Large Neural Language Models for Data-to-text Generation

Ondřej Dušek

collaboration with **Zdeněk Kasner**

AICZECHIA Seminar
22.3.2022

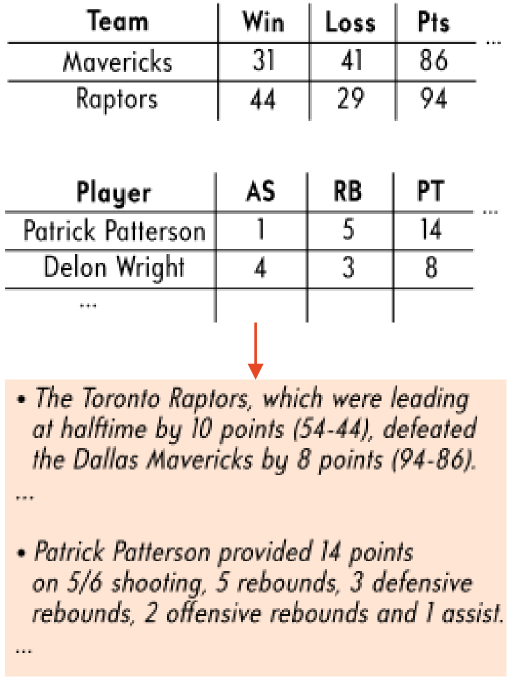
Charles University
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics

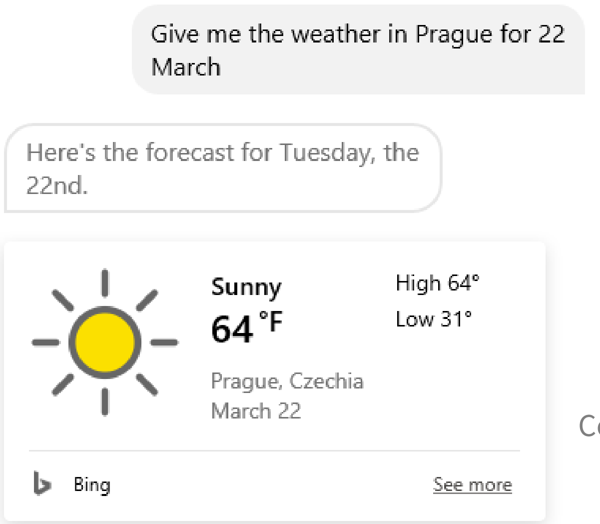
Data-to-text Generation

- **data-to-text NLG** = verbalizing structured outputs
  - RDF triples, dialogue acts etc. → text

Blue Spice | eat_type | pub
Blue Spice | area | riverside → NLG → *Blue Spice is a pub in the riverside area.*

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



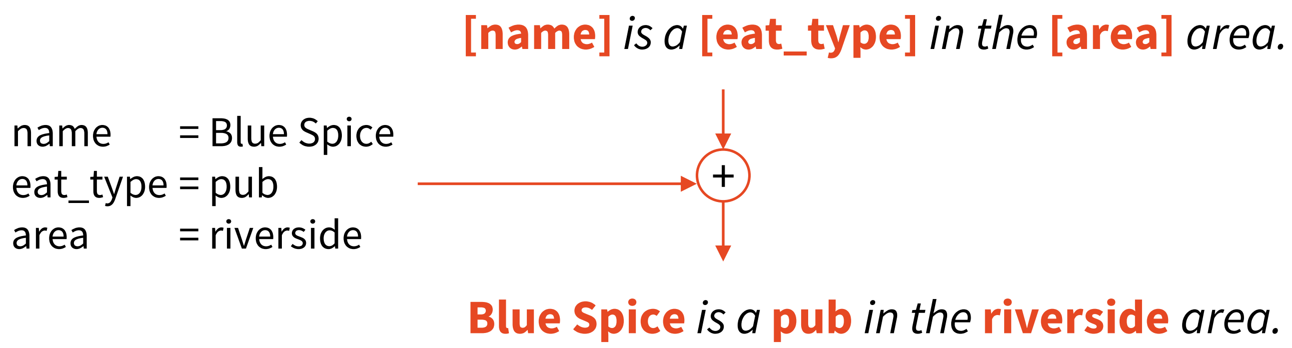


Cortana

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

Neural NLG vs. older methods

- Older methods:
  - **templates** – fill in blanks
    - most commercial systems still!
    - safe, tried & tested
    - needs handcrafting
  - rules/grammars
  - pipelines of statistical models



