School of CIT –
Social Computing
Research Group

Technical
University
of Munich

Advanced Natural Language Processing
CIT4230002

Prof. Dr. Georg Groh
Miriam Anschütz, M.Sc.

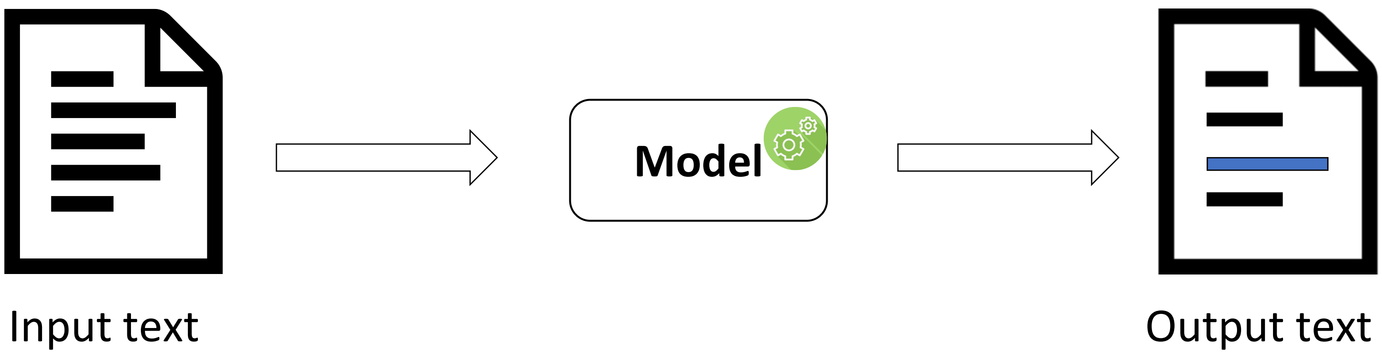
Lecture
Seq2Seq Introduction:
Tasks and architectures

Outline

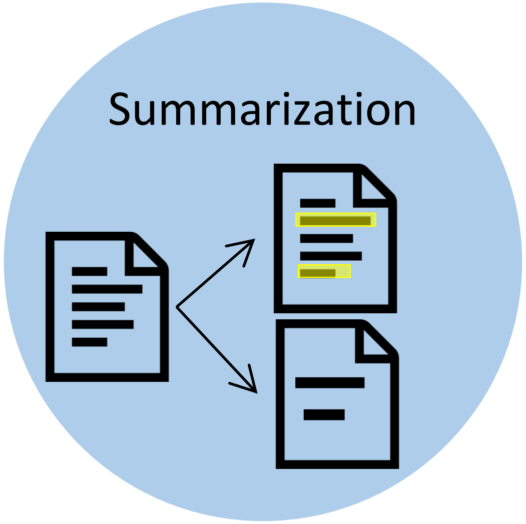
- **Seq2seq definition and tasks**
- Exploiting the Seq2Seq architecture
  - Summarization
  - Low-resource machine translation
- Factuality and hallucinations

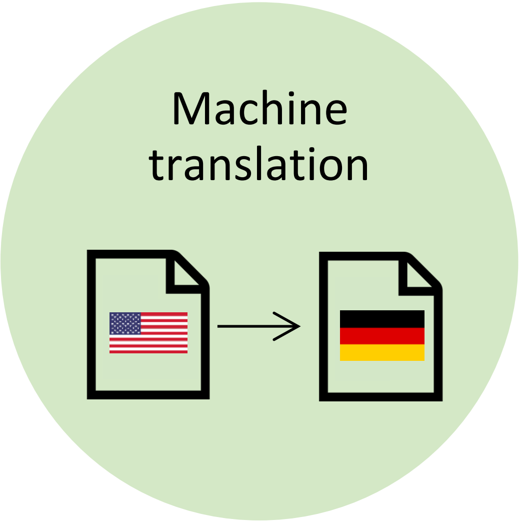
Terminology: Sequence to Sequence (Seq2Seq)

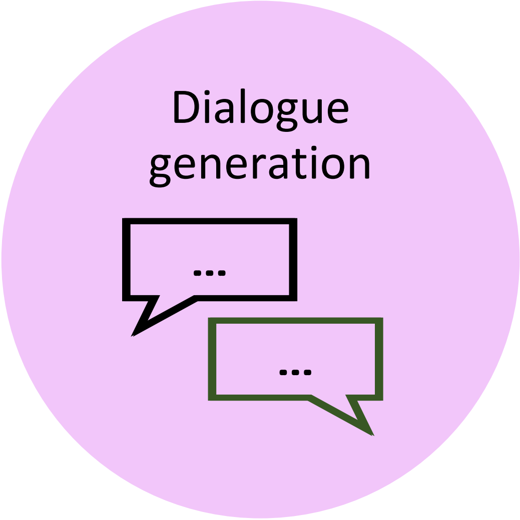
- Seq2Seq = task of generating output text based on input text
  - Output text with altered features
  - Can be solved by any model that can generate text



