AI in Context of Text Generation

Ondřej Dušek

9.3.2023

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

Natural Language Generation

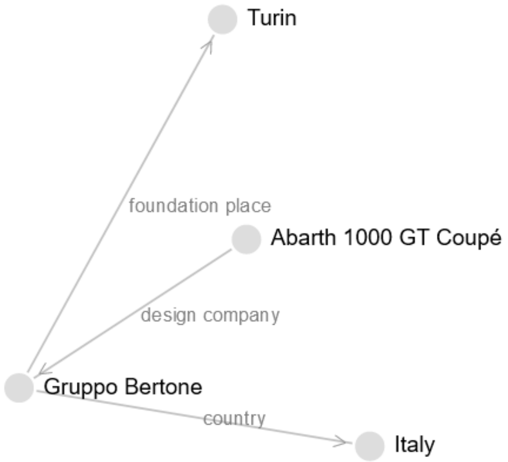
- Task of automatically producing text in e.g. English (or other language)
- many subtasks:

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

NLG in a narrow sense

Data-to-text NLG

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



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.

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

Team	Win	Loss	Pts	…
Mavericks	31	41	86	
Raptors	44	29	94	

Player	AS	RB	PT	…
Patrick Patterson	1	5	14	
Delon Wright	4	3	8	
…				

→

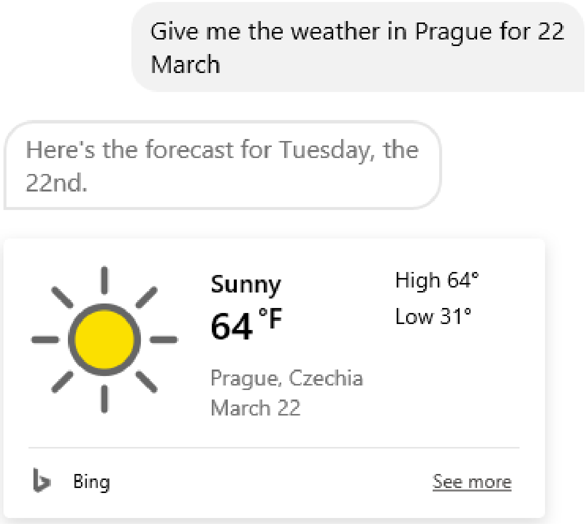
- *The Toronto Raptors, which were leading at halftime by 10 points (54-44), defeated the Dallas Mavericks by 8 points (94-86).*

…

- *Patrick Patterson provided 14 points on 5/6 shooting, 5 rebounds, 3 defensive rebounds, 2 offensive rebounds and 1 assist.*

…

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



Cortana

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

deciding
how to say it

inputs

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

content plan

sentence planning / microplanning — organizing content into sentences, merging sentences, choosing referring expressions

sentence plan

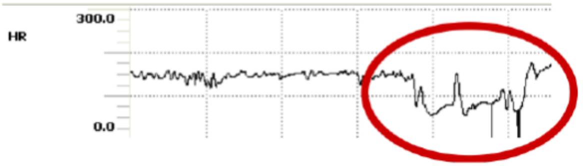
surface realization — linearization according to grammar, word order, morphology

text

NLG Subtasks (Textbook Pipeline)

(Gatt & Krahmer, 2018)
https://jair.org/index.php/jair/article/view/11173

Example: classical NLG pipeline in the medical domain



inputs

→ *content planning* →

TSEQUENCE

BRADYCARDIA (17:01:15) | BRADYCARDIA (17:03:57) | BRADYCARDIA (17:06:03)

content plan

→ *sentence planning* →

$$\begin{bmatrix} \textit{Event} & \\ \text{TYPE} & \textit{existential} \\ \text{PRED} & \textit{be} \\ \text{TENSE} & \textit{past} \\ \text{ARGS} & \begin{bmatrix} \text{THEME} & \{b_1, b_2, b_3\} \\ \text{MIN-VAL} & 69 \end{bmatrix} \end{bmatrix}$$

sentence plan

→ *realization* →

S
PRO | VP_{past}
there | V | NP_{pl} | PP
be | *three successive bradycardias* | *down to 69*

text

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!

