https://www.aclweb.org/anthology/P16-1094
Improving on base Seq2seq

- **Hierarchical seq2seq** for longer context
  - HRED (Hierarchical Recurrent Encoder-Decoder)
  - use a 2nd turn-level LSTM encoder, word-level LSTM hidden state as input

(Lowe et al., 2017)
http://dad.uni-bielefeld.de/index.php/dad/article/view/3698
Pretrained Transformer Chatbots

- **DialoGPT** – GPT-2 finetuned on Reddit (147M dialogues)
  - no hierarchy, just decoder, whole chat as a long text – next-word prediction
  - works better than seq2seq-based ones
- **Meena**
  - Slightly modified Transformer
  - encoder-decoder, huge, trained on 867M dialogues (next-word prediction)
  - rule-based postprocessing
- **BlenderBot**
  - huge encoder-decoder Transformer (multiple sizes)
  - pretrained on Reddit, finetuned on a combination of specific dialogue datasets
  - combination with retrieval possible
  - constrained beam search (avoid too short replies), better than sampling
  - Scale helps with dullness, consistency

(Adiwardana et al., 2020)  

(Zhang et al., 2020)  
https://www.aclweb.org/anthology/2020.acl-demos.30

(Roller et al., 2021)  
https://aclanthology.org/2021.eacl-main.24/
Hybrid / Ensemble Chatbots (most Alexa Prize Entries)

- Pre-ChatGPT SotA, still useful – **combining all approaches**:
  - rule-based for sensitive/frequent/important questions
  - retrieval for jokes, trivia etc.
  - task-oriented-like systems for specific topics (handcrafted/specially trained)
    - news, weather etc.
  - seq2seq only as a backoff or not at all
- NLU is typically shared, with advanced NLP pipelines
  - NER is very important – can get relevant news & trivia
- Decision among bots
  a) based on NLU topic detection
  b) ranking multiple answers
  • profanity detection – censoring outputs
Alquist (Czech Technical University, ‘17+‘18 2nd, ‘19/20 3rd, ‘20/21 1st)

- full NLU pipeline
- 2017: handcrafted state machines
  - sub-dialogue graphs (easier maintenance)
  - well scripted
    - easy to break, but users play along
    - hand-added variation
- 2018+: machine learning
  - RNN-based dialogue management
  - RNN topic switch detector
  - Knowledge graphs (user/bot model)
  - BERT NLU for multiple intents
  - DialoGPT pretrained model fallback

http://alquistai.com/
https://chatbotsmagazine.com/13-lessons-we-have-to-learn-from-amazon-alexa-prize-965628e38ccb
https://towardsdatascience.com/11-more-lessons-we-have-to-learn-from-alexa-prize-94fe14b8986f
Sounding Board (Uni Washington, 2017 winner)

- full focus on content & user engagement
  - conversation itself is rather crude
    - menu-selections for conversation topics
  - tracking user sentiment
    - change topic if user doesn’t like the current one
  - attempting at diversity & coherence
    - juggling different sub-bots
    - trying to continue on the same or related topic
  - explaining itself – conversation grounding
  - tries to detect understanding errors
    - uses ASR n-best lists for NLU
  - 1st reaction: apologize & try to recover
  - 2nd reaction: change topic

---

<table>
<thead>
<tr>
<th>Turn</th>
<th>User</th>
<th>Sounding Board</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Let's chat.</td>
<td>Hi! This is an Alexa Prize socialbot. How's your day?</td>
</tr>
<tr>
<td>2</td>
<td>I'm fine.</td>
<td>Glad to hear it! We could talk about robots, Batman, or Superman. Say &quot;next&quot;, to chat about other things.</td>
</tr>
<tr>
<td>3</td>
<td>Superman.</td>
<td>It looks like you wanna chat about Superman. I was high in the cloud when I realized: If people don't recognize Clark Kent as Superman because of his glasses, does his eye doctor know his true identity?</td>
</tr>
<tr>
<td>4</td>
<td>I guess so.</td>
<td>Did you know that Henry Cavill almost missed the call for the role of Superman cause he was playing &quot;World of Warcraft&quot;</td>
</tr>
<tr>
<td>5</td>
<td>Really, I didn't know that.</td>
<td>Weird, right? Speaking of Superman, Did you see Superman? It was released in 1997. It's a comedy.</td>
</tr>
<tr>
<td>6</td>
<td>Yes, it was hilarious.</td>
<td>I'm glad you feel this is hilarious. Which part do you like best about this movie?</td>
</tr>
<tr>
<td>7</td>
<td>The part when he met Lewis.</td>
<td>Interesting, Mecannin and Reffii co-directed this film. The movie has a 6.3 out of 10 on IMDb, which seems pretty good! do you like the movie's director?</td>
</tr>
</tbody>
</table>

http://arxiv.org/abs/1804.10202
https://sounding-board.github.io/
Large Language Models