- Neural models:
  - 1 step, **end-to-end**
  - **Train** fully from input-output pairs (no additional rules etc.)
  - Much more **fluent** outputs
  - Needs more training data (~10k range, 10x more than before)
  - Opaque & has **no guarantees on accuracy**

Accuracy in NLG

- **accuracy** = input-output correspondence
- basic accuracy error types
  - **hallucination** = output not grounded in input
  - **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)

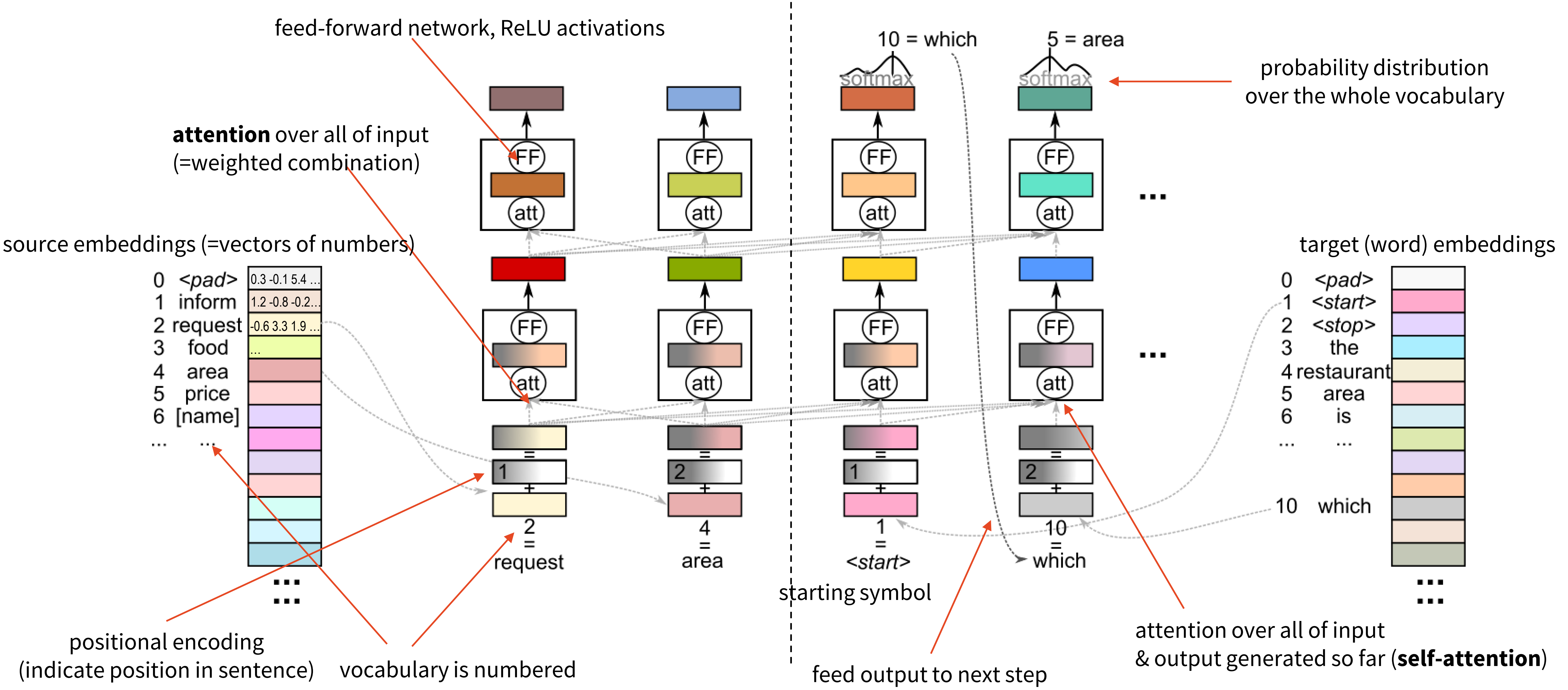
- measure: slot/semantic error rate (**SER**)
  - % incorrect “slots” (=pieces of info)

Neural NLG: Transformer Models (encoder-decoder, seq2seq)