[name] *is a* ***[eat_type]*** *in the* ***[area]*** *area.*

name	= Blue Spice
eat_type	= pub
area	= riverside

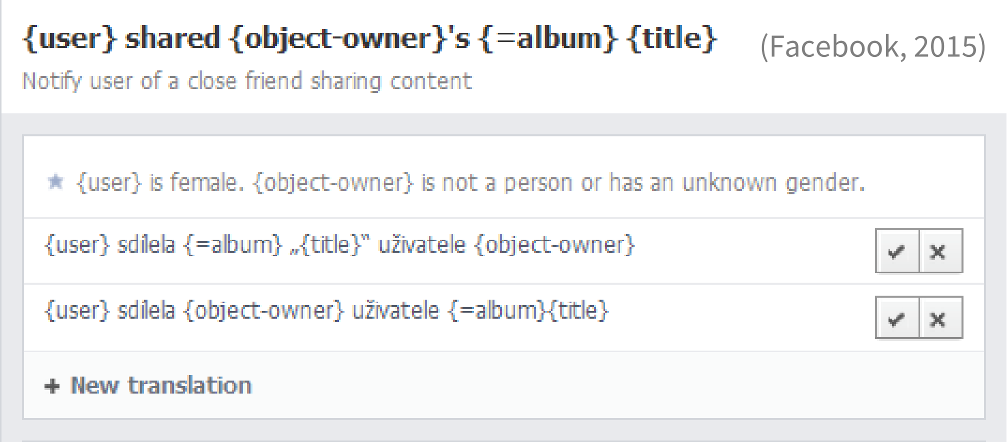
+

Blue Spice *is a* ***pub*** *in the* ***riverside*** *area.*

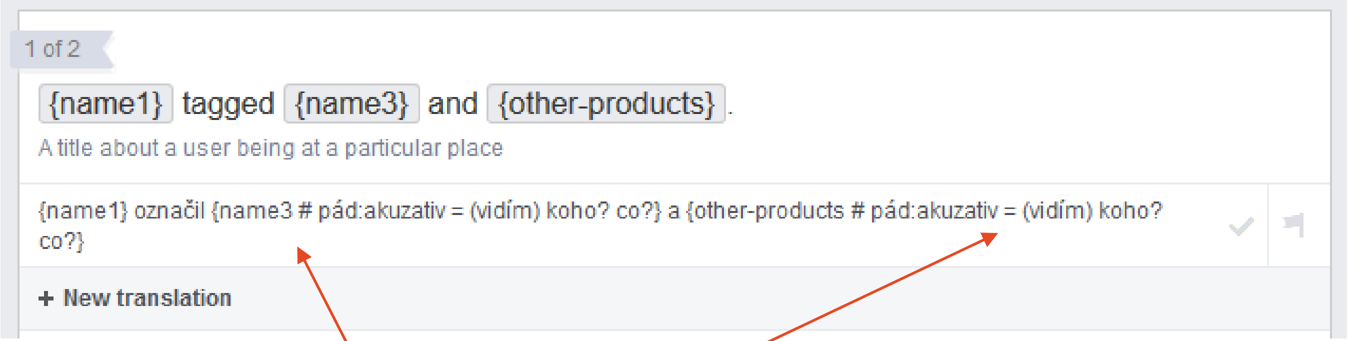
- **grammars & rules**
  - experimental, pipelines, more expressive but more laborious
- **machine learning** (neural LMs → →)

Template-based NLG – Examples

Facebook



(Facebook, 2015)



