Accuracy in Neural Text Generation

Ondřej Dušek

Selected topics in Machine Learning and Natural Language Processing

23. 7. 2021

Collaboration with: David Howcroft, Vojtěch Hudeček, Filip Jurčíček, Zdeněk Kasner, Jonáš Kulhánek, Simon Mille, Tomáš Nekvinda, Jekaterina Novikova, Ioannis Konstas, Verena Rieser, Xinnuo Xu

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

Outline

- NLG & Accuracy
  - mainly for data-to-text (~ conditioned on input)
  - motivation: what's new in neural vs. previous systems
- Making neural NLG systems (more) accurate
  1) overgenerate & rerank
  2) cleaning data
  3) additional classifier tasks
  4) explicit planning step
  5) neural editing only
- Detecting inaccuracy in NLG outputs
  1) sentence-level classification
  2) word-level error tagging
- stuff I was involved in, with links elsewhere

Accuracy in NLG

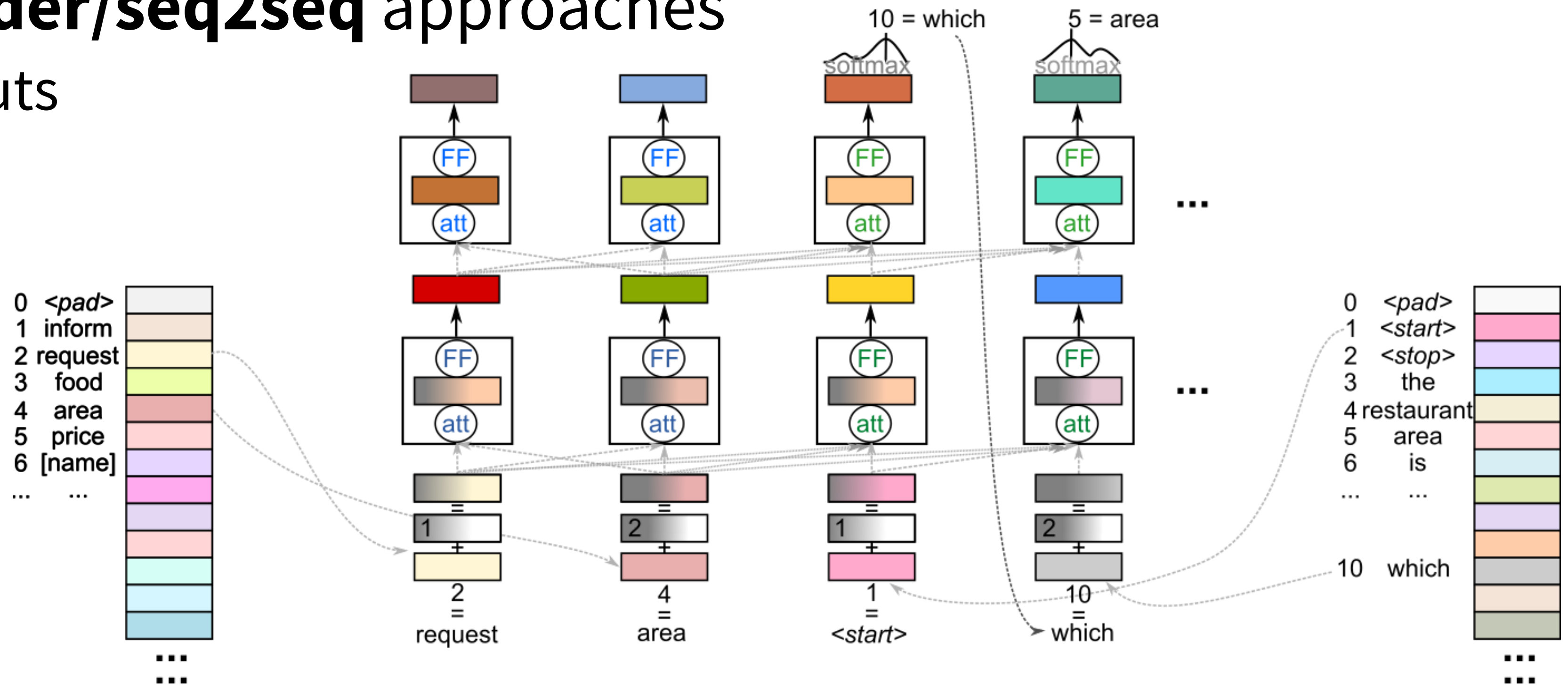
- **data-to-text NLG** = verbalizing structured outputs
  - RDF triples, dialogue acts etc. → text
- **accuracy** = input-output correspondence

