NPFL099 Statistical Dialogue Systems

8. Natural Language Generation

Zdeněk Kasner, Ondřej Dušek, Simone Balloccu, Mateusz Lango,
Ondřej Plátek, Patrícia Schmidtová

http://ufal.cz/npfl099

21.11.2023

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 any other language)

- covers many subtasks:

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

NLG in a narrow sense

NLG Objectives

- general NLG objective:

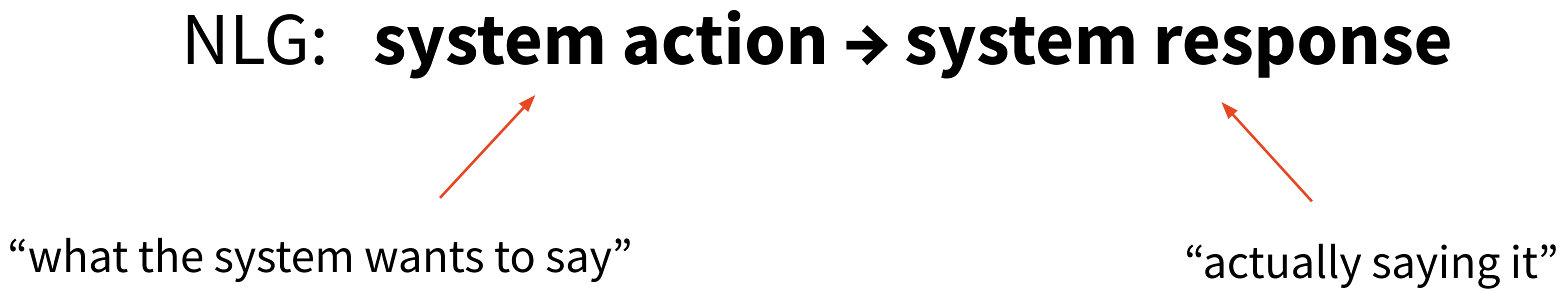
given **input & communication goal**

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 in Dialogue Systems

- in the context of dialogue systems:



NLG: **system action → system response**

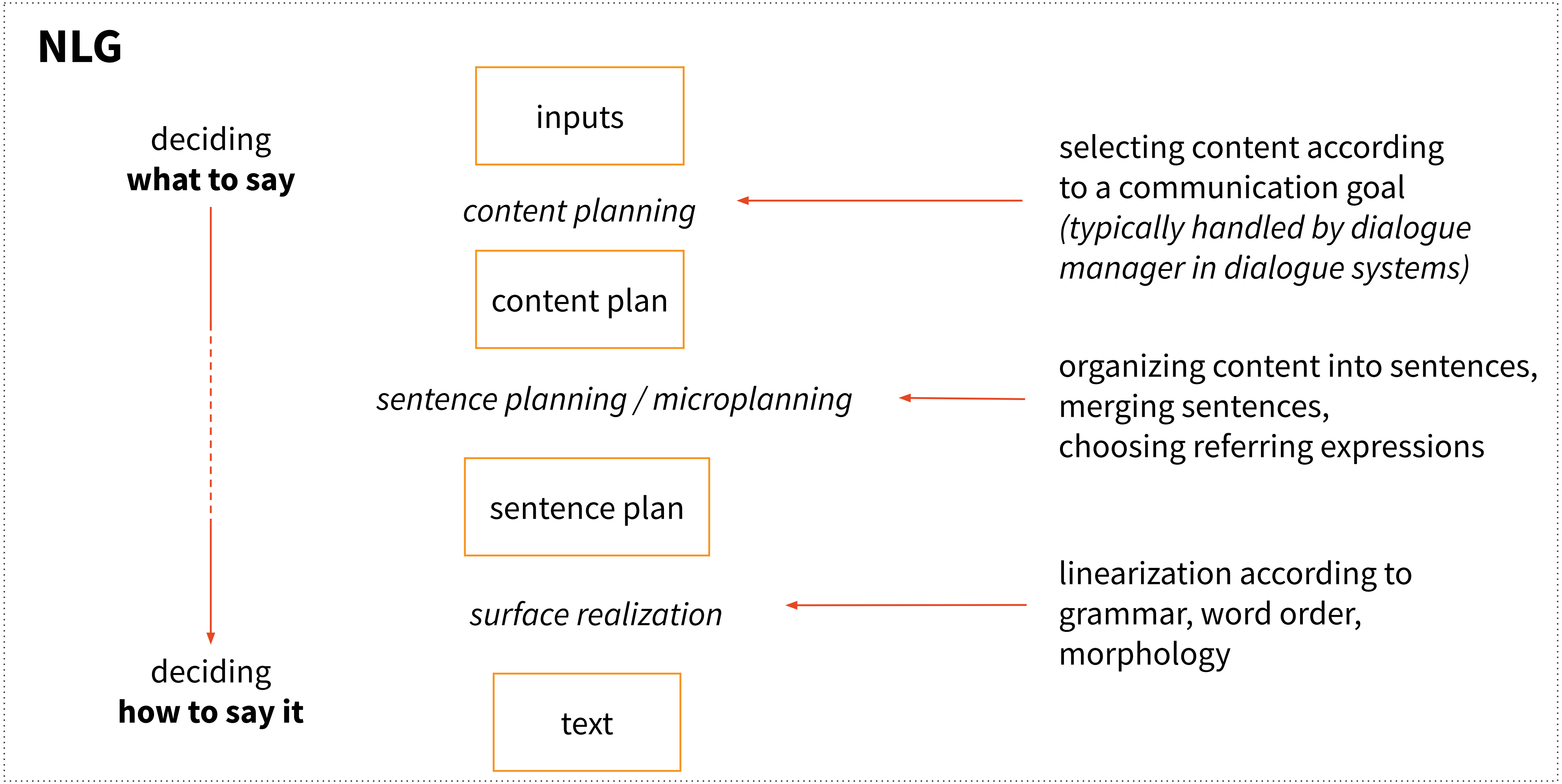
“what the system wants to say”

“actually saying it”

- system action
  - selected by the dialogue manager
  - may be conditioned on:
    - dialogue state
    - dialogue history (→ referring expressions, avoiding repetition)
    - user model (→ “user wants short answers”)

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

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)*

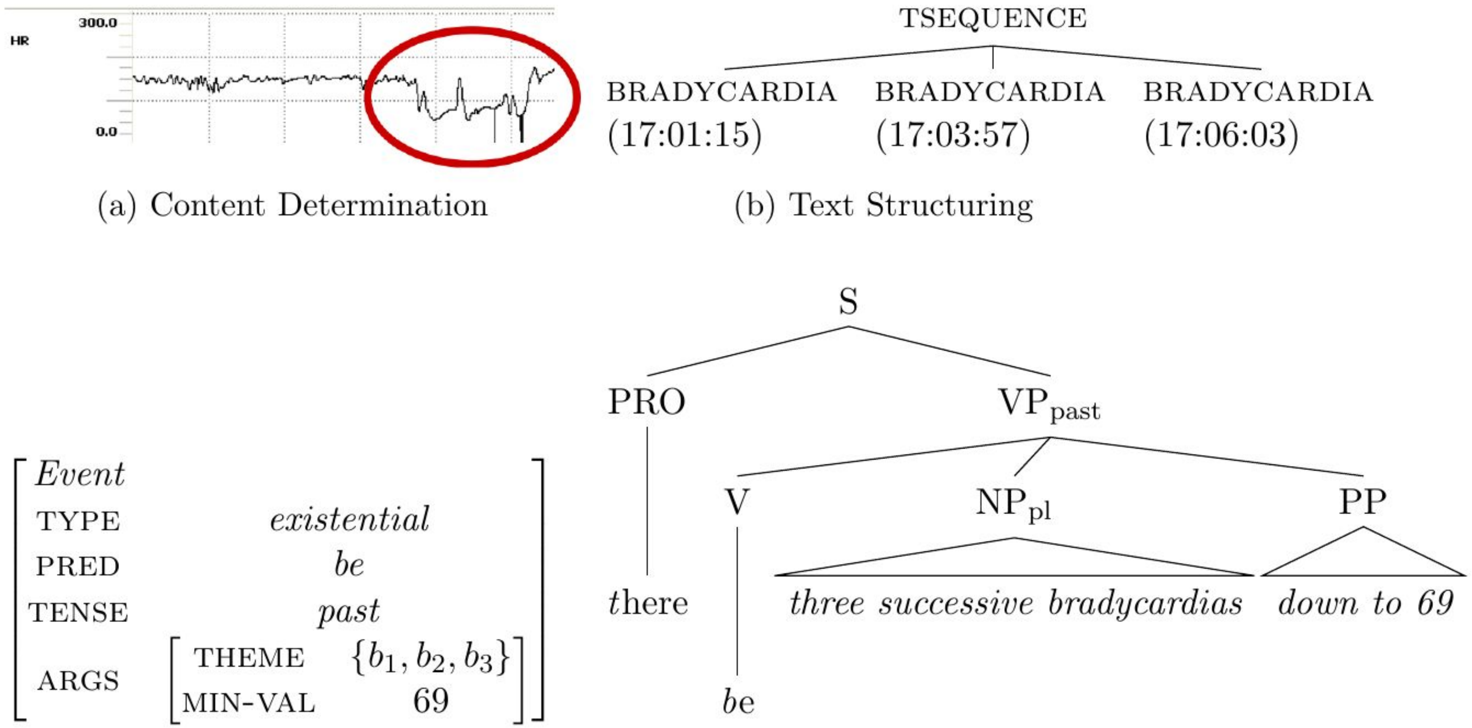
organizing content into sentences,
merging sentences,
choosing referring expressions

linearization according to grammar, word order, morphology

NLG Subtasks (Textbook Pipeline)

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

Example: classical NLG pipeline in medical domain



(a) Content Determination

(b) Text Structuring

(c) Lexicalisation etc.