(Facebook, 2019)

inflection rules

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

Neural NLG

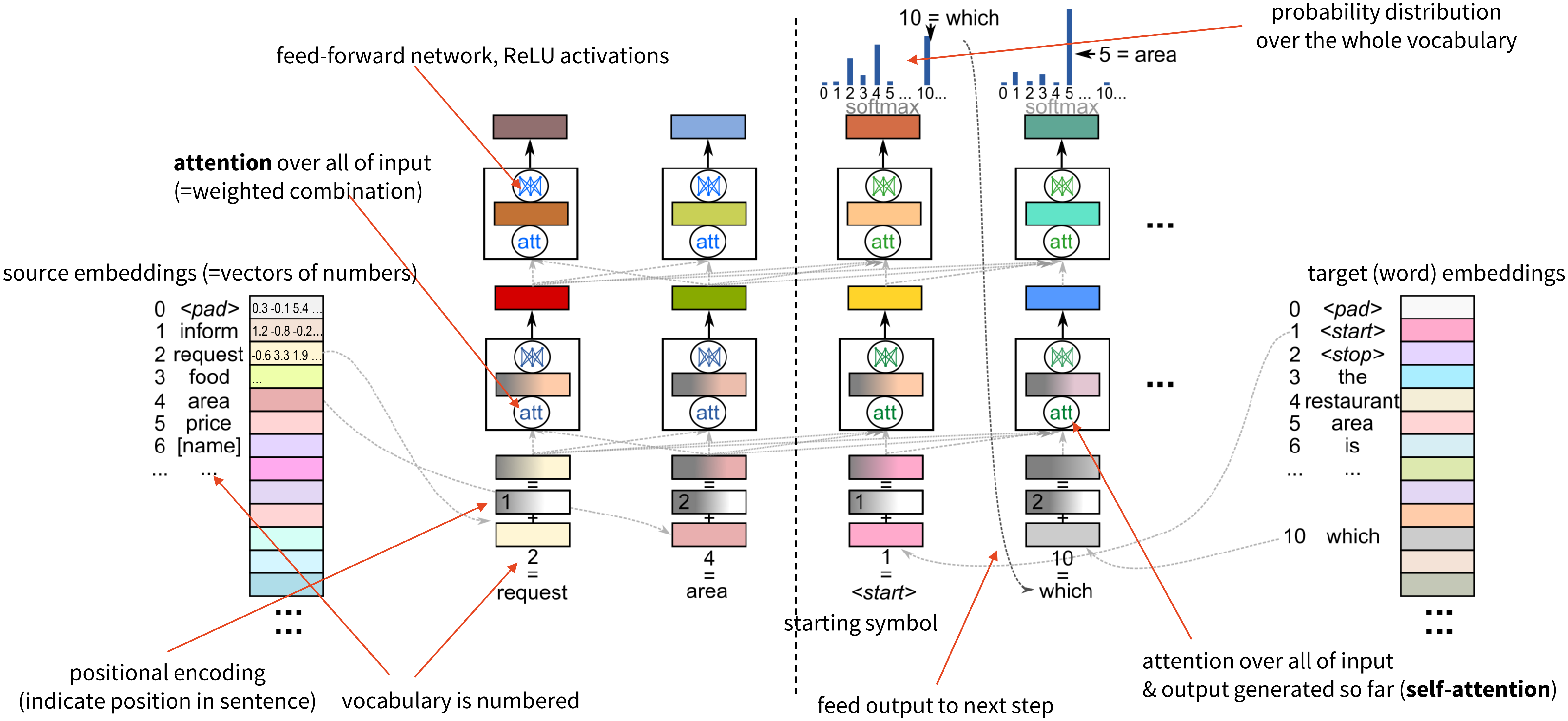
- 1 step, **end-to-end**
  - feed input data (linearized)
  - directly generates text word-by-word, left-to-right
- **Transformer** neural architecture
  - **encoder** (takes input) – **decoder** (produces output)
  - alt.: decoder-only (both input & output)
- **Train** fully from input-output pairs
  - 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