Blue Spice | eat_type | pub
Blue Spice | area | riverside
→ NLG → *You can bring your kids to Blue Spice in the riverside area.*

- accuracy error types
  - **hallucination** = output not grounded in input
  - **omission** = input not verbalized
- measurement: **semantic/slot error rate**
- different for other NLG tasks
  - summarization, NLG with content selection: omissions allowed
  - open-domain dialogue, creative text generation: just internal consistency

Neural NLG

- Neural **encoder-decoder/seq2seq** approaches
  - encode linearized inputs
  - decode word-by-word (autoregressively)



- Multiple architectures (but the same principle)
  - RNNs (LSTM, GRU) + attention
  - Transformer (=positional embeddings, feed forward & attention)
  - Pretrained Transformers (+ pretrained on lots of data for a self-supervised task)

Neural NLG vs. older methods

- End-to-end – 1 model does everything
  - previously: pipelines (or “end-to-end” templates, which aren’t trainable)
- No need for fine-grained alignments in data
  - previously: most trainable methods required that

`name [Loch Fyne], eatType[restaurant], food[Japanese], price[cheap], familyFriendly[yes]`

Loch Fyne is a kid-friendly restaurant serving cheap Japanese food.

- Very fluent outputs (especially with pretrained models)
  - previously: formulaic, sometimes incorrect outputs
- Needs more training data (~10k range)
  - previously: 100-1k range
- Opaque & has no guarantees on accuracy

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

E2E NLG Challenge (2017/18)

(Dušek et al., 2020)
http://arxiv.org/abs/1901.07931

- Known domain: restaurant data
- More data than prior approaches
  - 6k MRs, 50k texts
  - (prior: mostly 0-10k)
- Diverse & natural, but noisy
  - crowdsourced, partially based on images
- 17 challenge entries (12 neural)
- **Problems**:
  - **accuracy** – lots of omissions & hallucinations
    - only 100% accurate system: hand-written templates
  - **diversity** – repetitive vs. inaccurate

```
name [Loch Fyne], eatType[restaurant],
food[Japanese], price[cheap],kid-friendly[yes]
```

