AI in Context of Text Generation

Ondřej Dušek

9.3.2023
Natural Language Generation

• Task of automatically producing text in e.g. English (or other language)
• many subtasks:

<table>
<thead>
<tr>
<th>task</th>
<th>input</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>unconditional language generation</td>
<td>Ø</td>
<td>arbitrary text</td>
</tr>
<tr>
<td>conditional language generation</td>
<td>short text prompt</td>
<td>continuation text</td>
</tr>
<tr>
<td>machine translation</td>
<td>text in language A</td>
<td>text in language B</td>
</tr>
<tr>
<td>summarization</td>
<td>long text</td>
<td>text summary</td>
</tr>
<tr>
<td>image captioning</td>
<td>image</td>
<td>image caption</td>
</tr>
<tr>
<td>question answering</td>
<td>question</td>
<td>answer</td>
</tr>
<tr>
<td>end-to-end dialogue response generation</td>
<td>user input</td>
<td>system response</td>
</tr>
<tr>
<td><strong>data-to-text generation</strong></td>
<td>structured data</td>
<td>description of the data</td>
</tr>
<tr>
<td><strong>dialogue response generation</strong></td>
<td>dialogue act</td>
<td>system response</td>
</tr>
</tbody>
</table>

NLG in a narrow sense
• **data-to-text NLG** = verbalizing structured outputs
  • e.g. RDF triples (=2 entities & relation), tables, dialogue acts … → text

• main usage:
  • reports based on data (weather, sports…)
  • dialogue systems (Siri/Google/Alexa…)

---

Abarth 1000 GT Coupé | design company | Gruppo Bertone
Gruppo Bertone | foundation place | Turin
Gruppo Bertone | country | Italy

NLG

Gruppo Bertone, of Turin Italy, designed the Abarth 1000 GT Coupe.

---

(Kasner et al., 2021) [https://aclanthology.org/2021.inlg-1.25](https://aclanthology.org/2021.inlg-1.25)
NLG Objectives

- general NLG objective:
  
  given **input & communication goal**
  
  create **accurate + natural, well-formed, human-like text**

- additional NLG desired properties:
  
  • variation (avoiding repetitiveness)
  
  • simplicity (saying only what is intended)
  
  • adaptability (conditioning on e.g. user model)
NLG Subtasks (Textbook Pipeline) = how proper NLG had to be done before neural approaches

NLG

deciding what to say

content planning

content plan

sentence planning / microplanning

sentence plan

surface realization

text

selecting content according to a communication goal (typically handled by dialogue manager in dialogue systems)

organizing content into sentences, merging sentences, choosing referring expressions

linearization according to grammar, word order, morphology

deciding how to say it

inputs
NLG Subtasks (Textbook Pipeline)

Example: classical NLG pipeline in the medical domain

inputs

content planning

content plan

sentence planning

sentence plan

realization

(inputs)
NLG Basic Approaches

- **hand-written prompts** (“canned text”)
  - trivial – hard-coded, doesn’t scale (good for IVR/DTMF phone systems)

- **templates** (“fill in blanks”)
  - simple, but much more expressive
  - can scale if done right, still laborious
  - most commercial systems today!

- **grammars & rules**
  - experimental, pipelines, more expressive but more laborious

- **machine learning** (neural LMs → →)

```
[nt] is a [et] in the [a] area.
```

```
name = Blue Spice
eat_type = pub
area = riverside
```

Blue Spice is a pub in the riverside area.
Template-based NLG – Examples

Facebook

{user} shared {object-owner}'s {=album} {title}  
Notify user of a close friend sharing content

* (user) is female. (object-owner) is not a person or has an unknown gender.