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

1) encoder: encode linearized data

2) decoder: decode text word-by-word



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

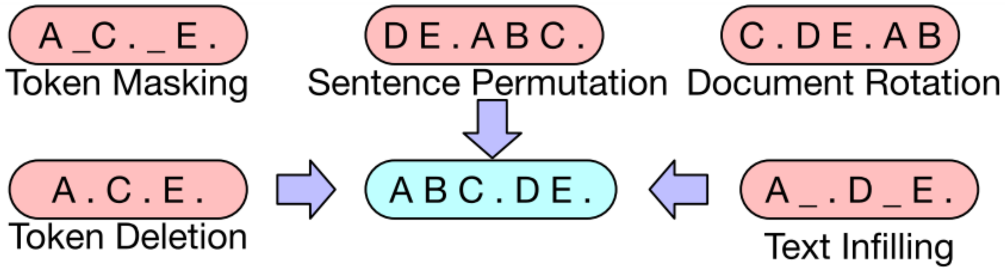
Blue Spice | price | expensive → NLG → *Blue Spice is expensive*
cheap
pricey
in the expensive price range

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



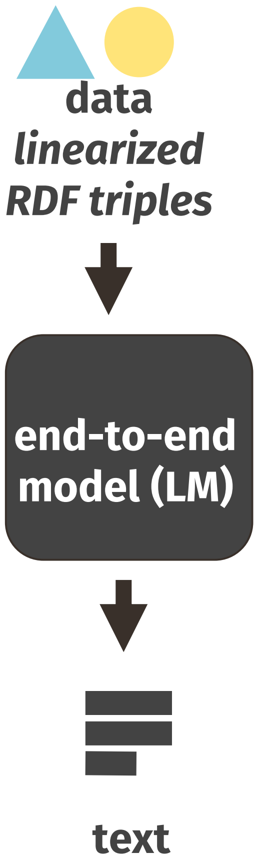
(Lewis et al., 2020)
https://www.aclweb.org/anthology/2020.acl-main.703

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

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 | country | Italy ▸ Italy | capital | Rome



Arrabiata sauce is found in Italy where capital city is Rome.

NLG with a pretrained LM: Results

(Kasner & Dušek, 2020)
https://aclanthology.org/2020.webnlg-1.20/

Good

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

in: *Bakewell tart | ingredient | Frangipane*

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

Bad

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

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!

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

- 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

Prompting examples

GPT2 – THEaiTRE

prompt

```
Below deck. A Russian war cruiser began attacking the ship.

Ferryman: We have to abandon the ship now!

Nina: How far is the mainland
```

output

```
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!
```

https://theaitre.com/demo

GPT3.5 data-to-text

prompt

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

Output description: Cotton is a cheap Indian pub located near the Portland Arms.

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/

Data Fixes

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

(Dušek et al., 2019) https://arxiv.org/abs/1911.03905

(Wang, 2019) https://www.aclweb.org/anthology/W19-8639/

- 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

Original MR and an accurate reference

MR name[Cotto], eatType[coffee shop], food[English], priceRange[less than £20], customer rating[low], area[riverside], near[The Portland Arms]

Reference At the riverside near The Portland Arms, Cotto is a coffee shop that serves English food at less than £20 and has low customer rating.

Example corrections

Reference: Cotto is a coffee shop that serves English food in the city centre. They are located near the Portland Arms and are low rated.
Correction: removed price range; changed area
Reference: Cotto is a cheap coffee shop with one-star located near The Portland Arms.
Correction: removed area

A faulty correction

Reference: Located near The Portland Arms in riverside, the Cotto coffee shop serves English food with *a price range of $20* and a low customer rating.
Correction: incorrectly(!) removed price range
– our script's slot patterns are not perfect