Seq2seq tasks







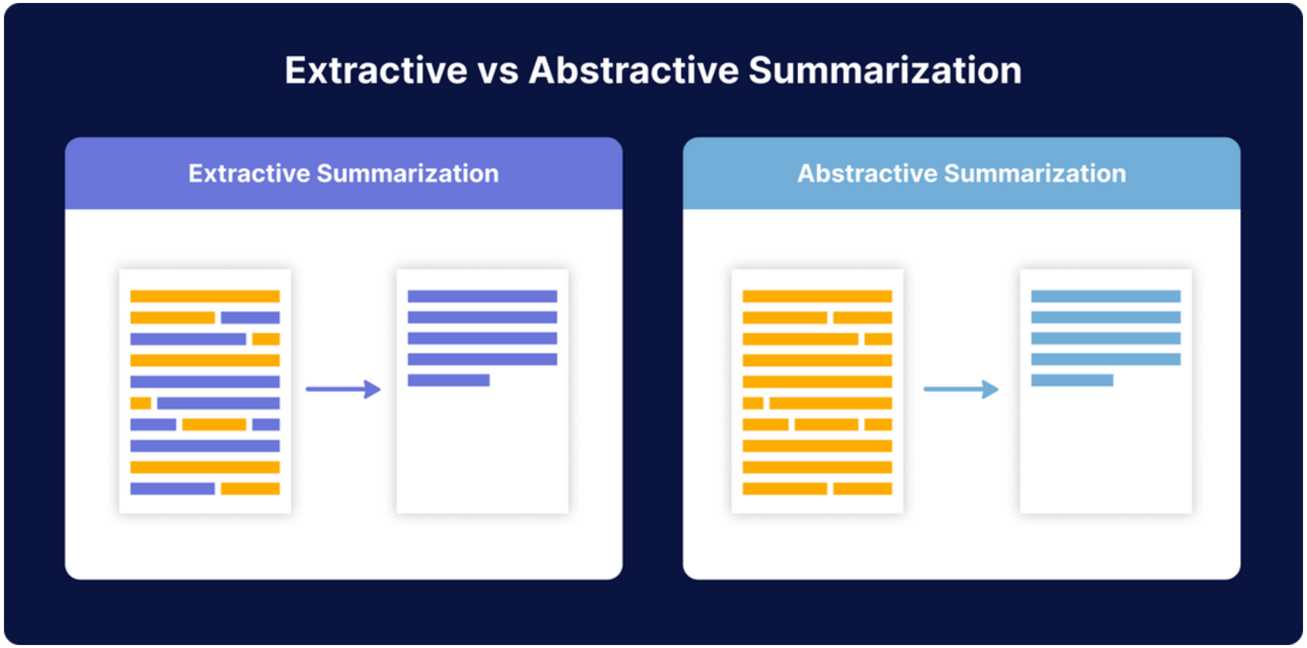
Summarization Approaches

Extractive Summarization

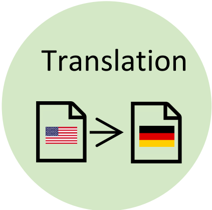
- Selects the most salient sentences from the input text and uses them to form the summary.

Abstractive Summarization

- Generates new sentences to provide a more coherent and fluent summary, often paraphrasing or rephrasing the original text.