(d) Realisation

NLG Basic Approaches

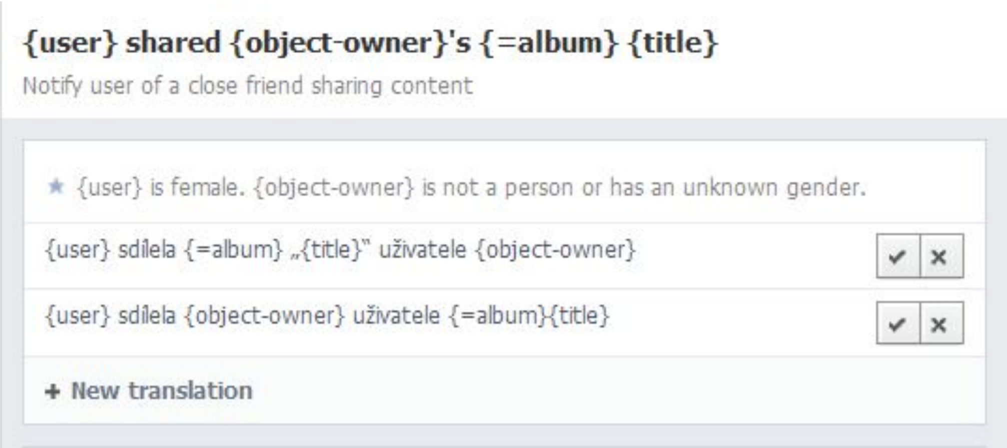
- **hand-written prompts** *("canned text")*
  - most trivial – hard-coded, no variation
  - doesn't scale (good for DTMF phone systems)
- **templates** *("fill in blanks")*
  - simple, but much more expressive – covers most common domains nicely
  - can scale if done right, still laborious
  - most production dialogue systems
- **grammars & rules**
  - grammars: mostly older research systems
  - rules: mostly content & sentence planning
- **machine learning**
  - modern research systems
  - pre-neural attempts often combined with rules/grammar
  - NNs made it work much better

Template-based NLG

- most common in commercial dialogue systems
- simple, straightforward, reliable
  - custom-tailored for the domain
  - complete control of the generated content
- lacks generality and variation
  - difficult to maintain, expensive to scale up
- can be enhanced with rules
  - e.g. articles, inflection of the filled-in phrases
  - template coverage/selection rules (heuristics, random variation)
- can be a good starting point for ML algorithms
  - post-editing / reranking the templates with neural language models

Template-based NLG – Examples

Example: Facebook



{user} shared {object-owner}'s {=album} {title}

Notify user of a close friend sharing content

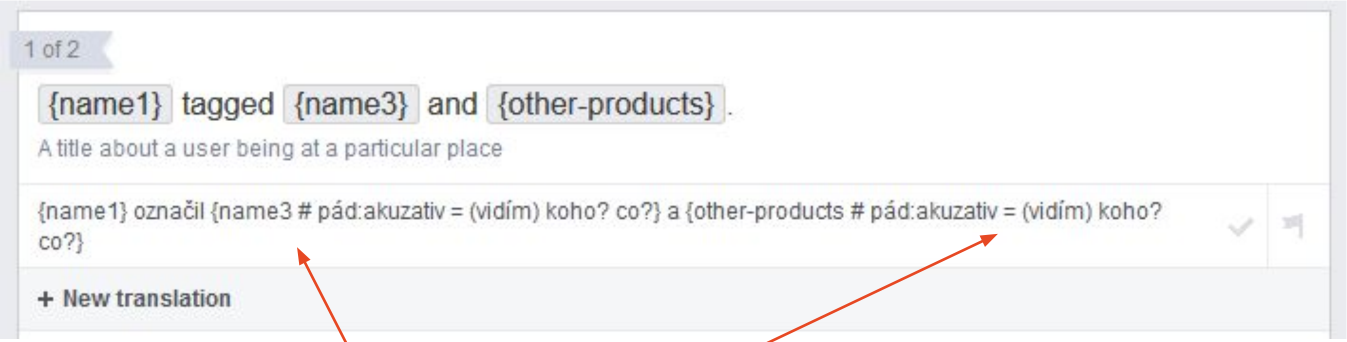
★ {user} is female. {object-owner} is not a person or has an unknown gender.

{user} sdílela {=album} „{title}“ uživatele {object-owner} ✓ ✕

{user} sdílela {object-owner} uživatele {=album}{title} ✓ ✕

+ New translation

(Facebook, 2015)



1 of 2

{name1} tagged {name3} and {other-products}.

A title about a user being at a particular place

{name1} označil {name3 # pád:akuzativ = (vidím) koho? co?} a {other-products # pád:akuzativ = (vidím) koho? co?}

+ New translation

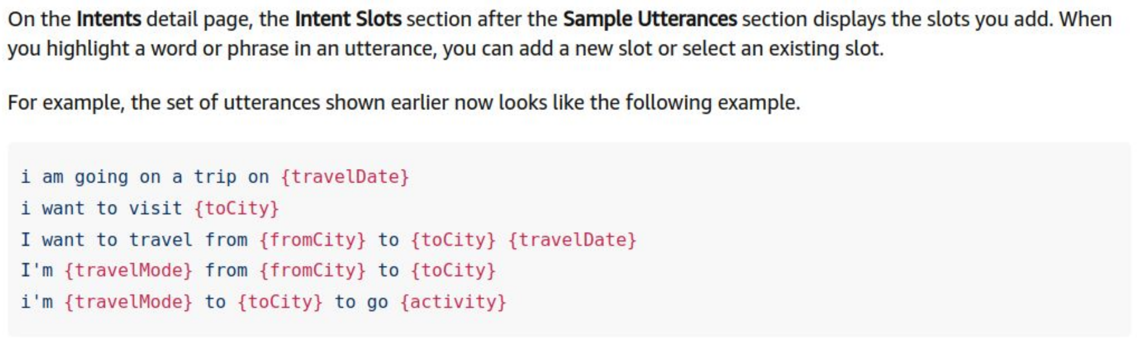
(Facebook, 2019)

inflection rules

Template-based NLG – Examples

Example: Dialogue assistants

Alexa



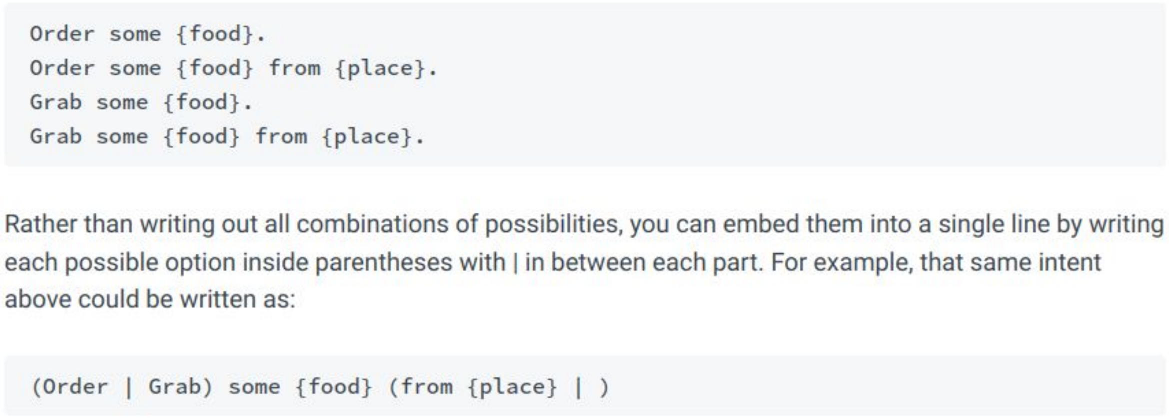
On the **Intents** detail page, the **Intent Slots** section after the **Sample Utterances** section displays the slots you add. When you highlight a word or phrase in an utterance, you can add a new slot or select an existing slot.

For example, the set of utterances shown earlier now looks like the following example.

```
i am going on a trip on {travelDate}
i want to visit {toCity}
I want to travel from {fromCity} to {toCity} {travelDate}
I'm {travelMode} from {fromCity} to {toCity}
i'm {travelMode} to {toCity} to go {activity}
```

(https://developer.amazon.com/en-US/docs/alexa/custom-skills/create-intents-utterances-and-slots.html)

Mycroft



```
Order some {food}.
Order some {food} from {place}.
Grab some {food}.
Grab some {food} from {place}.
```

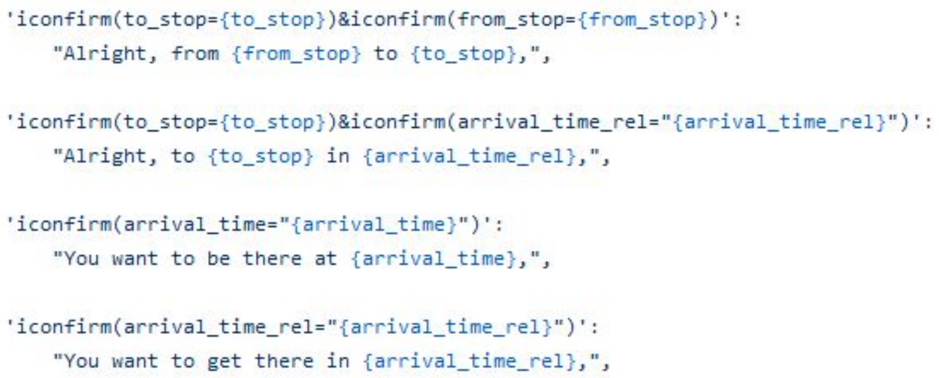
Rather than writing out all combinations of possibilities, you can embed them into a single line by writing each possible option inside parentheses with | in between each part. For example, that same intent above could be written as:

```
(Order | Grab) some {food} (from {place} | )
```

(https://mycroft-ai.gitbook.io/docs/mycroft-technologies/padatious)