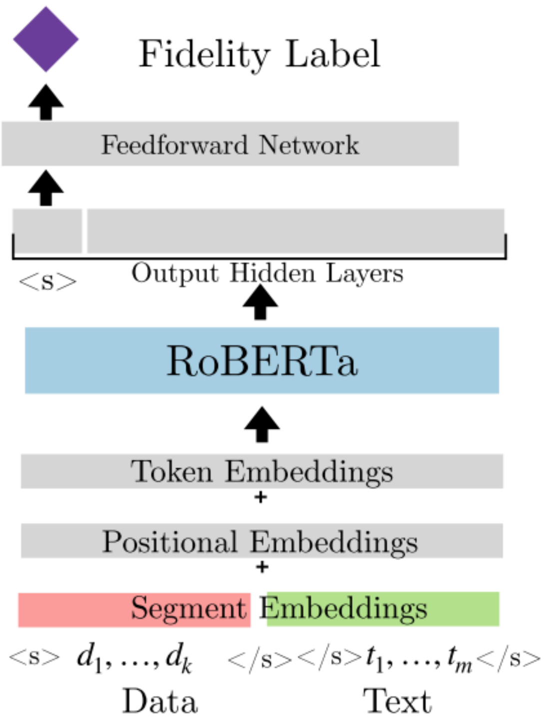
(Elder et al., 2020) https://www.aclweb.org/anthology/2020.acl-main.665

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

Better decoding

(Harkous et al., 2020)
http://arxiv.org/abs/2004.06577

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



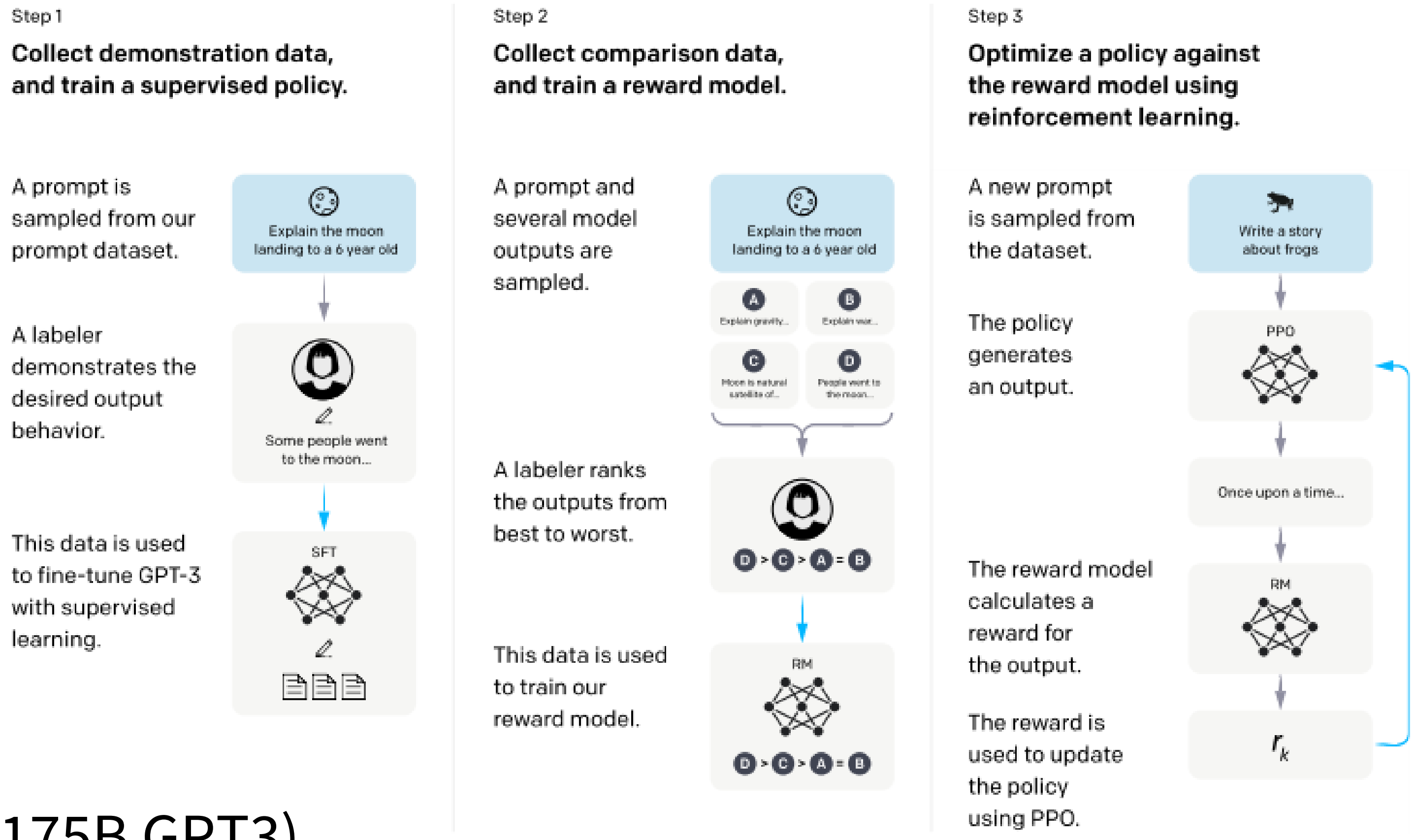
$\underbrace{\textit{<data> name[Zizzi] eatType[bar]}}_{\text{prompt (fed into finetuned GPT-2)}}$ $\underbrace{\textit{<text> Zizzi is a bar .}}_{\text{output given the prompt}}$