{user} sdílila (=album) "(title)" uživatele (object-owner)  
{user} sdílila (object-owner) uživatele (=album) (title)

+ New translation

Public Transport Dialogue

"iconfirm(to_stop={to_stop})&iconfirm(from_stop={from_stop})":  
"Alright, from {from_stop} to {to_stop},",

"iconfirm(to_stop={to_stop})&iconfirm(arrival_time_rel="{arrival_time_rel}")":  
"Alright, to {to_stop} in {arrival_time_rel},",

"iconfirm(arrival_time="{arrival_time}")":  
"You want to be there at {arrival_time},",

"iconfirm(arrival_time_rel="{arrival_time_rel}")":  
"You want to get there in {arrival_time_rel},",

(Alex public transport information rules)  
https://github.com/UFAL-DSG/alex

inflection rules

(Facebook, 2019)
Neural NLG

• 1 step, end-to-end
  • feed input data (linearized)
  • directly generates text word-by-word, left-to-right

• Transformer neural architecture
  • encoder (takes input) – decoder (produces output)
  • alt.: decoder-only (both input & output)

• Train fully from input-output pairs
  • Needs more training data (~10k range, 10x more than before)

• Much more fluent outputs

• Opaque & has **no guarantees on accuracy**
  • used essentially as a black box, internals unknown
Neural NLG: Transformer Models

1) **encoder**: encode linearized data

- feed-forward network, ReLU activations
- attention over all of input (=weighted combination)
- source embeddings (=vectors of numbers)
  - examples: <pad>, inform, request, food, area, price, [name], ...
- positional encoding (indicate position in sentence)
- vocabulary is numbered

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

- starting symbol
- feed output to next step
- probability distribution over the whole vocabulary
- target (word) embeddings
  - examples: 0: <pad>, 1: <start>, 2: <stop>, 3: the, 4: restaurant, 5: area, 6: is, ...
- attention over all of input & output generated so far (self-attention)

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

• Trained to produce sentences from data
  • replicate exact word at each position

• **Supervised** learning
  • initialize model with random parameters
  • didn’t hit the right word → incur **loss**, update parameters

Blue Spice | price | expensive → **NLG** → Blue Spice is expansive
reference:
*Blue Spice is expensive*

• Very **low level**, no concept of sentence / text / aim
Neural NLG: Pretraining + Finetuning

1. **Pretrain** a model on huge data *(self-supervised, language-based tasks)*
   - text-to-text (~ editing)
   - autoencoding & denoising

2. **Fine-tune** for your own task on your smaller data *(supervised)*
   - same as (↑), but much better starting point

   • Models free for download *(https://huggingface.co/)*
     - BERT/RoBERTa, GPT-2, BART, T5…
     - 100k-1B parameters – runs easily on regular GPUs

(Lewis et al., 2020)
https://www.aclweb.org/anthology/2020.acl-main.703
End-to-end NLG with a Pretrained LM

- Use a pretrained LM
  - e.g. (m)BART (GPT-2, T5… ~ 100M-1B params)
- Linearize data
  - concatenate, tokenize data
- Finetune PLM
  - direct data-text mapping: black box
  - needs domain-specific data
    - scarce (~10k max)
    - noisy (crowdsourced)
  - no guarantees on accuracy

Arrabiata sauce is found in Italy where capital city is Rome.
NLG with a pretrained LM: Results

Good

- Generally fluent and accurate
- Robust on input perturbations
- Can be multilingual

Bad

- Fails to generalize
  - factual or grammar errors
  - specifically on unseen relations
- Hallucinations
  - connects unrelated data

in: Bakewell tart | ingredient | Frangipane
out: Франжипан - один из ингредиентов тарта Бейквелл.
(=Frangipane is one of the ingredients of the Bakewell tart.)

in: Ciudad_Ayala | populationMetro | 1777539
out: The population metro of Ciudad Ayala is 1777539.
not seen in training data

in: Nurhan Atasoy | birth date | 1934-01-01 ▶
Nurhan Atasoy | residence | Istanbul ▶
Nurhan Atasoy | nationality | Turkish people
out: Nurhan Atasoy was born on January 1, 1934 in Istanbul and is a
Turkish national.
residence, not birthplace!
Large language models (LLMs): Pretrain & prompt