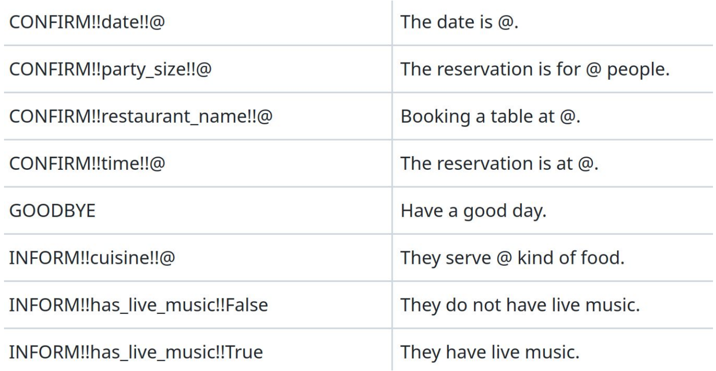
Template-based NLG – Examples

Example: Research systems

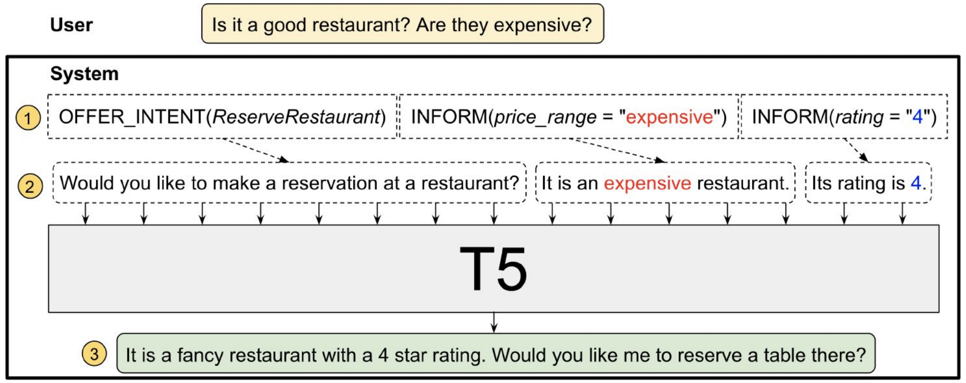


```
'iconfirm(to_stop={to_stop})&iconfirm(from_stop={from_stop})':
    "Alright, from {from_stop} to {to_stop},",

'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



CONFIRM!!date!!@	The date is @.
CONFIRM!!party_size!!@	The reservation is for @ people.
CONFIRM!!restaurant_name!!@	Booking a table at @.
CONFIRM!!time!!@	The reservation is at @.
GOODBYE	Have a good day.
INFORM!!cuisine!!@	They serve @ kind of food.
INFORM!!has_live_music!!False	They do not have live music.
INFORM!!has_live_music!!True	They have live music.

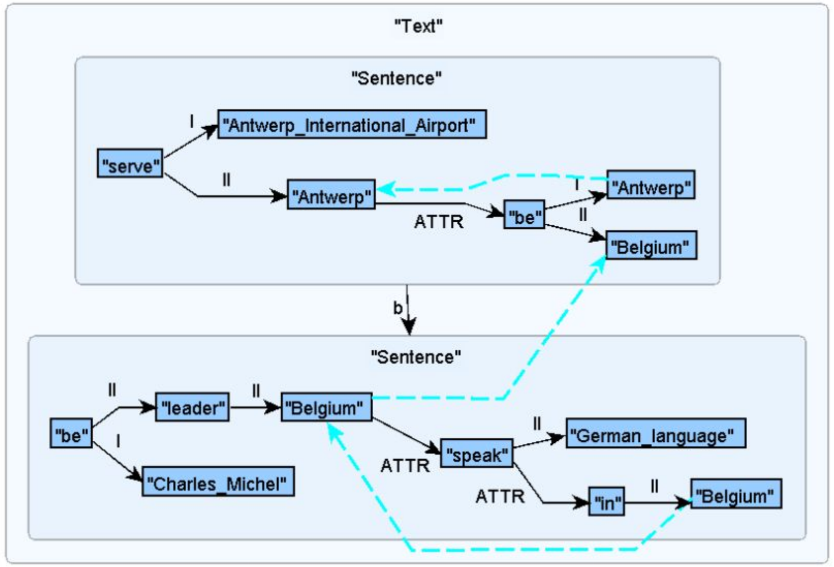


User
Is it a good restaurant? Are they expensive?
System
1
OFFER_INTENT(*ReserveRestaurant*)
INFORM(*price_range* = "expensive")
INFORM(*rating* = "4")
2
Would you like to make a reservation at a restaurant?
It is an expensive restaurant.
Its rating is 4.
T5
3
It is a fancy restaurant with a 4 star rating. Would you like me to reserve a table there?

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

Grammar / Rule-based NLG

- based on top of linguistic theories
- state-of-the-art research systems until NLG the arrival of NNs
- rules for building tree-like structures
  → rules for tree linearization
- reliable, more natural than templates
- takes a lot of effort, naturalness still not human-level
- see NPFL123 for more details



(Mille et al., 2019)
https://aclanthology.org/W19-8659.pdf

Neural NLG

- learning the task from the data
- sequence-to-sequence generation / editing / re-ranking
- fluency can match human-level, minimal hand-crafting
- not controllable (“black-box”),
  semantic inaccuracies (omissions / hallucinations),
  low diversity,
  expensive data gathering,
  expensive training,
  expensive deployment

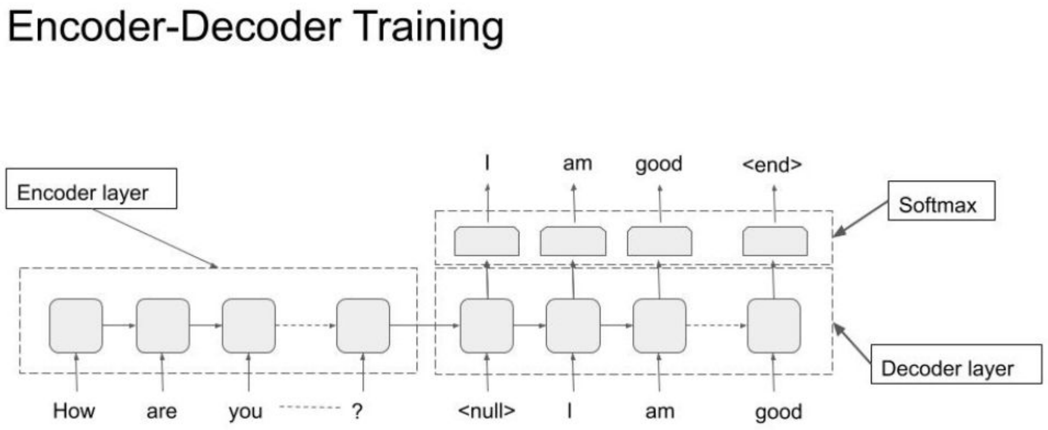
  → promising research area 😉
- getting better with larger models

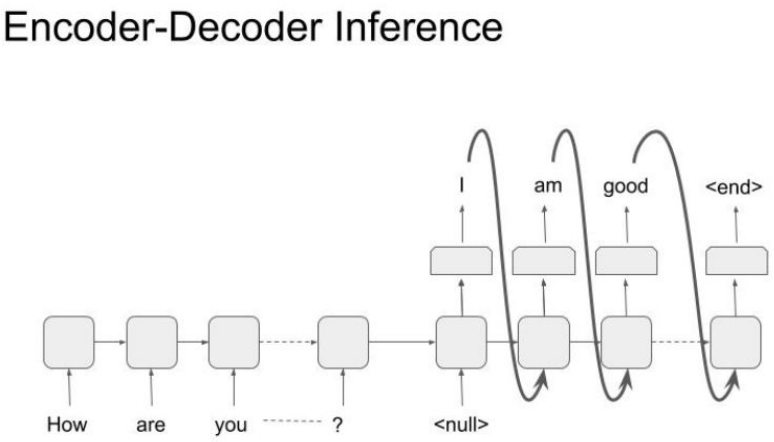
Seq2seq Generation

https://slds-lmu.github.io/seminar_nlp_ss20/natural-language-generation.html

- **encoder-decoder**
  - *RNN:* encoder updates the hidden state → decoder is initialized with the hidden state
  - *Transformer*: encoder generates a sequence of hidden states → decoder attends to this sequence
  - pretrained Transformers (PLMs): BART, T5 (trained on sequence denoising)
- **decoder-only**
  - input sequence is prepended as a context, the decoder generates continuation
  - PLMs: GPT-2, GPT-3 (trained on autoregressive language modelling)

- training vs. inference:





Decoding Algorithms

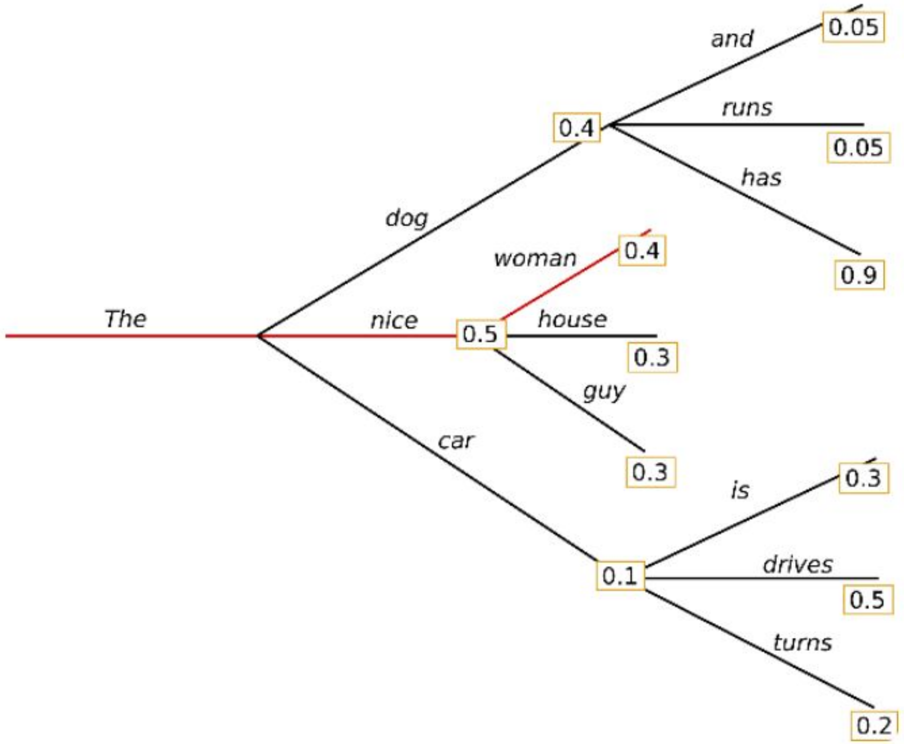
- for each time step *t*, the decoder outputs a probability distribution: $P(y_t | y_{1:t-1,} \mathbf{X})$
- how to use it?
- **exact inference:** find a sequence maximizing $P(y_{1:T} | \mathbf{X})$
  - not possible in practice (why? and is it our goal?)
- **approximation algorithms**
  - greedy search
  - beam search
- **stochastic algorithms**
  - random sampling
  - top-k sampling
  - nucleus sampling (=top-p sampling)

(+ repetition penalty → decreasing probabilities of generated tokens)

Decoding Algorithms