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)



(Ouyang 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?

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.

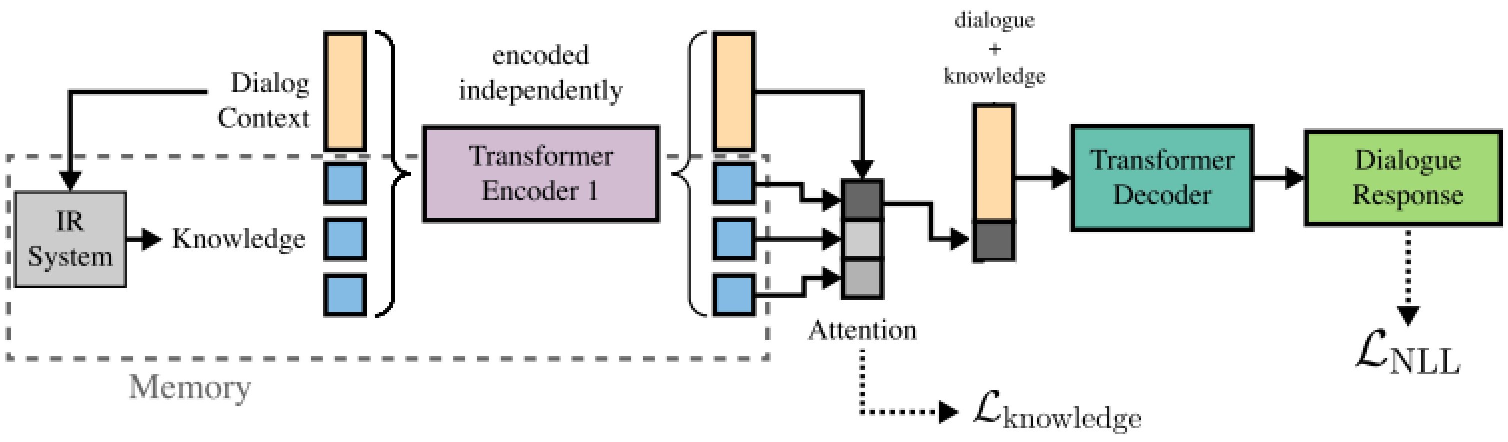
There once was a programmer in Vim
Whose exit strategy was looking grim
They tried to escape
But couldn't find the right shape
Until they remembered to type ":q!" and win!

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!