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

1) encoder: encode linearized data

2) decoder: decode text word-by-word



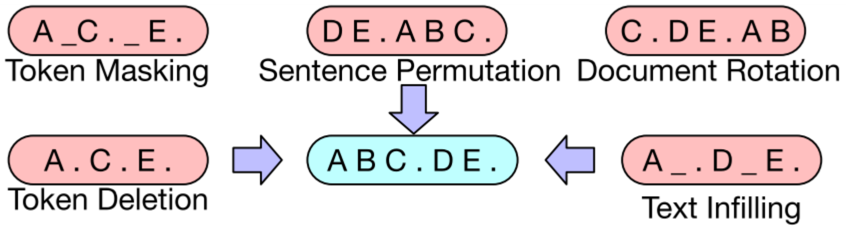
Neural NLG: (Pre-)Training

- Trained to produce sentence in data
  - low-level: exact word at each position

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

reference:
Blue Spice is expensive

- Pretrained language models:
  1. **Pretrain** a model on a huge dataset (**self-supervised**, language-based tasks)
     - text-to-text: autoencoding & denoising
  2. **Fine-tune** for your own task
     on your smaller data (**supervised**)
     - models available online
       – get pretrained model, finetune yourself



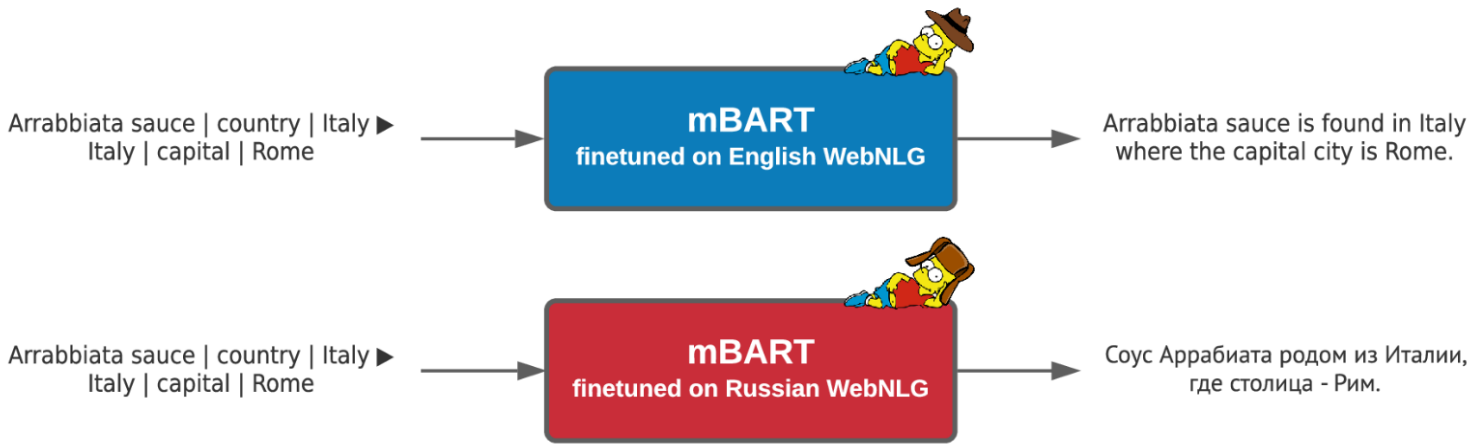
(Lewis et al., 2020)
https://www.aclweb.org/anthology/2020.acl-main.703

NLG with a Pretrained LM: Base

(Kasner & Dušek, 2020)
https://www.aclweb.org/anthology/2020.webnlg-1.20/

- Most basic setup:
  - using mBART pretrained model
  - representing data as text

    subject | predicate | object ▶
    subject | predicate | object
  - finetuning to generate English & Russian