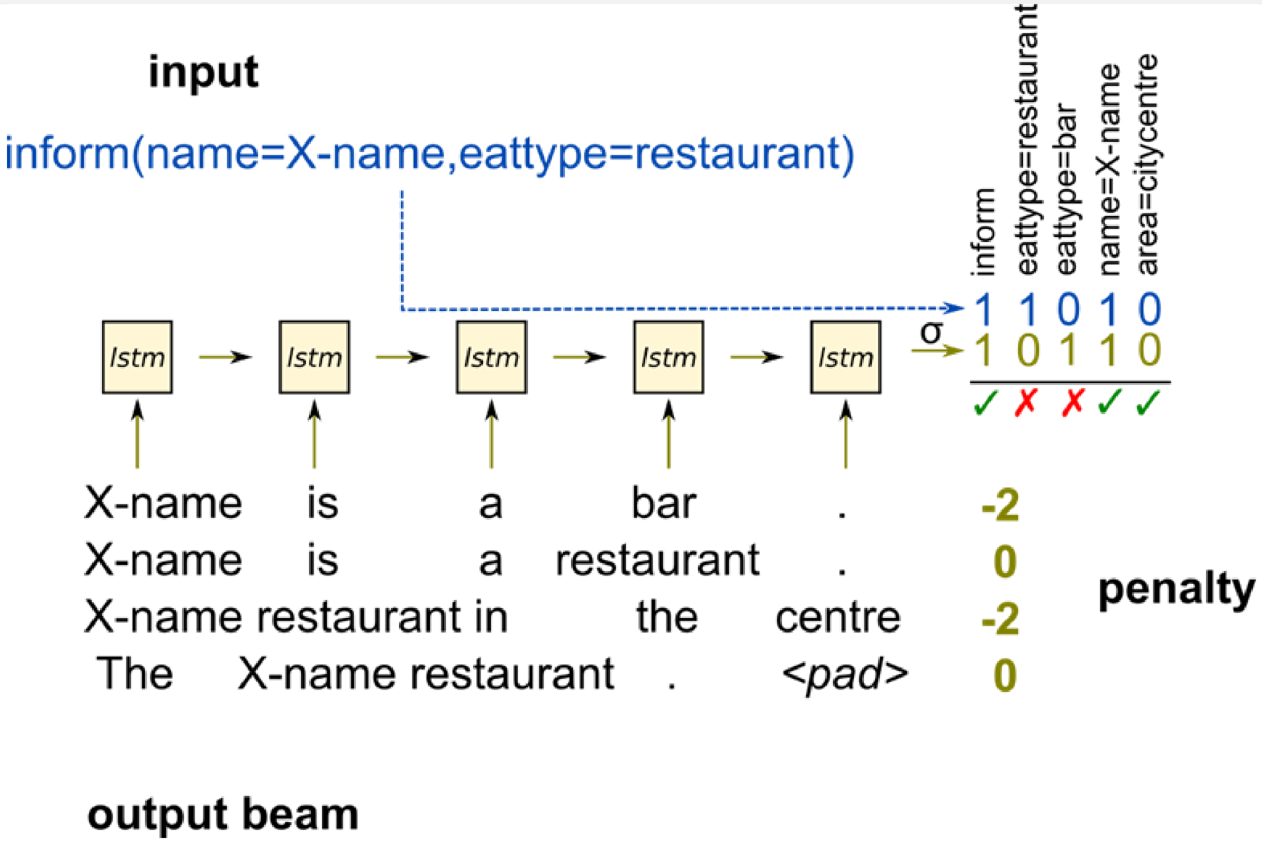
Loch Fyne is a kid-friendly restaurant serving cheap Japanese food.

Serving low cost Japanese style cuisine, Loch Fyne caters for everyone, including families with small children.

NLG1: Reranking

(Dušek & Jurčíček, 2016)
https://aclweb.org/anthology/P16-2008

- 2-step:
  1) Generate multiple outputs
     - beam search
  2) Rerank (penalize inaccuracies)
     - classify MRs
     - penalty for each difference w. r. t. input
- **TGen**
  - LSTM-based seq2seq with attention
  - LSTM-based MR classifier for ranking
- increases accuracy significantly
  - but still can't guarantee it completely
  - also, it's slow



system / E2E	BLEU	SER
Seq2seq	63.4	15.94%
TGen	66.4	4.27%

NLG2: Data cleaning

- NLG errors are often caused by **data errors**
  - ungrounded facts (← hallucination)
  - missing facts (← omission)
  - noise (e.g. source instead of target)
    - just 5% untranslated stuff kills an NMT system

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

- Easy-to-get data are noisy
  - web scraping – lot of noise, typically not fit for purpose
  - crowdsourcing – workers forget/don't care
- E2E data: 11-17% slot error rate
  - approx. 40% references have ≥1 error

(Dušek et al., 2019)
https://www.aclweb.org/anthology/W19-8652/

- Rotowire: 40% ungrounded

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

NLG2: Data cleaning

(Dušek et al., 2019) https://www.aclweb.org/anthology/W19-8652/

- E2E data: SER evaluation script
  - based on regular expressions
  - can be used for data cleaning
- Keep text, adjust MR
  - works up to a point (SER 4.2%, 19% error refs)
  - keep test set, remove overlaps from train
- Retraining Seq2Seq&TGen on cleaned E2E
  - less training examples
  - still 94-97% SER reduction
  - confirmed by manual analysis
- Extensions:
  (Nie et al., 2019) https://www.aclweb.org/anthology/P19-1256
  - cleaning by a trained classifier (two-step)
  - generating more data (& checking)