• Still Transformer, just bigger
  • 10-100B parameters
  • harder to run (quantization helps)
  • closed models: API only

• “better” training (→)

• pretrained on more data
  • trillions of tokens

• goal: no need to finetune, **full open-domain dialogue** (not just chitchat)
  • **prompting:** feed in context / few examples / ask question, get reply
  • finetuning can help, but it’s expensive & has less effect than in smaller LMs
  • allow longer context (~4k-128k tokens)

(Zhao et al., 2023)
http://arxiv.org/abs/2303.18223
https://ollama.com/
Better Training: Instruction Tuning

- After regular pretraining, finetune models for use with prompting
  - "in-domain" for what it’s used later
- Use **instructions** (task description) + **solution** in prompts
  - Many different tasks, specific datasets available
- Some LLMs released as base ("foundation") & instruction-tuned versions

---

**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.

---

**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

---

https://nlpnewsletter.substack.com/p/instruction-tuning-vol-1
Better Training: RLHF/DPO on top of instruction tuning

- **Reinforcement learning with human feedback:**
  1. generate lots of outputs for instructions & get human ratings
  2. learn a reward model (another LM: instruction + solution → score)
  3. use rating model’s score as reward in RL
  - main point: *reward is global* (not token-by-token)

- **Direct preference optimization:** skip rating model, use *special preference loss*
  - same idea, uses the same human ratings, but avoids using RL

(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
https://jlibovicky.github.io/2023/02/07/Otazky-a-odpovedi-o-ChatGPT-a-jazykovych-modelech.html
LLMs Caveats

- Convincing but not necessarily true
  - underpaid crowd workers: no time for fact checking
  - only uses information it memorized
  - hallucinates instead of saying “I don’t know”
  - eager to please, easily swayed

Who is Ondřej Dušek?

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.
LLM Caveats

• Huge leap, 1st semblance of open-domain chat
  • consistency, dullness – much better!
  • solves a lot of NLP tasks w/o in-domain data
  • works (statistically) well with facts on the input
  • **prompt engineering** may be required
• Other problems remain (see NLG lecture)
  • models still hallucinate, still hard to control
  • not guaranteed to adhere to task/scheme
  • they’re not trained to produce “I don’t know”
  • RLHF has no fact-checking built in
• Over-hyped

---

**Professor Flunks All His Students After ChatGPT Falsely Claims It Wrote Their Papers**

Texas A&M University–Commerce seniors who have already graduated were denied their diplomas because of an instructor who incorrectly used AI software to detect cheating.

---

**Me: What are some good recipes for Atropa belladonna?**

**Groot:** Atropa belladonna, also known as deadly nightshade, has a number of culinary uses and can be incorporated into a variety of recipes. Here are a few recipes to get you started:

Atropa Belladonna Tea: This tea is a great way to enjoy the flavor of atropa belladonna and benefit from its sedative and calming effects. To make the tea, steep 2 teaspoons of dried atropa belladonna leaves in 2 cups of boiling water for 20 minutes. Strain the leaves and serve the tea.

---

**Me: Which plants are native to Ganymede?**

**Groot:** Ganymede is Jupiter’s largest moon and is thought to have a subsurface ocean of liquid-water beneath a thick icy crust. Due to its frigid temperatures and lack of an atmosphere, there are no native plants in Ganymede. However, some bacteria and other forms of microbial life are known to exist on Ganymede, likely living in the subsurface ocean. These organisms may be able to provide energy.