- Fluent outputs, but…
  - fails to generalize
  - hallucinates occasionally

in: *Ciudad_Ayala | populationMetro | 1777539*

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

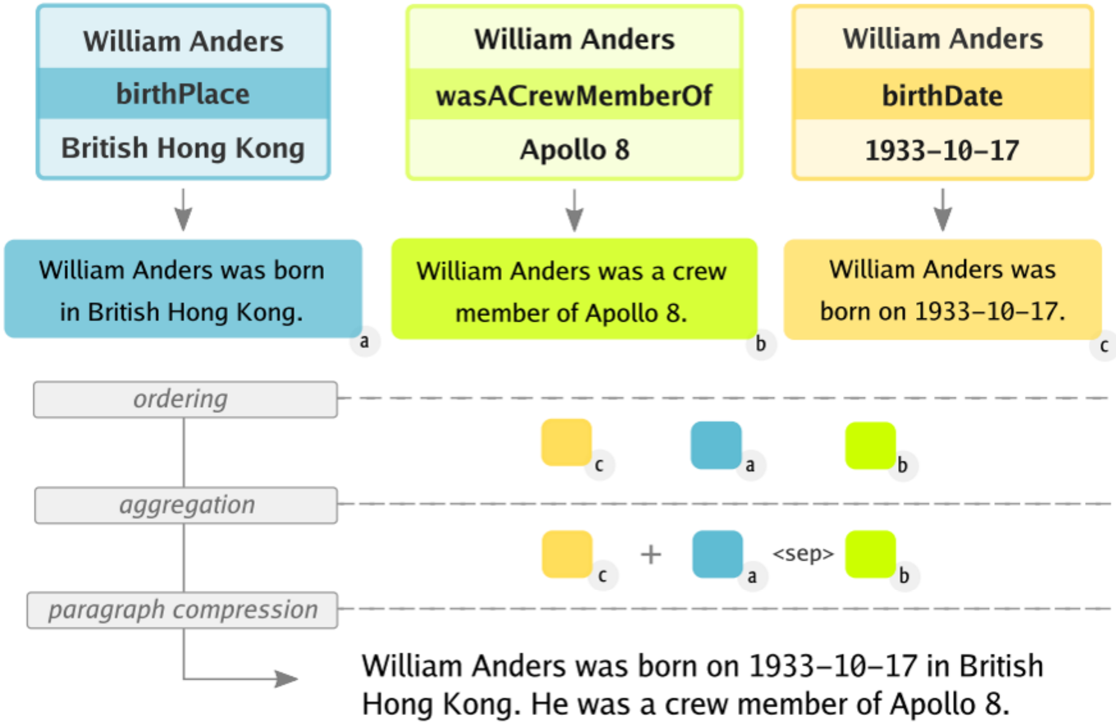
in: *Nurhan Atasoy | birth_date | 1934-01-01 ▶*
Nurhan Atasoy | residence | Istanbul ▶ — residence, not birthplace!
Nurhan Atasoy | nationality | Turkish people

out: Nurhan Atasoy was born on January 1, 1934 in Istanbul and is a Turkish national.