(Kedzie & McKeown, 2019) https://www.aclweb.org/anthology/W19-8672/

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

system	data	BLEU	SER	
Seq2seq	original	63.4	15.94%	-94%
	cleaned	65.8	0.97%	
TGen	original	66.4	4.27%	-97%
	cleaned	66.2	0.12%	

NLG3: Additional Classification Tasks

(Peng et al., 2020) https://arxiv.org/abs/2005.05298
(Kulhánek et al., 2021) https://arxiv.org/abs/2102.05126

- Generate & classify at the same time
  - additional classification layer
  - on top of decoder – last layer logits, last step
- Aim: robustness – detect problems
  - ½ training data are artificially corrupted
    - corrupted state (does not fit context)
      - whole (SOLOIST)
      - per domain (AuGPT)
    - corrupted system response
- improves dialogue success
  - MultiWOZ (corpus-based & simulation)

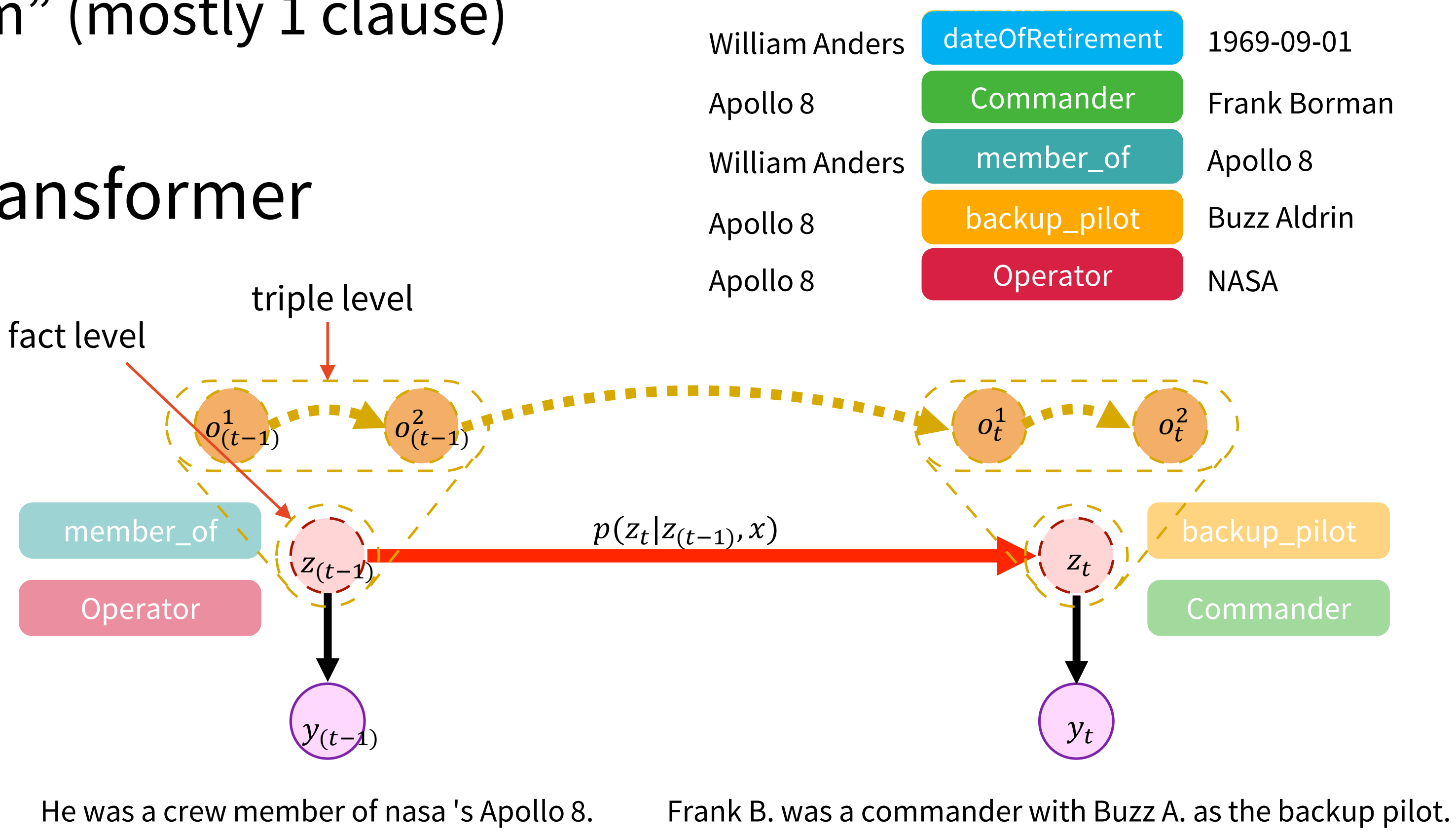
system	inform	success	BLEU
baseline	81.9	64.5	16.3
SOLOIST	81.4	65.8	17.0
AuGPT	83.5	67.3	17.2