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



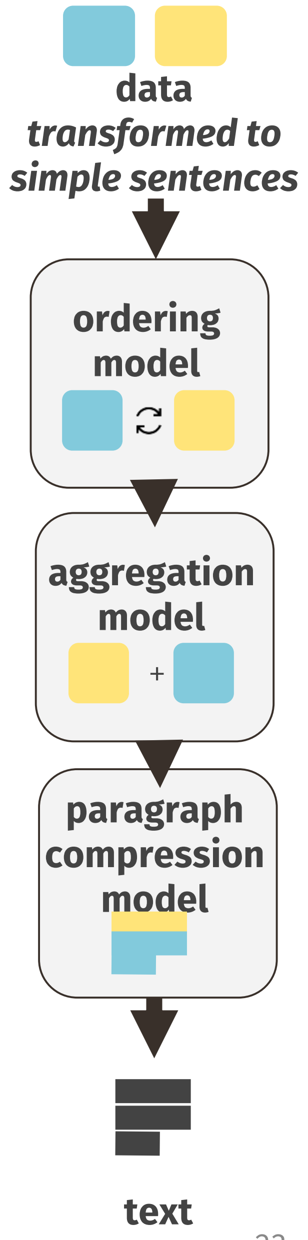
- Models trained to (partially) copy via attention
  - explicitly: classify – copy vs. generate (old)
  - implicitly: shape of data (new)
- Tradeoff: right amount of copying
  - Don't ignore the retrieved
  - Don't copy it verbatim

(Pandey et al., 2018) https://aclanthology.org/P18-1123/
(Weston et al., 2018) https://aclanthology.org/W18-5713/
(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

Data-to-text: Editing only

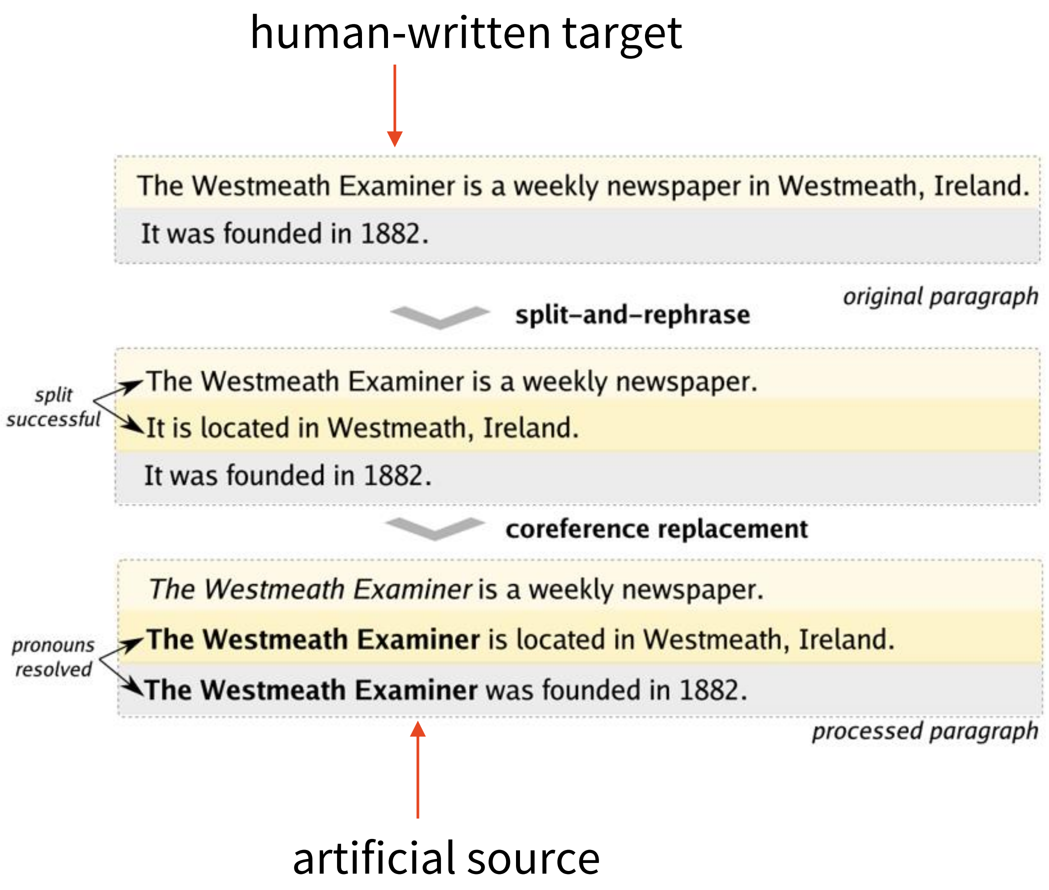
(Kasner & Dušek, 2022)
https://aclanthology.org/2022.acl-long.271/