Templates + Neural Fuse & Rephrase

(Kasner & Dušek, 2022)
ACL conference, arXiv coming soon

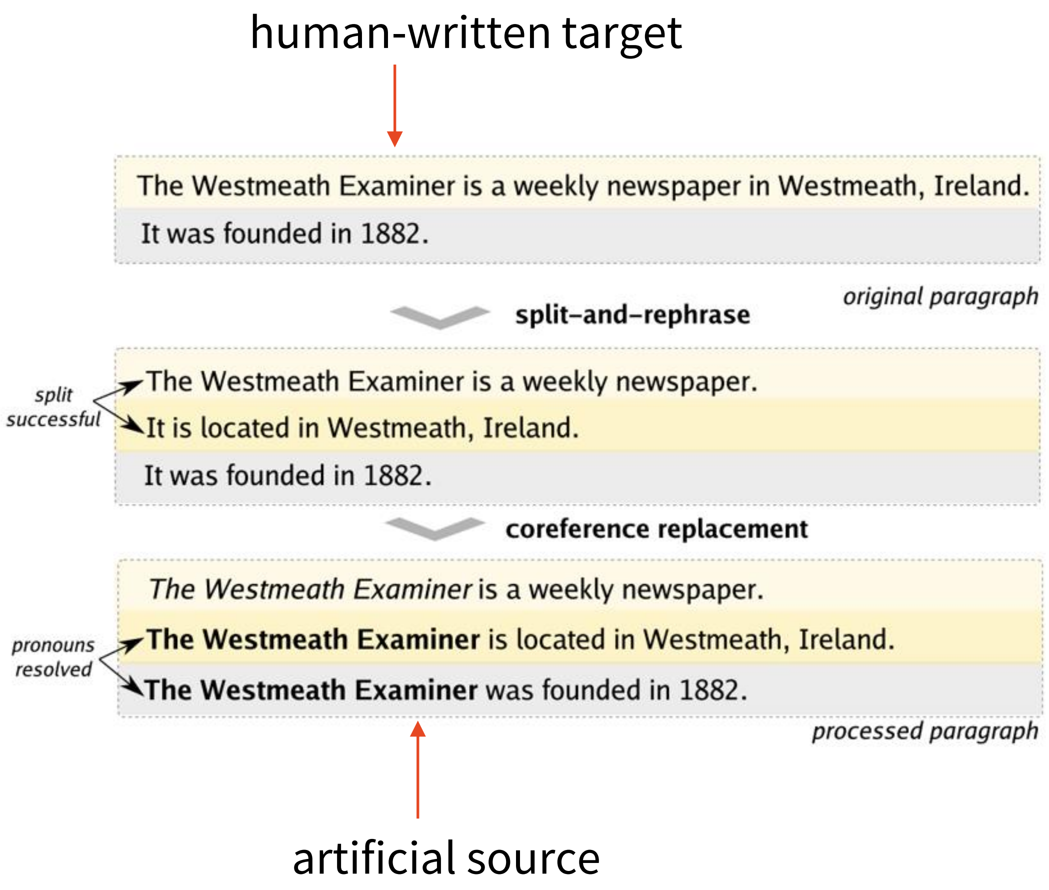
- Templates to represent individual triples
  - Guaranteed accurate
  - Not so many needed (usually)
  - No need for high fluency
- Neural LM to **fuse & rephrase:**
  1) **order** (related together)
  2) **aggregate** (into sentences)
  3) **compress** (produce short sentences)
  - Do what neural models are good at – fluency
  - Less space for semantic errors
- Works **zero-shot** – with no in-domain data (just the templates)



Templates + Neural Fuse & Rephrase

(Kasner & Dušek, 2022)
ACL conference, arXiv coming soon

- 3 neural models, one step each
  - all based on pretrained LMs
- Large Wikipedia data
  - Wikipedia sentences as targets
  - creating artificial source data, which looks like single-triple templates
  1) split sentences
  2) replace pronouns
  3) randomize order
- ~1M sentences, various topics
  - much more than in-domain available



Templates + Neural Fuse & Rephrase

- Good accuracy
  - perfect for simpler data (E2E restaurants)
  - worse for complex data (WebNLG DBPedia knowledge)
- Slightly lower fluency (~older neural systems)
- Can be further improved (reranking/filtering)

E2E	BLEU	Omission/ #facts	Hallucination/#examples
Older neural	40.73	0.016	0.083
Templates	24.19	0.000	0.000
Ours	36.04	0.001	0.001

WebNLG	BLEU	Omission/ #facts	Hallucination/#examples
Rule-based	38.65	0.075	0.101
Older neural	45.13	0.237	0.202
Templates	37.18	0.000	0.000
Ours	42.92	0.051	0.148

input: *Allen Forrest | background | solo singer ▸ Allen Forrest | genre | pop music ▸ Allen Forrest | birthplace | Dothan, Alabama*
templates: *Allen Forrest is a solo singer. Allen Forrest performs Pop music. Allen Forrest was born in Dothan, Alabama.*
output: Allen Forrest is a solo singer who performs Pop music. He was born in Dothan, Alabama.

input: *Wildwood | eatType | restaurant ▸ Wildwood | food | French ▸ Wildwood | area | riverside ▸ Wildwood | near | Raja Indian Cuisine*
templates: *Wildwood is a restaurant. Wildwood serves French food. Wildwood is in the riverside. Wildwood is near Raja Indian Cuisine.*
output: Wildwood is a restaurant serving French Food. It is in the riverside near Raja Indian Cuisine.

input: *Alfa Romeo 164 | relatedMeanOfTransportation | Fiat Croma ▸ Alfa Romeo 164 | assembly | Italy ▸ Italy | capital | Rome*
templates: *Alfa Romeo 164 is related to Fiat Croma. Alfa Romeo 164 was assembled in Italy. Italy's capital is Rome.*
output: Alfa Romeo 164 was assembled in Italy's capital, Rome. It is related to Fiat Croma.