https://huggingface.co/blog/how-to-generate
https://towardsdatascience.com/decoding-strategies-that-you-need-to-know-for-response-generation-ba95ee0faadc

- **greedy search:** always take the argmax
  - does not necessarily produce the most probable sequence (why?)
  - often produces dull responses



Example:

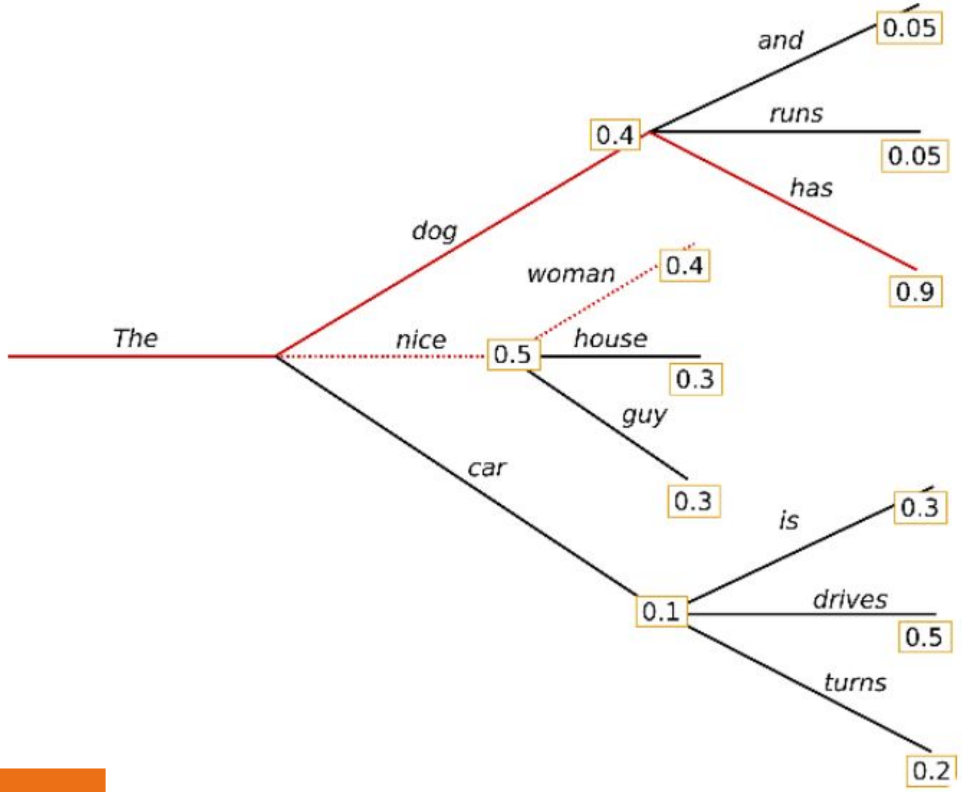
Context:	Try this cake. I baked it myself.
Optimal Response :	This cake tastes great.
Greedy search:	This is okay.

many examples start with “This is”,
no possibility to backtrack

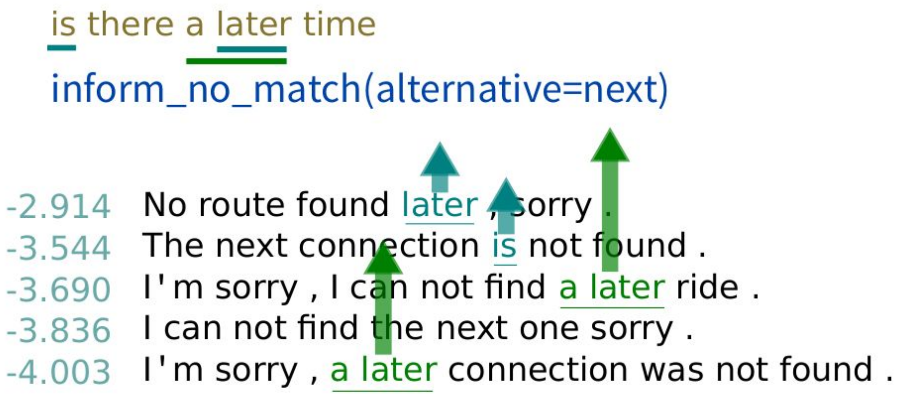
Decoding Algorithms

https://huggingface.co/blog/how-to-generate

- **beam search:** try *k* continuations of *k* hypotheses, keep *k* best
  - better approximation of the most probable sequence, bounded memory & time
  - allows re-ranking generated outputs
  - k=1 → greedy search



Reranking:


is there a later time

inform_no_match(alternative=next)

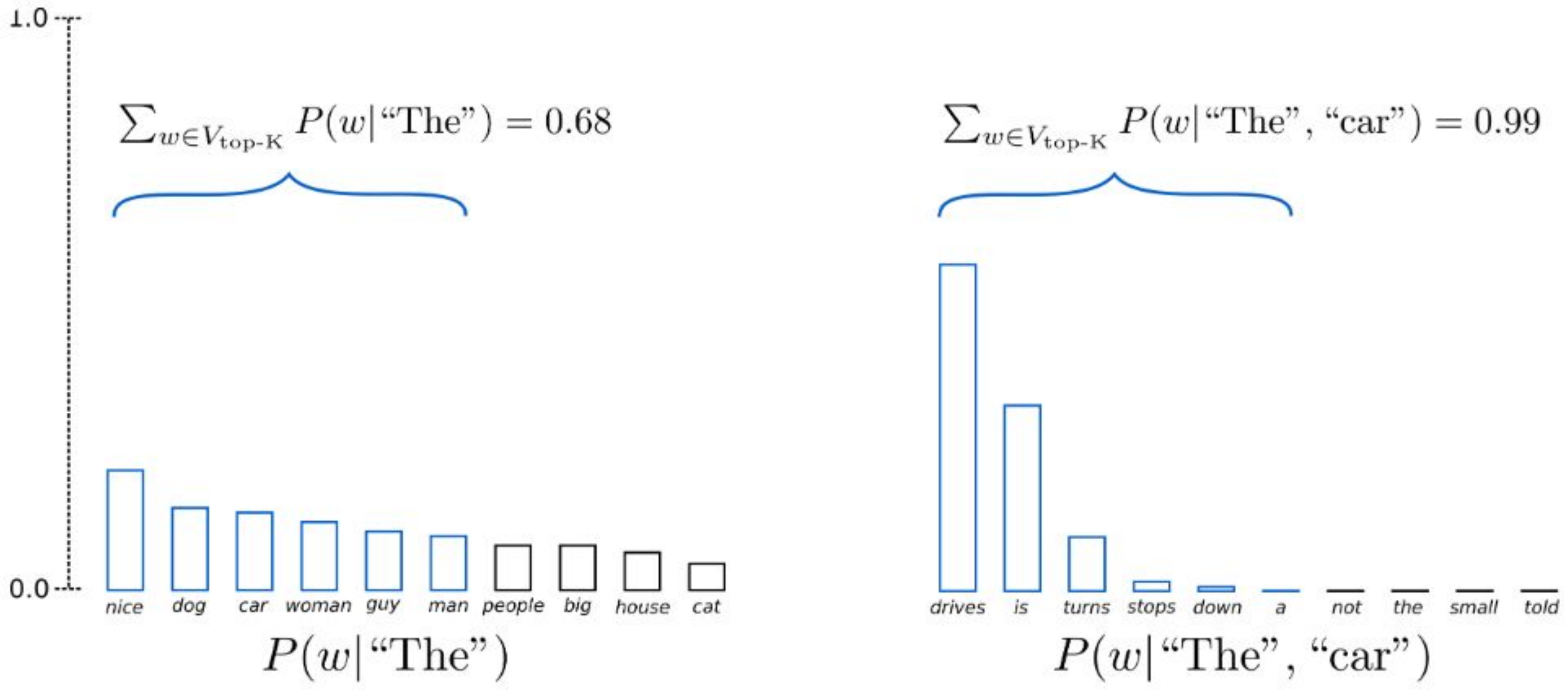
```
-2.914  No route found later , sorry .
-3.544  The next connection is not found .
-3.690  I'm sorry , I can not find a later ride .
-3.836  I can not find the next one sorry .
-4.003  I'm sorry , a later connection was not found .
```

(Ondřej's PhD thesis, Fig. 7.7)
http://ufal.mff.cuni.cz/~odusek/2017/docs/thesis.print.pdf

Decoding Algorithms

https://huggingface.co/blog/how-to-generate

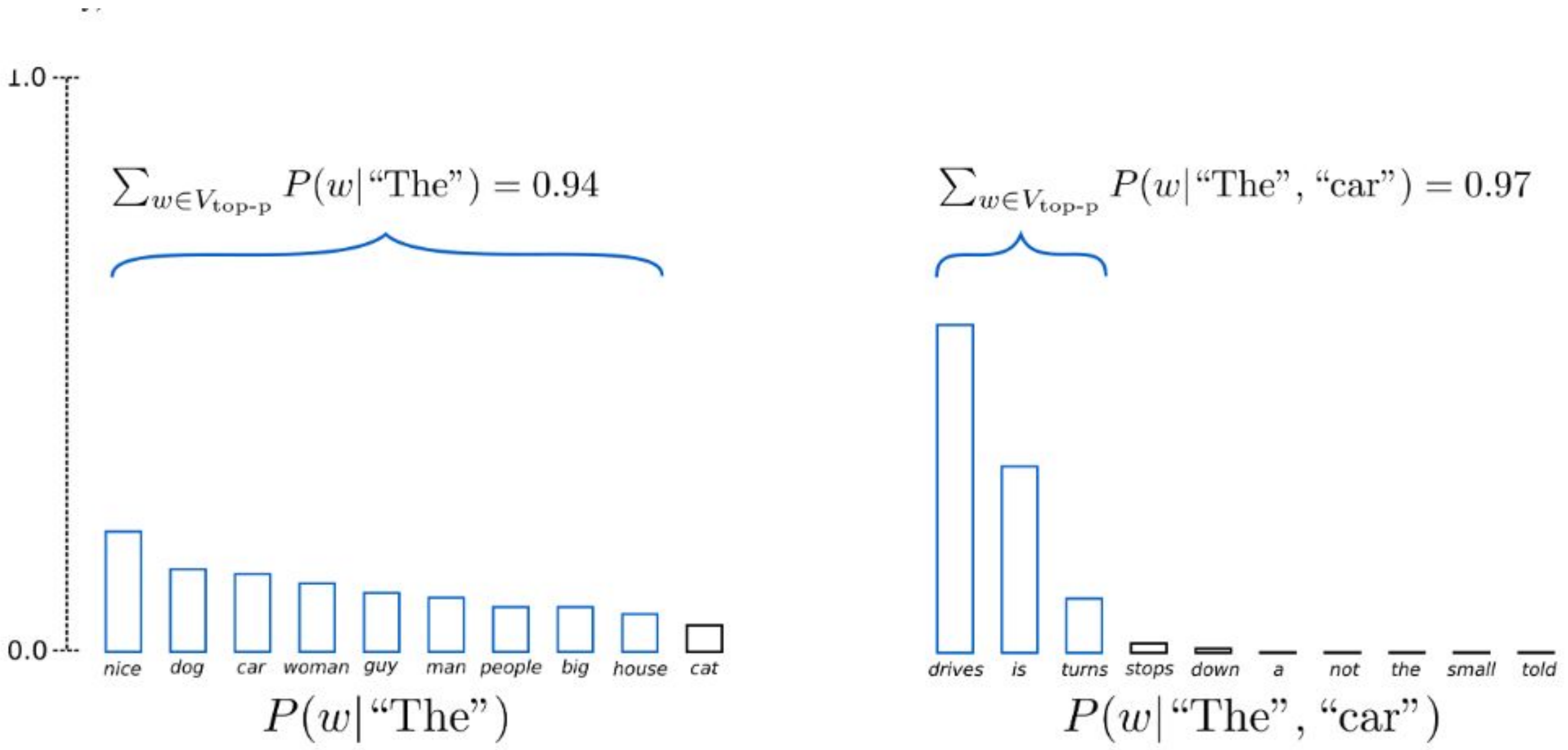
- **top-k sampling:** choose top k options (~5-500), sample from them
    - avoids the long tail of the distribution
    - more diverse outputs



Decoding Algorithms

https://huggingface.co/blog/how-to-generate

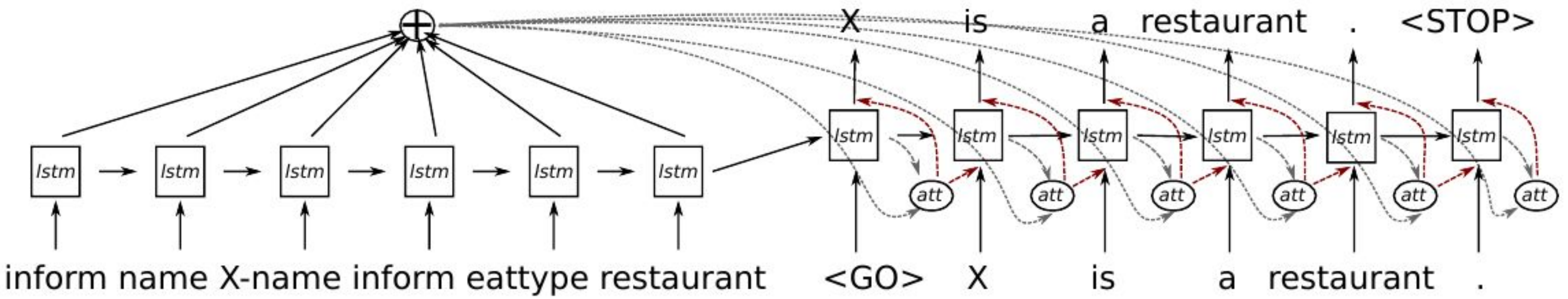
- **nucleus sampling:** choose top options that cover >= *p* probability mass (~0.9)
  - “*k*” is adapted according to the distribution shape



RNN-based Approaches

- first neural approaches: ~2015
- TGen: standard LSTM with attention
  - encoder – triples <intent, slot, value>
  - decodes words (possibly delexicalized)
  - beam search & reranking

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

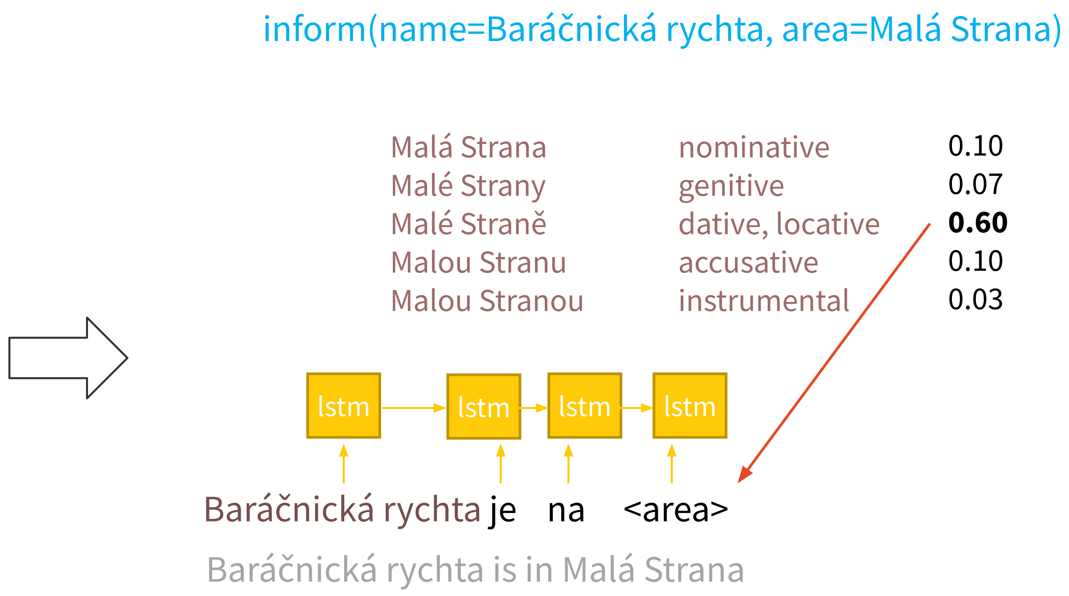


- RNNLM
  - using special LSTM gate cells (SC-LSTM) to control slot mentions

(Wen et al, 2015; 2016)
http://aclweb.org/anthology/D15-1199
http://arxiv.org/abs/1603.01232

Delexicalization Alternatives

- **copy mechanism** (see NLU & the next slide)
  - select (or interpolate) between:
    - generating a new token
    - copying a token from input
  - removes the need for pre/postprocessing
- **inflection model**
  - useful for languages with rich morphology (e.g., Czech)
  - a simple LM such as RNN LM
- **pretrained models**
  - the model learns to copy and inflect words implicitly during pretraining
  - works well for high-resource languages

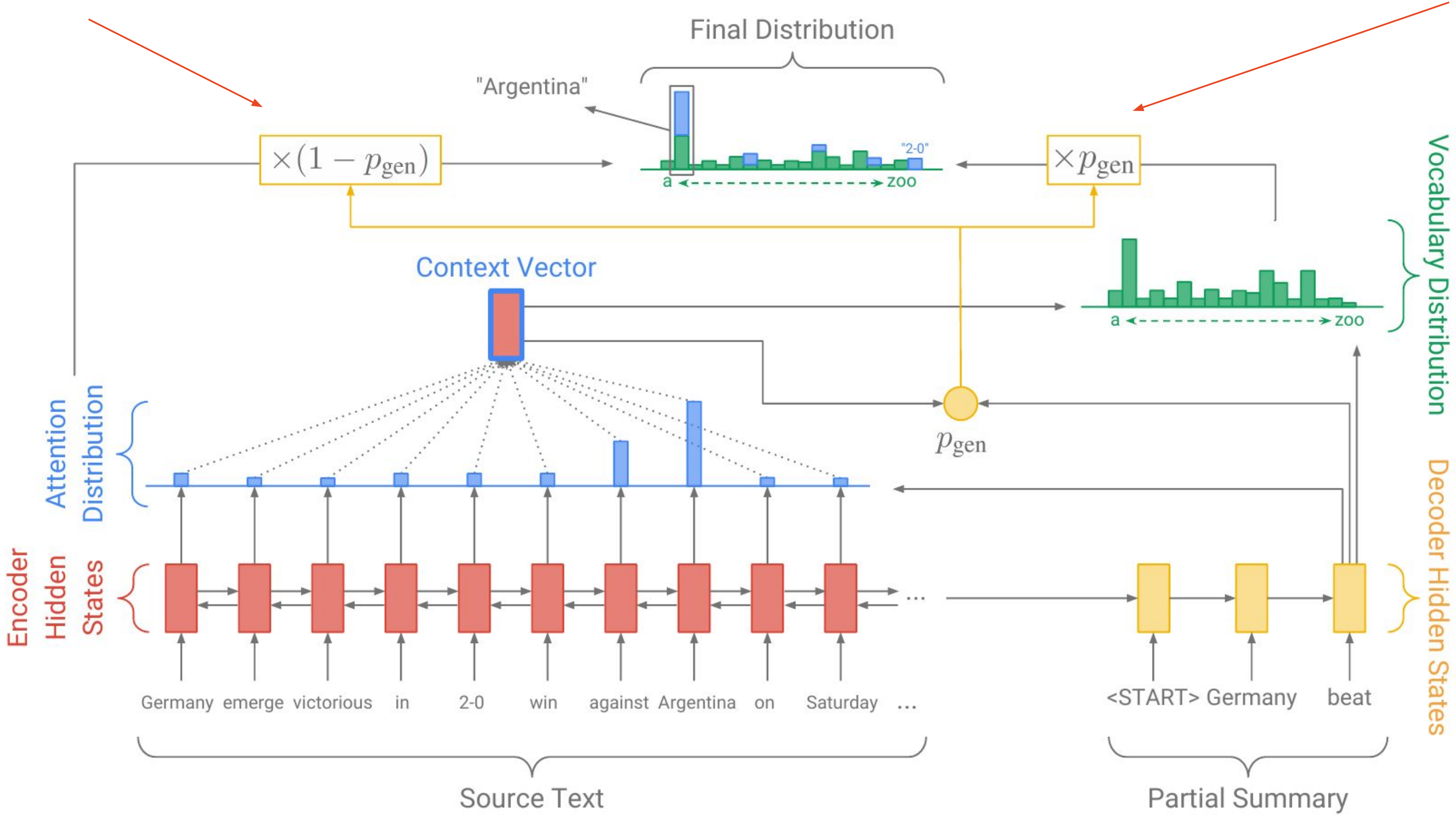


(Dušek & Jurčíček, 2019)
https://arxiv.org/abs/1910.05298

Delexicalization Alternatives – Copy Mechanism

probability of copying a token from the input

probability of generating a new token from the vocabulary



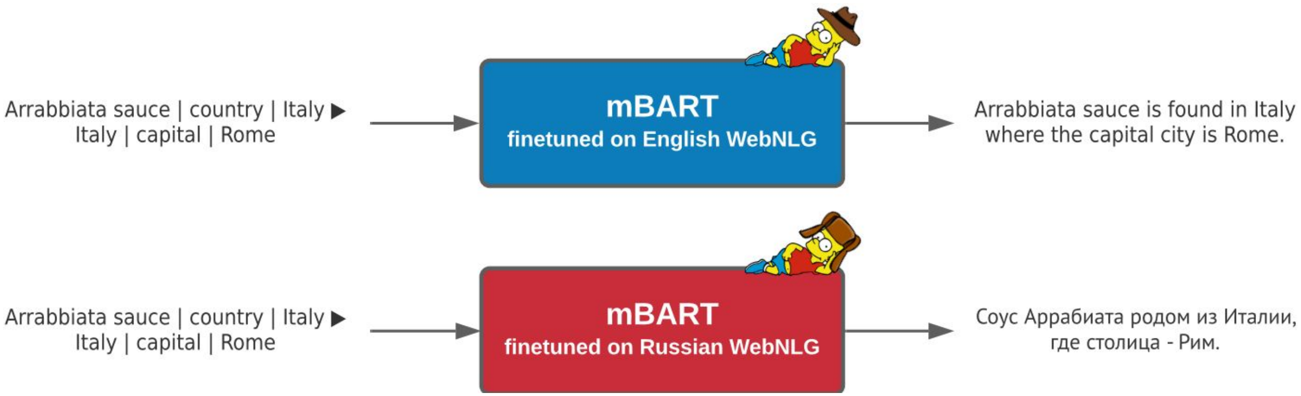
(See et al., 2017)
http://arxiv.org/abs/1704.04368

Finetuning PLMs

- GPT-2 or BART / T5 finetuned for NLG