• 10-100B parameters
  • hard to run in-house (OPT, BLOOM, LlaMa) or not free (GPT-3, ChatGPT, LaMDa)
  • ~10B can run on regular GPUs with 8-bit mode
    • inference only!
• architecture mostly the same
  • 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

https://huggingface.co/blog/hf-bitsandbytes-integration
Prompting examples

GPT2 – THEaiTRE

<table>
<thead>
<tr>
<th>Prompt</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Below deck. A Russian war cruiser began attacking the ship. Ferryman: We have to abandon the ship now! Nina: How far is the mainland</td>
<td>Ferryman: Twenty minutes. Nina: It's getting worse, I'm losing contact with him...! Ferryman: There is no hope, Nina! He can't get out of there! Nina: But we must go now! The Russians will destroy us if they see us - and I want you here as my emergency contact!! You are a brave man! I am grateful for that! Now! Go back!! Go back!!! Ferryman: ...You don't understand. I can only make it through the door and back in five more hours. You won't survive it!</td>
</tr>
</tbody>
</table>

GPT3.5 data-to-text

Write a short description based on data.

Example: Input data: name = Cocum, eatType = coffee shop, food = English, priceRange = high, customer rating = average, familyFriendly = yes
Output description: An English coffee shop named Cocum, though child-friendly, has only an average rating given its high prices.

Input data: eatType = pub, food = Indian, priceRange = cheap, name = Cotton, near = Portland Arms
Output description: Cotton is a cheap Indian pub located near the Portland Arms.

https://theaitre.com/demo

https://platform.openai.com/playground/
Accuracy in NLG

- **NLG semantic accuracy** (fidelity) = input-output correspondence
- Basic error types:
  - **hallucination** = output not grounded in input
    - conflicting with input / unrelated to it
  - **omission** = input not verbalized

OMitted: Blue Spice | eat_type | pub
Blue Spice | area | riverside

NLG: You can bring your kids to Blue Spice in the riverside area.

Hallucinated (not grounded)

- Approx. measure: logical entailment (NLI)
  - output entailed by data & vice-versa, neural models available (BART-NLI)

(Dušek & Kasner, 2020) [https://aclanthology.org/2020.inlg-1.19/](https://aclanthology.org/2020.inlg-1.19/)
Data Fixes

- NLG errors are often caused by **data errors**
  - ungrounded facts (← hallucinating)
  - missing facts (← forgetting)
  - noise (e.g. source instead of target)
    - just 5% untranslated stuff kills an NMT system
- easy-to-get data (web, crowdsourcing) are noisy
- **cleaning** improves situation a lot
  - can be done semi-automatically, up to a point
- **augmentation** – creating synthetic data: more = better
  (assuming reasonable quality texts looking like desired outputs)
  - synthesizing/guessing input for unlabeled texts
  - recombining existing texts
  - paraphrasing

(Chang et al., 2021) https://aclanthology.org/2021.eacl-main.64/

Khayrallah & Koehn, 2018 https://www.aclweb.org/anthology/W18-2709
Better decoding

1. Generate more outputs
   - sampling, beam search…

2. Use some kind of verification system & score outputs
   - language understanding
   - natural language inference

3. Pick the one with the best score
   - still might not be 100% correct

AI in Context: Text Generation

(Harkous et al., 2020)

**<data> name[Zizzi] eatType[bar] <text> Zizzi is a bar .**

prompt (fed into finetuned GPT-2)  output given the prompt
Better Training: InstructGPT/ChatGPT

- Reinforcement learning with human feedback (RLHF)
- multi-step training (start from GPT3/code model)

1) GPT3 supervised finetuning
   - standard, but better data (“in-domain”)

2) evaluation/ranker model
   - lots of rankings
   - used to train RL rewards

3) RL with proximal policy optimization
   - more truthful, can be safer
   - can be smaller (1.3B InstructGPT > 175B GPT3)

(Chen et al., 2022) http://arxiv.org/abs/2203.02155
https://openai.com/blog/chatgpt/, https://www.youtube.com/watch?v=0A8ljAkdFtg
ChatGPT

- Impressive text editing & composition skills
- Still hallucinates a lot
  - only uses information it memorized

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.