- 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)

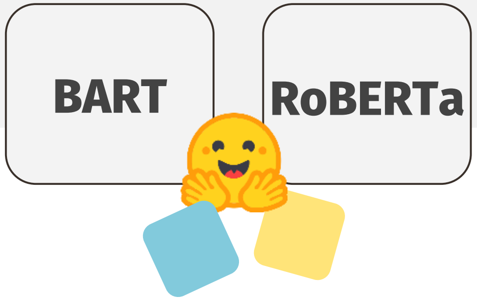


WikiFluent Corpus

- Wikipedia 1st 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)

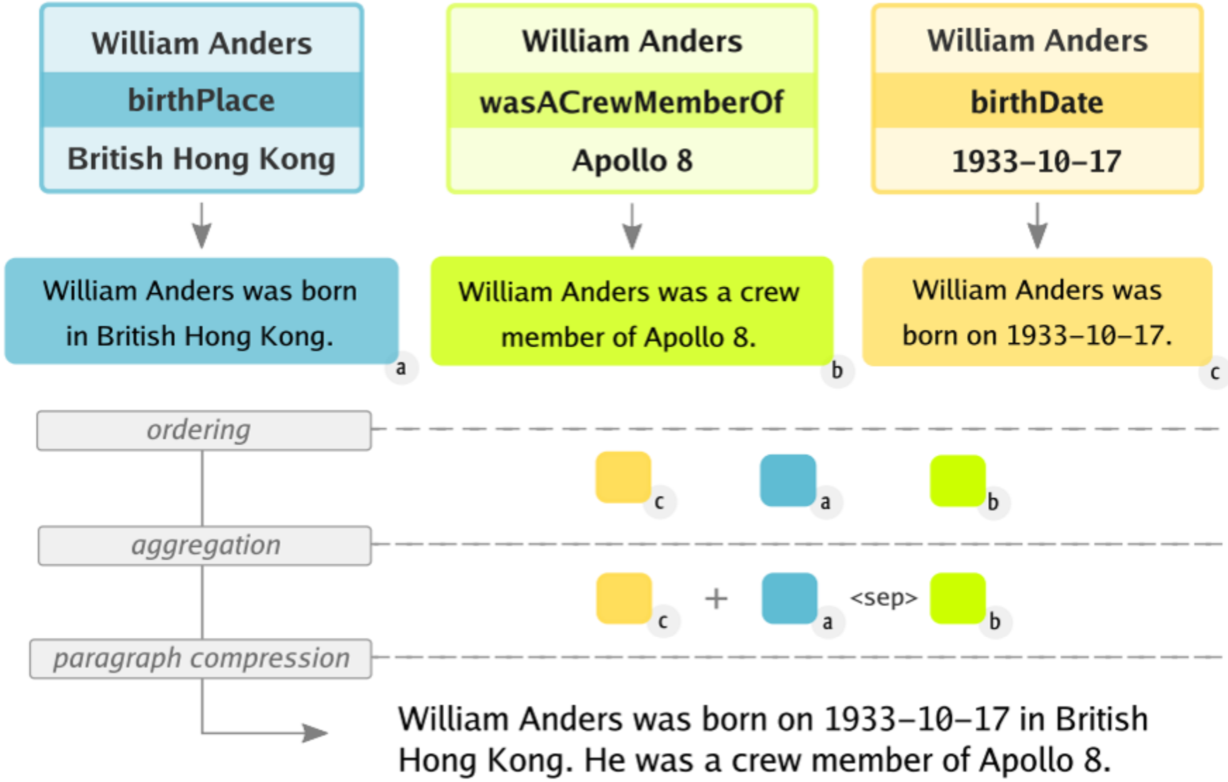


Pipeline modules



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:

http://bit.ly/aivk-nlg

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).

European Research Council
Established by the European Commission