  - different pretraining tasks – similar outcomes
- works nicely when simply finetuned for data-to-text
  - encode linearized data, decode text
  - the model learns copying implicitly
- mBART / mT5 (multilingual) → allows multilingual generation
  - can generate e.g. Russian outputs from English triples
- are we done now?

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

(Liu et al., 2020) http://arxiv.org/abs/2001.08210


Arrabbiata sauce | country | Italy ▶ Italy | capital | Rome
mBART
finetuned on English WebNLG
Arrabbiata sauce is found in Italy where the capital city is Rome.
Arrabbiata sauce | country | Italy ▶ Italy | capital | Rome
mBART
finetuned on Russian WebNLG
Соус Аррабиата родом из Италии, где столица - Рим.

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

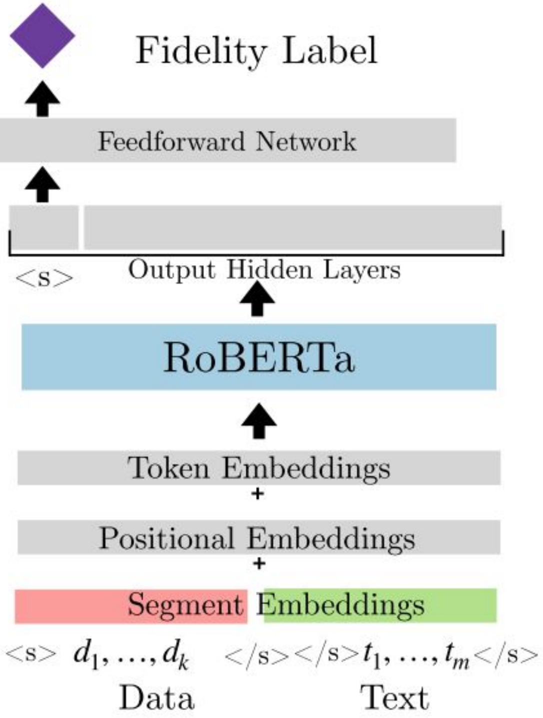
Finetuning PLMs with Reranking

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

- **goal: improving semantic accuracy**
- seq2seq + reranking with GPT-2 & RoBERTa
- GPT-2 fine-tuned for *<data> name[Zizzi] eatType[bar]* *<text> Zizzi is a bar .*

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

- beam search decoding
- RoBERTa for classification
  - accurate/omission/repetition/hallucination/value error
  - training data synthesized
    - “accurate” examples from original training data
    - others created by manipulating the data and texts (adding/removing/replacing sentences and/or data items)

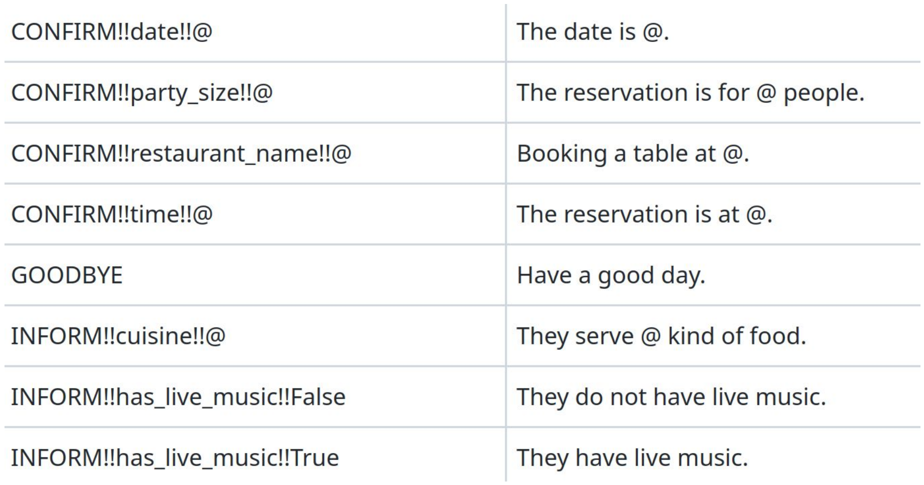


PLMs + Templates

- combining advantages of templates (controllability) and PLMs (fluency)
- concatenate simple templates and then use pretrained LMs (e.g. T5/BART) to rephrase them
  - basically text-to-text denoising, i.e. what the models were originally trained to do
- needs less data & generalizes to new domains

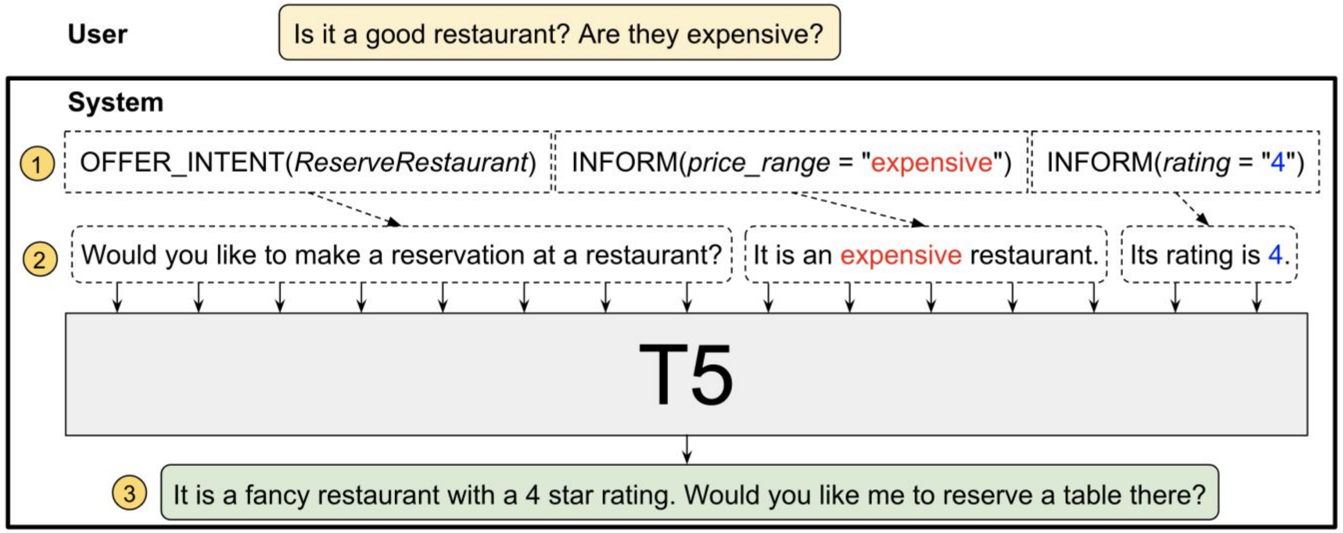
(Kale & Rastogi, 2020)
https://www.aclweb.org/anthology/2020.emnlp-main.527

templates



CONFIRM!!date!!@	The date is @.
CONFIRM!!party_size!!@	The reservation is for @ people.
CONFIRM!!restaurant_name!!@	Booking a table at @.
CONFIRM!!time!!@	The reservation is at @.
GOODBYE	Have a good day.
INFORM!!cuisine!!@	They serve @ kind of food.
INFORM!!has_live_music!!False	They do not have live music.
INFORM!!has_live_music!!True	They have live music.

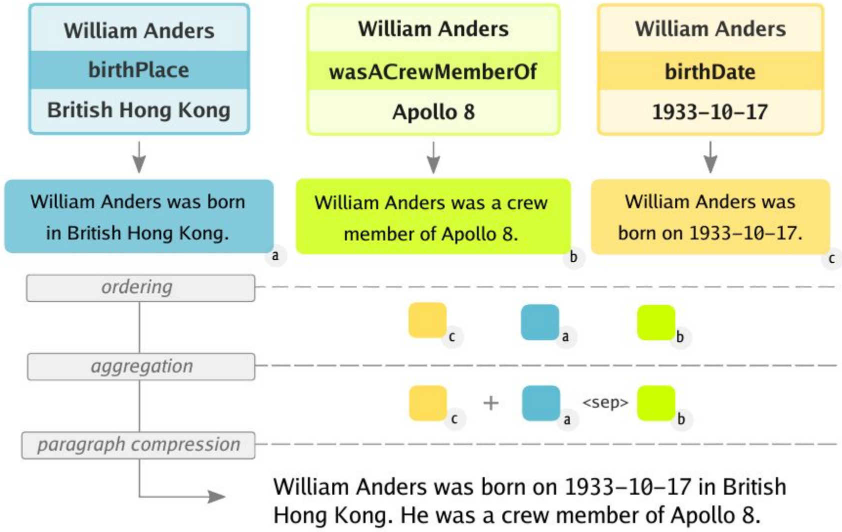
system



User
Is it a good restaurant? Are they expensive?
System
1
OFFER_INTENT(ReserveRestaurant)
INFORM(price_range = "expensive")
INFORM(rating = "4")
2
Would you like to make a reservation at a restaurant?
It is an expensive restaurant.
Its rating is 4.
T5
3
It is a fancy restaurant with a 4 star rating. Would you like me to reserve a table there?

PLMs + Templates

- **data-to-text NLG without human-written references**
- start with templates → postprocess them with BART-based models
  - trained for text-based operations learned from Wikipedia


William Anders
birthPlace
British Hong Kong
William Anders
wasACrewMemberOf
Apollo 8
William Anders
birthDate
1933–10–17
William Anders was born in British Hong Kong.
a
William Anders was a crew member of Apollo 8.
b
William Anders was born on 1933–10–17.
c
ordering
c
a
b
aggregation
c
+
a
<sep>
b