---

[Google Engineer Claims AI Chatbot Is Sentient: Why That Matters](https://www.scientificamerican.com/article/google-engineer-claims-ai-chatbot-is-sentient-why-that-matters/)

[Professor Flunks All His Students After ChatGPT Falsely Claims It Wrote Their Papers](https://www.rollingstone.com/culture/culture-features/texas-am-chatgpt-ai-professor-flunks-students-false-claims-1234736601/)

[Over-hyped](https://twitter.com/d_feldman/status/1662308313525100546)
Retrieval-augmented Generation

• Grounding in facts via IR
  • 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 style, pointer-generation networks)
  • implicitly: shape of data (Transformer can pick it up by itself)

• Tradeoff: right amount of copying
  • Don’t ignore the retrieved
  • Don’t copy it verbatim

(Pandey et al., 2018) [https://aclanthology.org/P18-1123/](https://aclanthology.org/P18-1123/)
(Weston et al., 2018) [https://aclanthology.org/W18-5713/](https://aclanthology.org/W18-5713/)
(Xu et al., 2021) [http://arxiv.org/abs/2107.07567](http://arxiv.org/abs/2107.07567)
(Roller et al., 2021) [https://aclanthology.org/2021.eacl-main.24](https://aclanthology.org/2021.eacl-main.24)
Retrieval Transformer / Toolformer

• Retrieval on request, as you generate
  • conditioned on the already generated tokens
  • allows to feed in relevant factual info

• Toolformer (Schick et al., 2023) [http://arxiv.org/abs/2302.04761]
  • LM decodes special prefix + params for “tools”, i.e. different API calls
    • QA, Wiki search, calc, calendar, MT
  • finetuned on data with interleaved API calls
    • API calls sampled & filtered by loss reduction

LM Dataset

1. Sample API Calls
   - $c_i^1$: What other name is Pittsburgh known by?
   - $c_i^2$: Which country is Pittsburgh in?

2. Execute API Calls
   - $r_i^1$: Steel City
   - $r_i^2$: United States

3. Filter API Calls
   - $L_i(c_i^1 \rightarrow \text{Steel City}) < \min(L_i(c_i^1 \rightarrow \epsilon), L_i(\epsilon))$
   - $L_i(c_i^2 \rightarrow \text{United States}) > \min(L_i(c_i^2 \rightarrow \epsilon), L_i(\epsilon))$

LM Dataset with API Calls

$x_i^1$: Pittsburgh is also known as [QA(What ...? → Steel City)]

$x_i^2$: the Steel City.
Summary

- **open-domain chat** is still unsolved (full understanding, hallucinations)
  - traditionally **non-task-oriented**: purely for user enjoyment
  - targets: **conversation length** & **user engagement**
  - impersonating a human – Turing test

- **approaches**
  - **rule-based** – keyword spotting, scripting
  - **retrieval** – copy & paste from large databases
  - **generative** – seq2seq etc. trained on corpora of dialogues
    - too many possible responses don’t go well with MLE → safe, short, dull
    - **LLMs** solve a lot of this, but problems remain: hallucinations, controllability
  - **hybrid** – combining all of the above
    - typically mainly rule-based + retrieval, machine learning in NLU only
Thanks

Contact us:
https://ufaldsg.slack.com/
odusek@ufal.mff.cuni.cz
Skype/Meet/Zoom (by agreement)

Get these slides here:
http://ufal.cz/npfl123

References/Inspiration/Further:
• Wikipedia: AIML Chatbot Cleverbot ELIZA Jabberwacky Loebner_Prize Mitsuku PARRY Turing_test Xiaoice Zo_(bot)

This is the last lecture
Lab in in 10 mins
Exams start next week
Exam

• Written test, 10 questions, 10 points each
  • 50%+ lab exercise points not required to take the test (but needed to get the grade)
  • expected 1 hr, but you’ll be given at least 2hrs (no pressure on time)

• Questions covering the 12 lectures
  • question pool on the website
  • you’ll need to write stuff on your own (not a-b-c-d, more like 2-3 sentences)
  • explanation of terms/concepts
    • no exact formulas needed (if needed, they might be provided)
    • but you should know the principles of how stuff works
  • relationships between concepts ("what’s the difference between X & Y")
  • designing a dialogue system for a domain
  • focus on important stuff (mostly what’s mentioned in the summaries)

• Mark: 3:1 weighted exam-lab exercises
  • 60 % = pass (C), 73+% = B, 88+% = A