	consistent?
i want a cheap italian restaurant { price range = cheap , food = Italian } ok which area ?	✓
i want a cheap italian restaurant { area = north , food = Indian } ok which area ?	✗
i want a cheap Italian restaurant { price range = cheap , food = Italian } what price range ?	✗

NLG4: Planning

(Xu et al., 2021)
https://arxiv.org/abs/2106.05580

- Add an explicit planning step (ordering & aggregation)
- Split texts into **facts** by SRL
  - ~1 event "who did what to whom" (mostly 1 clause)
  - ~1 or more input triples
- Hierarchical HMM planner + Transformer
  1) order triples
  2) aggregate into facts
  3) generate each fact
     - condition on triples for current fact only
- trained: backward algorithm
  - end-to-end with generation
  - no explicit annotation needed



NLG4: Planning

(Xu et al., 2021)
https://arxiv.org/abs/2106.05580

- Stays fluent + is more accurate
  - less “compressed” outputs than Transformer
- Allows explicit control
  - order & aggregation of triples is visible
    - interpretable, allows direct evaluation
  - you can set it manually
    - or set a parameter to control aggregation
- Needs some hacks to make it tractable
  - max. 3 triples per fact, partial hard alignment
- Still not a complete control
  - Transformer may hallucinate

system / E2E	BLEU	SER
TGen	66.4	4.27
Transformer	68.2	5.16
AggGen	64.1	2.16

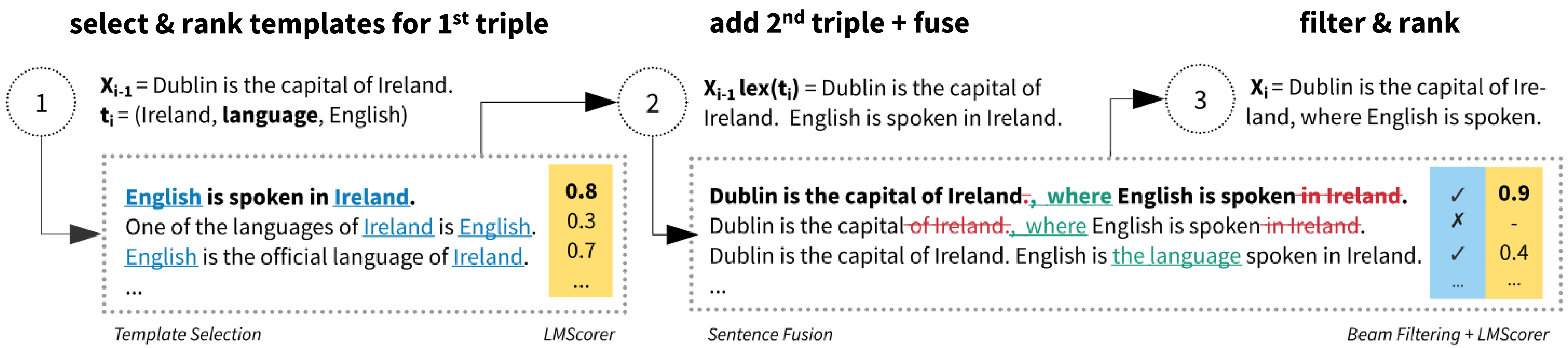
order & agg.	K-τ_{max}	K-τ_{avg}
Human	0.84	0.25
AggGen	0.64	0.21

see also: (Wiseman et al., 2018) http://aclweb.org/anthology/D18-1356
(Moryossef et al., 2019) https://www.aclweb.org/anthology/N19-1236/