paragraph compression
William Anders was born on 1933–10–17 in British Hong Kong. He was a crew member of Apollo 8.

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

- improvement: using prompted GPT-3 instead of hand-crafted templates

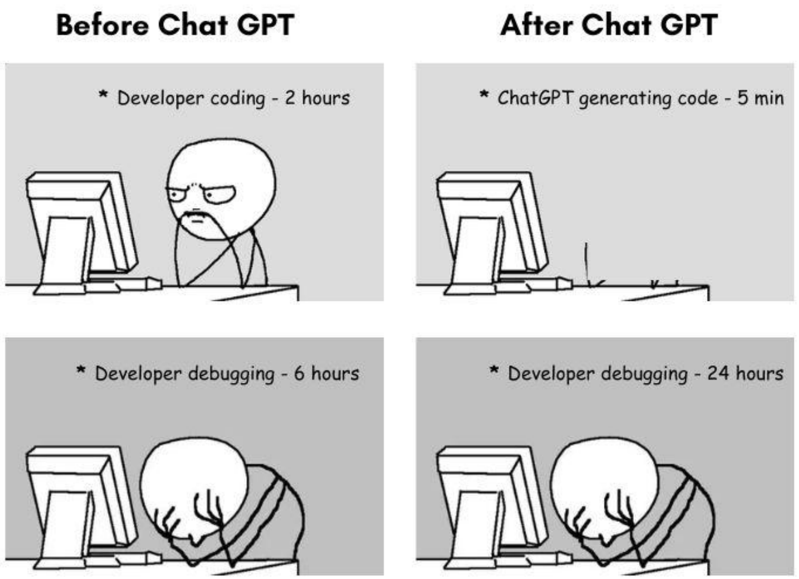
(Xiang et al., 2022)
http://arxiv.org/abs/2210.04325

Prompting LLMs

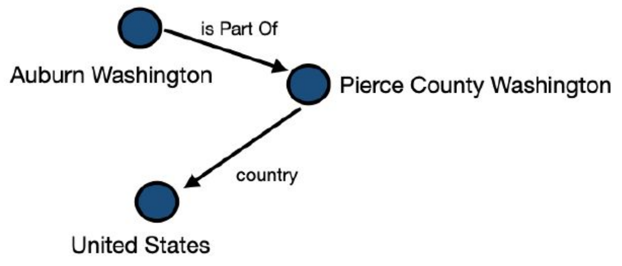
(Axelsson and Skantze, 2023)
http://arxiv.org/abs/2307.07312

- **direct instructions instead of task-specific finetuning** (see Lecture 4, slide 21)
- works only with very large models (about >1B par.)
- ChatGPT, GPT-4 → best performance, but many issues (replicability, cost, data contamination, ...)
- preliminary results for NLG: prompting competitive to finetuning, but different kinds of problems:
  - variability in responses ("Here is the answer: (...)", "As an AI language model (...)"
  - prompt sensitivity
  - hallucinations
- for NLG in dialog, overgenerate-and-rerank still helps

(Ramirez et al., 2023)
https://aclanthology.org/2023.sigdial-1.32


Before Chat GPT
After Chat GPT
* Developer coding - 2 hours
* ChatGPT generating code - 5 min
* Developer debugging - 6 hours
* Developer debugging - 24 hours

https://www.boredpanda.com/chatgpt-memes/



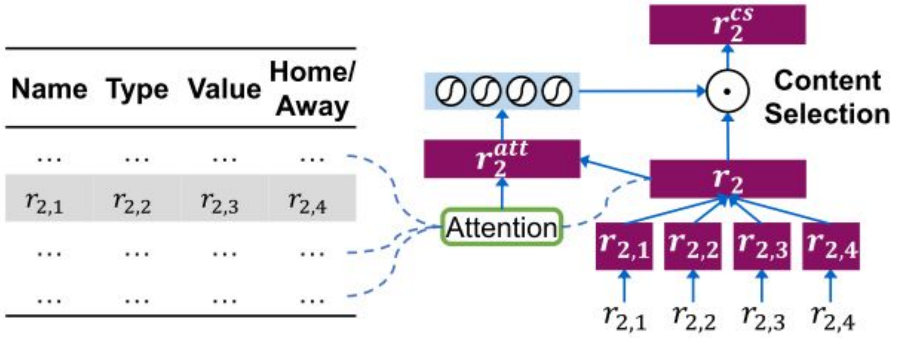
(b) Generate text from graph: **<H>** Auburn Washington **<R>** is Part Of **<T>** Pierce County Washington **<H>** Pierce County Washington **<R>** country **<T>** United States

(Yuan and Färber, 2023)
http://arxiv.org/abs/2307.14712

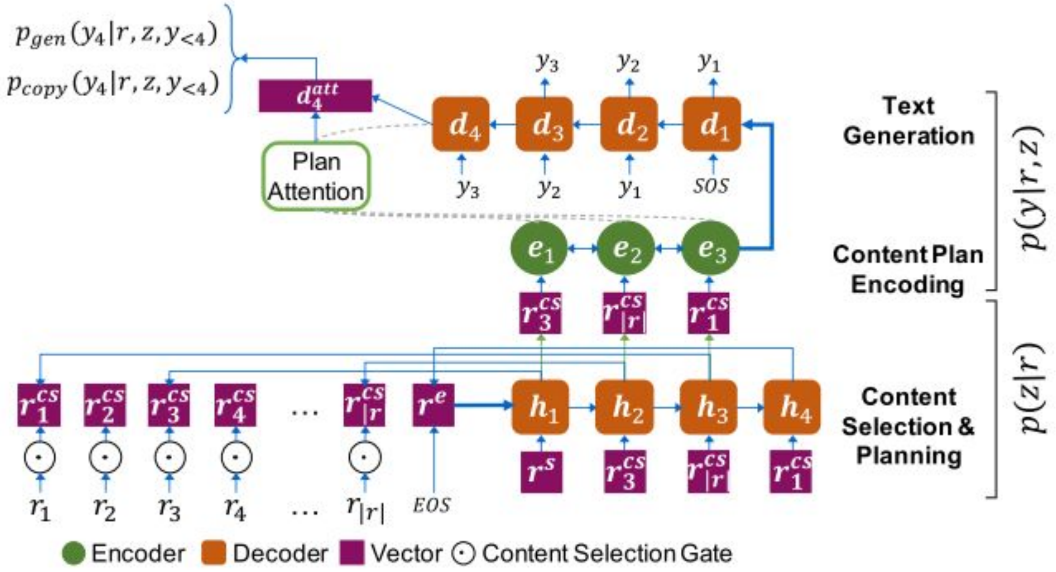
Content Planning: Content Selection

- **explicit content selection**
- usually done by DM in dialogue systems
- needed for complex inputs, e.g. sports report generation
  - records (team / entity / type / value) → summary
  - content selection: pointer network
- still largely unsolved problem w.r.t. semantic accuracy

(Thomson & Reiter, 2022)
http://arxiv.org/abs/2108.05644



(Puduppully et al., 2019)
http://arxiv.org/abs/1809.00582

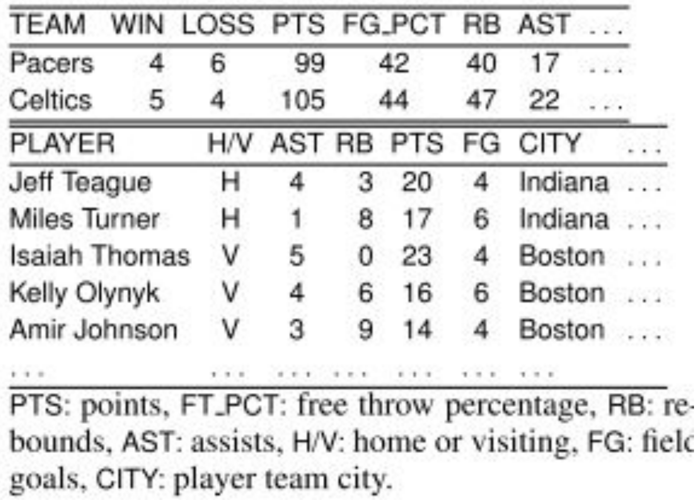


$p_{gen}(y_4|r,z,y_{<4})$
$p_{copy}(y_4|r,z,y_{<4})$
d_4^{att}
Plan Attention
d_4 d_3 d_2 d_1
y_3 y_2 y_1
y_3 y_2 y_1 SOS
Text Generation
e_1 e_2 e_3
r_3^{cs} $r_{|r|}^{cs}$ r_1^{cs}
Content Plan Encoding
$p(y|r,z)$
r_1^{cs} r_2^{cs} r_3^{cs} r_4^{cs} … $r_{|r|}^{cs}$ r^e
h_1 h_2 h_3 h_4
r^s r_3^{cs} $r_{|r|}^{cs}$ r_1^{cs}
r_1 r_2 r_3 r_4 … $r_{|r|}$ EOS
Content Selection & Planning
$p(z|r)$
● Encoder ■ Decoder ■ Vector ⊙ Content Selection Gate

Content Planning: Content Selection

Example of NLG with content planning

source statistics (excerpt)

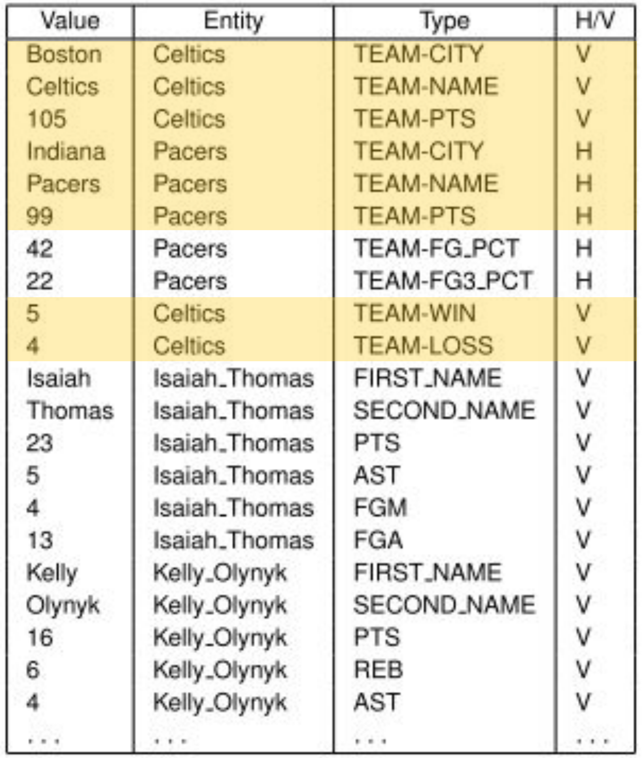


TEAM	WIN	LOSS	PTS	FG_PCT	RB	AST	...
Pacers	4	6	99	42	40	17	...
Celtics	5	4	105	44	47	22	...

PLAYER	H/V	AST	RB	PTS	FG	CITY	...
Jeff Teague	H	4	3	20	4	Indiana	...
Miles Turner	H	1	8	17	6	Indiana	...
Isaiah Thomas	V	5	0	23	4	Boston	...
Kelly Olynyk	V	4	6	16	6	Boston	...
Amir Johnson	V	3	9	14	4	Boston	...
...	...	...	...	...	...	...	

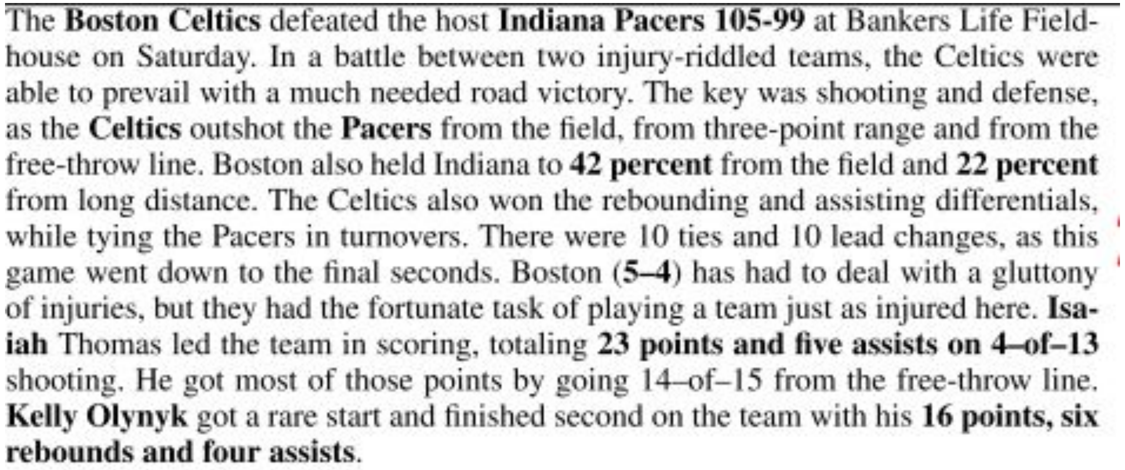
PTS: points, FT_PCT: free throw percentage, RB: rebounds, AST: assists, H/V: home or visiting, FG: fielc goals, CITY: player team city.

content plan (for the 1st sentence)