AI in Context: Text Generation

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

Arr, 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!

I only teach dialogue systems

no PhD graduate yet

no other tasks than generation & dialogue

14, depending how you count
Adding Search: 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 via attention
  - 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/
(Xu et al., 2021) http://arxiv.org/abs/2107.07567
Data-to-text: Editing only

- Represent input data by templates
  - handcrafted, but not so many needed (1 per input fact/triple)
  - entities inserted verbatim, don’t need to be fluent

- Neural LMs to **fuse & rephrase:**
  - All text-to-text steps (=editing only, making text more fluent)
    1. **order** (put related stuff together)
    2. **aggregate** (into sentences)
    3. **compress** (produce shorter sentences)

- Less space for semantic errors
  - Only use LMs for what they’re good at – fluency

- Can use large general-domain data (~1M+)

- Works **zero-shot** – needs no in-domain data (just the templates)

(Kasner & Dušek, 2022)
https://aclanthology.org/2022.acl-long.271
WikiFluent Corpus

- Wikipedia 1\textsuperscript{st} paragraphs
  - human-written sentences as targets
  - creating artificial source data resembling single-triple templates

- Data creation process:
  1) split sentences (split & rephrase LM)
  2) replace pronouns
  3) randomize order
  4) opt. filter by logical entailment (NLI LM)

- much bigger than in-domain data (~1M sentences)

human-written target

\begin{quote}
The Westmeath Examiner is a weekly newspaper in Westmeath, Ireland. It was founded in 1882.
\end{quote}

\begin{quote}
\textit{split-and-rephrase} \hspace{1cm} \textit{original paragraph}
\end{quote}

\begin{quote}
The Westmeath Examiner is a weekly newspaper. It is located in Westmeath, Ireland. It was founded in 1882.
\end{quote}

\begin{quote}
\textit{coreference replacement}
\end{quote}

\begin{quote}
The Westmeath Examiner is located in Westmeath, Ireland. The Westmeath Examiner was founded in 1882.
\end{quote}

\begin{quote}
\textit{pronouns resolved} \hspace{1cm} \textit{processed paragraph}
\end{quote}

artificial source
Pipeline modules

1) Templates
2) Ordering
   • BART LM with a pointer network
3) Aggregation
   • RoBERTa LM + token classification
   • 0/1: same/other sentence
4) Paragraph compression
   • BART LM – generation
   • All trained on WikiFluent
   • huge (1M), domain-general, accurate
   • Good accuracy & fluency
   • though still not 100% accurate
Summary

• **NLG is useful** in many applications
  • and not really well-defined (MT, captioning, summarization…)
• Can be solved by **templates** pretty well
• **Neural** models: much better fluency
  • more data-hungry
  • not accurate!
• **pretrained LMs**: finetuning / **LLMs**: prompting
  • even more fluent, less data hungry, still not accurate
• fixes: reranking, RLHF, grounding, text editing
• **still not 100% accurate** – needs more control, more semantics
  • we’re working on that right now
Thanks

Contacts:

Ondřej Dušek
odusek@ufal.mff.cuni.cz
https://tuetschek.github.io
@tuetschek

Link to these slides:


Supported by: European Research Council (ERC StG No. 101039303 NG-NLG), Charles University projects PRIMUS/19/SCI/10, GAUK 140320 and SVV 260575, Apple NLU Research Grant for Heriot-Watt University and Charles University.
Using LINDAT/CLARIAH-CZ Research Infrastructure resources (Czech Ministry of Education, Youth and Sports project No. LM2018101).