NLG5: Iterative Editing

(Kasner & Dušek, 2020)
https://www.aclweb.org/anthology/2020.inlg-1.9

- Concatenate templates & fuse them into sentences by a neural model
  - Template-based generation is accurate
  - Neural model only fuses sentences together
    - Less power = less opportunity to screw up
  - Inaccuracies filtered out & fallback to templates – ensures 0 entity errors
  - Ranking by fluency (neural model)



NLG5: Iterative Editing

(Kasner & Dušek, 2020)
https://www.aclweb.org/anthology/2020.inlg-1.9

- Templates: 1 triple only (extracted from training data + handcrafted + backoff)

Arrabiata sauce | country | Italy

default → *The <subject> is found in <object>.* → *The Arrabiata sauce is found in Italy.*

backoff → *The <predicate> of <subject> is <object>.* → *The country of Arrabiata sauce is Italy.*

- Neural model: LaserTagger – BERT encoder & Transformer decoder
  - vocabulary limited (100 tokens): KEEP, DELETE, ADD word, ADD more words
- Fluency: vanilla GPT-2 geom. mean token cond. probability
- Semantic filter: entity match (regex/exact)
- Accurate but fluency suffers
  - fallback steps (no fusion): 28% E2E & 54% WebNLG
  - no reordering possible

	WebNLG		Clean E2E	
system	BLEU	METEOR	BLEU	METEOR
templates	27.7	37.9	20.7	33.4
fusion	35.3	38.6	25.2	33.8
T5 (~SotA)	57.1	44.0	42.1	38.5

(Kale & Rastogi, 2020)
https://www.aclweb.org/anthology/2020.inlg-1.14
https://www.aclweb.org/anthology/2020.emnlp-main.527

Evaluating NLG Accuracy

- n-gram metrics (BLEU, METEOR)
  - derived from MT, no good for accuracy
  - dubious even as measures for overall quality

(Reiter, 2018) https://ehudreiter.com/2018/11/12/hallucination-in-neural-nlg/

(Liu et al., 2016) https://aclanthology.org/D16-1230/
(Novikova et al., 2017) http://aclweb.org/anthology/D17-1238

- Neural metrics (BERTScore, BLEURT) mix accuracy & fluency
  - slightly better than n-gram, but still not ideal

(Zhang et al., 2020) http://arxiv.org/abs/1904.09675
(Sellam et al., 2020) https://aclanthology.org/2020.acl-main.704/

- SER evaluation uses regex or exact match
  - tedious to make / inaccurate
  - does not translate to other datasets
- Proper evaluation means full NLU
  - pretrained models are quite good at NLU-like tasks → use them?

Eval1: Natural Language Inference Classification

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

- NLI task – 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.*

H$_1$: *Blue Spice is located in the riverside.* ⟶ **E**

H$_2$: *You can bring your kids to Blue Spice .* ⟶ **N**

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 (as in iterative editing)
  - needed, unlike summarization / open-domain dialogue

(Maynez et al., 2020)
https://www.aclweb.org/anthology/2020.acl-main.173
(Welleck et al., 2019)
https://www.aclweb.org/anthology/P19-1363

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.*

$\mathbf{H_1}$: Blue Spice is a pub. C: 0.01 N: **0.97** E: 0.02
→ *omission*

$\mathbf{H_2}$: 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

Eval1: NLI Classification

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

- NLI model: **RoBERTa-large-MNLI**, used as-is (no finetuning)
- WebNLG & E2E data
  - comparison vs. human ratings (WebNLG) & SER script (E2E)
  - both datasets: default & backoff-only versions of templates

system	WebNLG 2-way	E2E 4-way	E2E 2-way
Default templates	77.5%	91.1%	93.3%
Backoff template	76.8%	84.6%	87.4%