Value	Entity	Type	H/V
Boston	Celtics	TEAM-CITY	V
Celtics	Celtics	TEAM-NAME	V
105	Celtics	TEAM-PTS	V
Indiana	Pacers	TEAM-CITY	H
Pacers	Pacers	TEAM-NAME	H
99	Pacers	TEAM-PTS	H
42	Pacers	TEAM-FG_PCT	H
22	Pacers	TEAM-FG3_PCT	H
5	Celtics	TEAM-WIN	V
4	Celtics	TEAM-LOSS	V
Isaiah	Isaiah_Thomas	FIRST_NAME	V
Thomas	Isaiah_Thomas	SECOND_NAME	V
23	Isaiah_Thomas	PTS	V
5	Isaiah_Thomas	AST	V
4	Isaiah_Thomas	FGM	V
13	Isaiah_Thomas	FGA	V
Kelly	Kelly_Olynyk	FIRST_NAME	V
Olynyk	Kelly_Olynyk	SECOND_NAME	V
16	Kelly_Olynyk	PTS	V
6	Kelly_Olynyk	REB	V
4	Kelly_Olynyk	AST	V
...	...	...	...

target text

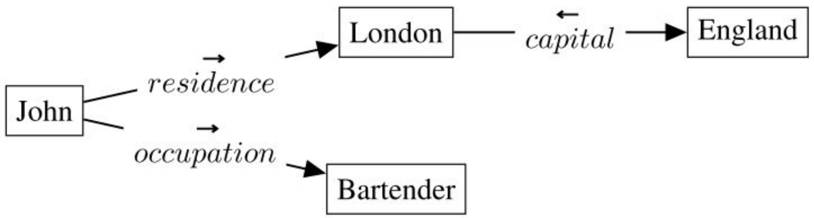


The **Boston Celtics** defeated the host **Indiana Pacers 105-99** at Bankers Life Fieldhouse on Saturday. In a battle between two injury-riddled teams, the Celtics were able to prevail with a much needed road victory. The key was shooting and defense, as the **Celtics** outshot the **Pacers** from the field, from three-point range and from the free-throw line. Boston also held Indiana to **42 percent** from the field and **22 percent** from long distance. The Celtics also won the rebounding and assisting differentials, while tying the Pacers in turnovers. There were 10 ties and 10 lead changes, as this game went down to the final seconds. Boston (**5–4**) has had to deal with a gluttony of injuries, but they had the fortunate task of playing a team just as injured here. **Isaiah** Thomas led the team in scoring, totaling **23 points and five assists on 4–of–13** shooting. He got most of those points by going 14–of–15 from the free-throw line. **Kelly Olynyk** got a rare start and finished second on the team with his **16 points, six rebounds and four assists**.

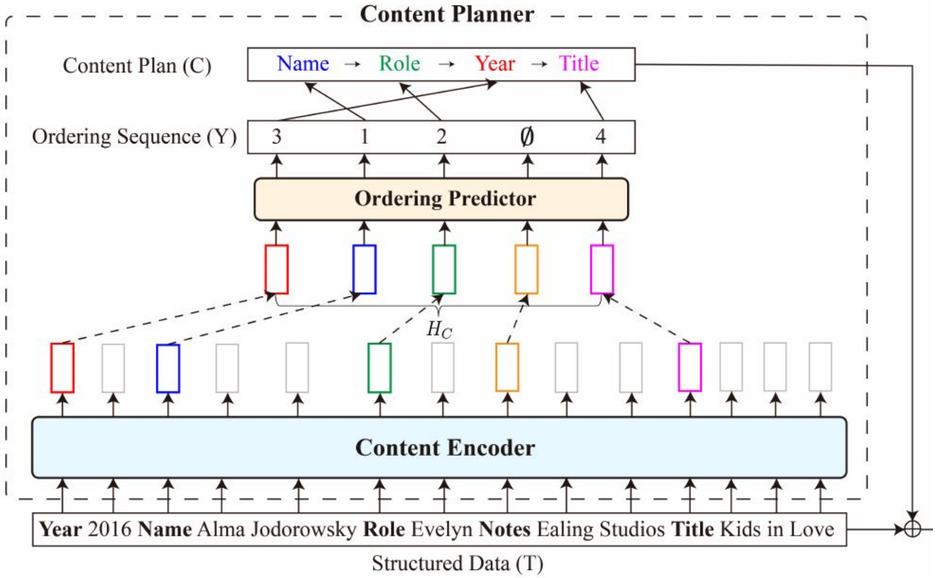
(Puduppully et al., 2019)
http://arxiv.org/abs/1809.00582

Content Planning: Ordering & Aggregation

- **ordering the facts + aggregating them into sentences**
- content already selected at this point
- can help the generator not to miss any facts
- for graphs with oriented edges:
  - generating all possible content plans using DFS (possibly pruning unpromising branches) → re-ranking the plans using a feature-based classifier
- for a set of key-value pairs:
  - using Conditional Random Field (CRF) for finding the optimal plan


John
$\overrightarrow{residence}$
$\overrightarrow{occupation}$
London
$\overleftarrow{capital}$
England
Bartender

(Moryossef et al., 2019a,b)
http://arxiv.org/abs/1904.03396
https://arxiv.org/pdf/1909.09986.pdf



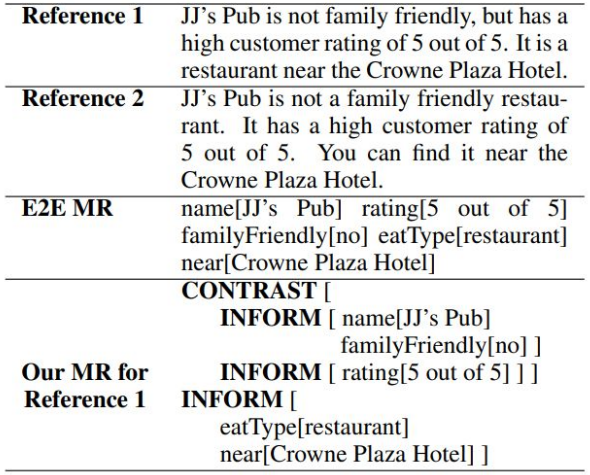
(Su et al., 2020)
http://arxiv.org/abs/2108.13740

Realizing from Trees

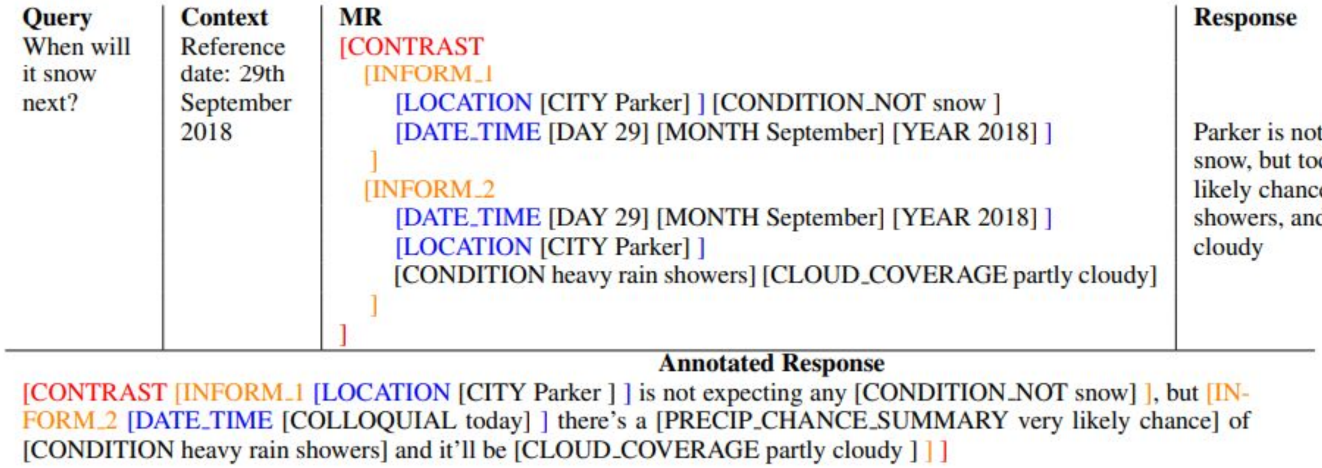
(Balakrishnan et al., 2019) http://arxiv.org/abs/1906.07220
(Rao et al., 2019) https://www.aclweb.org/anthology/W19-8611/
(Li et al., 2021) https://aclanthology.org/2021.inlg-1.10

seq2seq + copy | seq gen

- **NLG with tree-shaped inputs**
- simple case: discourse relations (discourse connectives, sentence splits) between individual fields
  - much flatter than usual syntactic trees
- improvements to account for the input structure:
  - constrained beam search decoding, tree-LSTM, self-training on synthetic data