Evaluating Data-to-text NLG

- **n-gram metrics** (BLEU, METEOR)
  - derived from MT, no good for accuracy
  - dubious even as measures for overall quality
- **Neural metrics** (BERTScore, BLEURT) mix accuracy & fluency
  - slightly better than n-gram, but still not ideal
- **SER** evaluation uses regex or exact match
  - tedious to make / inaccurate
  - does not translate to other datasets
- Proper evaluation means full NLU
  - pretrained LMs are good at NLU-like tasks → use them?

Checking for Errors in NLG Output: Natural Language Inference

- **NLI**: relation of premise (= starting point) & hypothesis (= relating text)
  - **E**ntailment = all hypothesis facts are included in premise
  - **N**eutral = not all hypothesis facts included, but no directly opposing facts
  - **C**ontradiction = premise is opposed by hypothesis

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

$\mathbf{H_1}$: *Blue Spice is located in the riverside.* ⟶ **E**
$\mathbf{H_2}$: *You can bring your kids to Blue Spice .* ⟶ **N**
$\mathbf{H_3}$: *Blue Spice is a coffee shop.* ⟶ **C**

- We'll use a vanilla model trained for NLI
- Check entailment in both directions
  - data entails text = no hallucination + text entails data = no omission
- Use templates to represent data (same as previously)

(Dušek & Kasner, 2020)
https://www.aclweb.org/anthology/2020.inlg-1.19

Eval1: NLI Classification

(Dušek & Kasner, 2020)
https://www.aclweb.org/anthology/2020.inlg-1.19

1) **Check for omissions**
   - premise = whole generated text
   - hypothesis = each single fact, loop
     → also checks which fact is omitted
2) **Check for hallucination**
   - premise = concatenated facts
   - hypothesis = whole generated text
     - can't easily split into simpler checks

- output:
  - 4-way – *OK, omission, hallucination, o+h*
  - 2-way – *OK, not_OK*
  - OK confidence (min. **E** confidence)
  - list of omitted facts

Blue Spice | eat_type | pub
Blue Spice | area | riverside

NLG

↓

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

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

H$_1$: Blue Spice is a pub. C: 0.01 N: **0.97** E: 0.02
→ *omission*

H$_2$: Blue Spice is located in the riverside. C: 0.00 N: 0.01 E: **0.99**
→ *OK*

P: *Blue Spice is a pub. Blue Spice is located in the riverside.*

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

C: 0.00 N: **0.99** E: 0.01
→ *hallucination*

---

omission+hallucination

OK: 0.01 omitted: Blue Spice | eat_type | pub

Error Checking with NLI

- WebNLG & E2E data
  - comparison vs. human ratings (WebNLG) & SER regex script (E2E)
  - both datasets: default & backoff-only versions of templates

system	**WebNLG** data	**E2E data**	
		4-way	2-way
Accuracy / agreement	77.5%	91.1%	93.3%

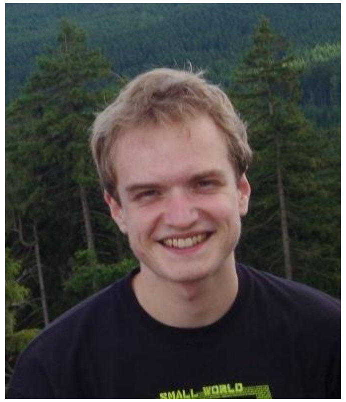
- manual analysis: ca. ½ “errors” are in fact correct
  - annotation noise / SER script errors
  - noisy templates
  - edge cases (*high restaurant*)
  - stuff SER script doesn’t catch (*with full service*)

Summary

- Neural models produce very fluent outputs
  - especially true of pretrained Transformer LMs
  - due to data & model reasons, not guaranteed to be accurate
- There are ways to make them more accurate
  - combining with templates & only editing for fluency
  - constraining the neural component
- Finding errors in NLG is as hard as NLU
  - pretrained LMs are good at some NLU tasks, such as NLI → can be applied
- Many other accuracy-increasing approaches
  - reranking / data cleaning / multi-task training / constrained decoding
  - more to come: semantic formalisms & inference

Thanks

Contact us:



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

Zdeněk Kasner
kasner@ufal.mff.cuni.cz
http://ufal.cz/zdenek-kasner
@ZdenekKasner

References:

- Base pretrained LMs: https://aclanthology.org/2020.webnlg-1.20/
- Fuse & rephrase: coming soon (on arXiv/my website)
- Error checking via NLI: https://aclanthology.org/2020.inlg-1.19/