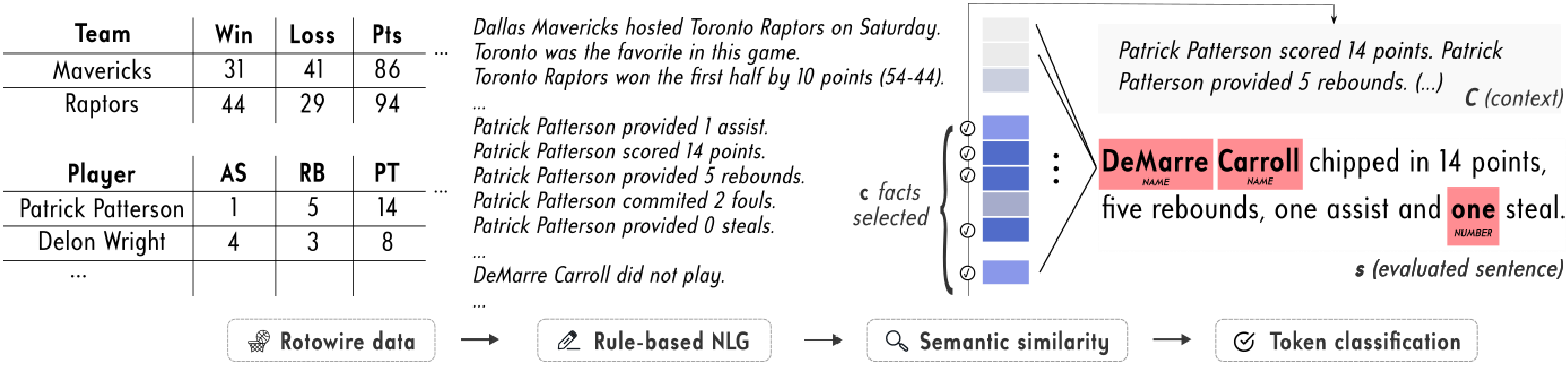
- manual analysis: ca. ½ “errors” are in fact correct
  - annotation noise / SER script errors
  - mined templates noise for WebNLG
  - edge cases (*high restaurant*)
  - irrelevant stuff that SER script doesn’t catch (*with full service*)

Eval2: Token-level Error Detection

(Kasner et al., 2021)
INLG Accuracy Evaluation shared task

- Not just OK/not checks, also identify individual errors
  - good for longer texts – Rotowire basketball summaries
- 3-stage:
  1) **convert input** table into texts (templates / rules) ← whole summary
  2) **select** relevant **context** using SBERT embedding similarity
  3) **tag errors** given context using RoBERTa with token-level classification head

  for each sentence



Eval2: Token-level Error Detection

(Kasner et al., 2021)
INLG Accuracy Evaluation shared task

- Training data:
  - 60 annotated NLG output summaries
  - Synthetic errors introduced into Rotowire training set (3.8k summaries)
    - only random replacement of names & numbers
- Best setup:
  - rule-based generator (more compact contexts)
  - using synthetic data, with 25% errors
  - contexts of 40 sentences (~what fits into RoBERTa)
- Evaluation: 30 annotated summaries
  - best out of 3 systems
  - still lagging behind human evaluation
  - the task is much more difficult than just OK/not OK

team	Recall	Prec
Laval (human eval)	84.1%	87.9%
Charles + UPF	69.1%	75.6%
NIJL	52.3%	49.4%
Eurocom	8.0%	31.1%

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
  - reranking / data cleaning / multi-tasking / editing templates
  - always constraining the neural component
  - there are always downsides
    - lower speed, worse fluency, more annotation needed
- Finding errors in NLG is as hard as NLU
  - pretrained LMs are good at some NLU tasks, such as NLI → can be applied
  - works quite on well sentence-level, token level is hard
- Other interesting areas: data augmentation, few-shot, open domain

Thanks

Contact me:

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

Collaboration with:

Charles University	Vojtěch Hudeček, Filip Jurčíček, Zdeněk Kasner, Jonáš Kulhánek, Tomáš Nekvinda
Heriot-Watt University	David Howcroft, Jekaterina Novikova, Ioannis Konstas, Verena Rieser, Xinnuo Xu
Pompeu Fabra University	Simon Mille