Reference 1	JJ's Pub is not family friendly, but has a high customer rating of 5 out of 5. It is a restaurant near the Crowne Plaza Hotel.
Reference 2	JJ's Pub is not a family friendly restaurant. It has a high customer rating of 5 out of 5. You can find it near the Crowne Plaza Hotel.
E2E MR	name[JJ's Pub] rating[5 out of 5] familyFriendly[no] eatType[restaurant] near[Crowne Plaza Hotel]
Our MR for Reference 1	**CONTRAST** [**INFORM** [name[JJ's Pub] familyFriendly[no]] **INFORM** [rating[5 out of 5]]] **INFORM** [eatType[restaurant] near[Crowne Plaza Hotel]]



Query	**Context**	**MR**	**Response**
When will it snow next?	Reference date: 29th September 2018	[CONTRAST [INFORM_1 [LOCATION [CITY Parker]] [CONDITION_NOT snow] [DATE_TIME [DAY 29] [MONTH September] [YEAR 2018]]] [INFORM_2 [DATE_TIME [DAY 29] [MONTH September] [YEAR 2018]] [LOCATION [CITY Parker]] [CONDITION heavy rain showers] [CLOUD_COVERAGE partly cloudy]]]	Parker is not expecting any snow, but today there's a very likely chance of heavy rain showers, and it'll be partly cloudy

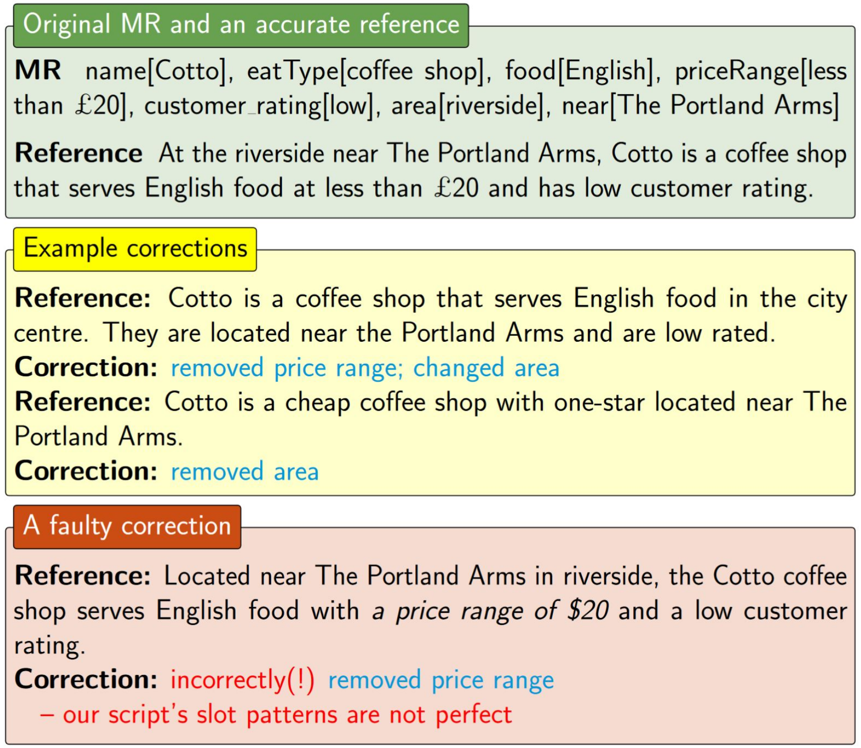
Annotated Response

[CONTRAST [INFORM_1 [LOCATION [CITY Parker]] is not expecting any [CONDITION_NOT snow]], but [INFORM_2 [DATE_TIME [COLLOQUIAL today]] there's a [PRECIP_CHANCE_SUMMARY very likely chance] of [CONDITION heavy rain showers] and it'll be [CLOUD_COVERAGE partly cloudy]]]

Data Noise & Cleaning

- NLG errors are often caused by **data errors**
  - ungrounded facts (← hallucinating)
  - missing facts (← forgetting)
  - domain mismatch
  - noise (e.g. source instead of target)
    - just 5% untranslated stuff kills an NMT system
- easy-to-get data are noisy
  - web scraping – lot of noise, typically not fit for purpose
  - crowdsourcing – workers forget/don't care
- **cleaning** improves situation a lot
  - can be done semi-automatically up to a point

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



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

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

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

Summary

- **NLG**: system action → system response
- **templates** work pretty well
- **seq2seq generation** with finetuned PLMs
  - best among data-driven
  - problems – hallucination, not enough diversity, needs lots of data
- **prompting-based** approaches with LLMs
  - less effort than finetuning
  - problems – hallucination, controllability, prompt sensitivity, model access
- mitigating problems: re-ranking, modularization, data cleaning

Thanks

Contact us:

https://ufaldsg.slack.com/
{kasner,odusek,hudecek}@ufal.mff.cuni.cz
Skype/Meet/Zoom/Troja (by agreement)

Labs in 10 minutes
Assignment 4

Next week: End-to-end models

Get these slides here:

http://ufal.cz/npfl099

References/Inspiration/Further:

- Gatt & Krahmer (2017): Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation http://arxiv.org/abs/1703.09902
- Sharma et al. (2022). Innovations in Neural Data-to-text Generation. https://arxiv.org/pdf/2207.12571.pdf
- Ondřej's PhD thesis (2017), especially Chapter 2: http://ufal.mff.cuni.cz/~odusek/2017/docs/thesis.print.pdf

Icons from https://www.flaticon.com/