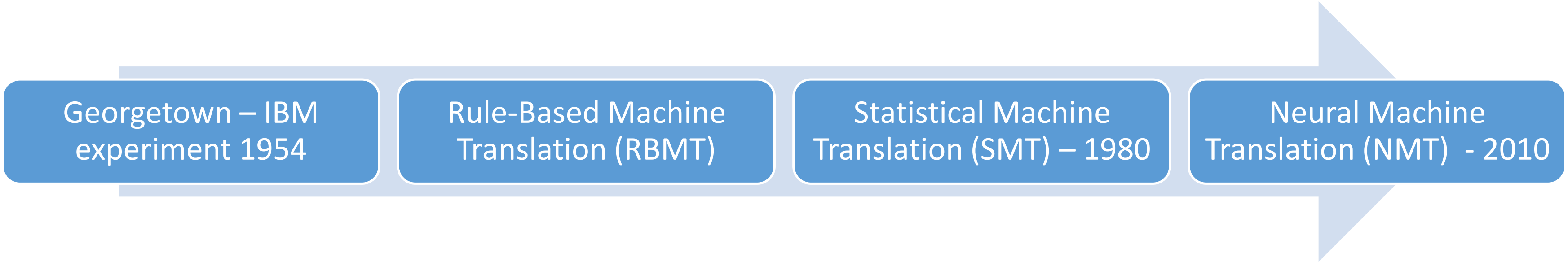


Machine Translation

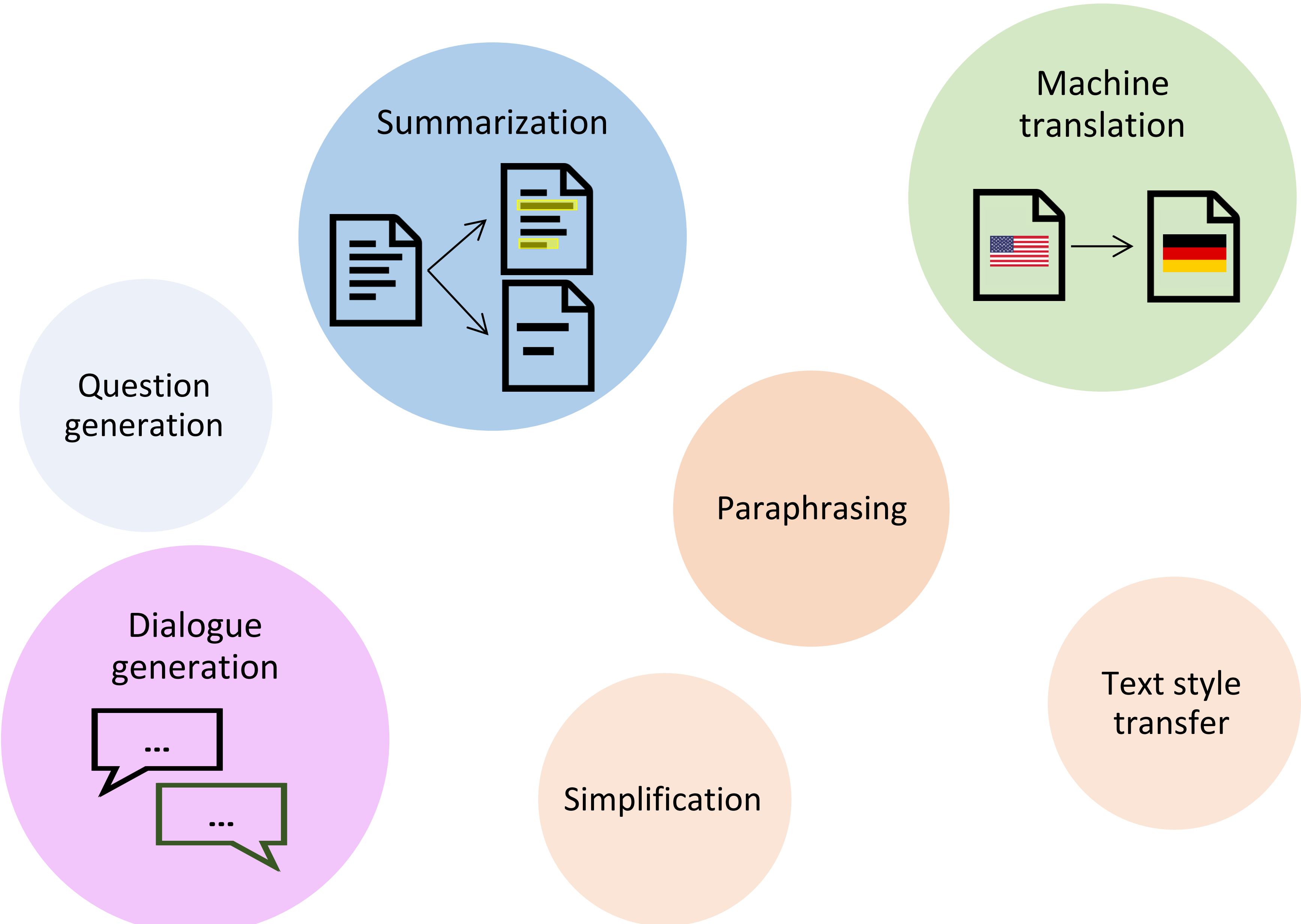


- Task: Translate sequence in source language into target language
  - No single perfect translation

    -> multiple outputs can be correct
  - Requires models to have multilingual understanding



Seq2seq tasks

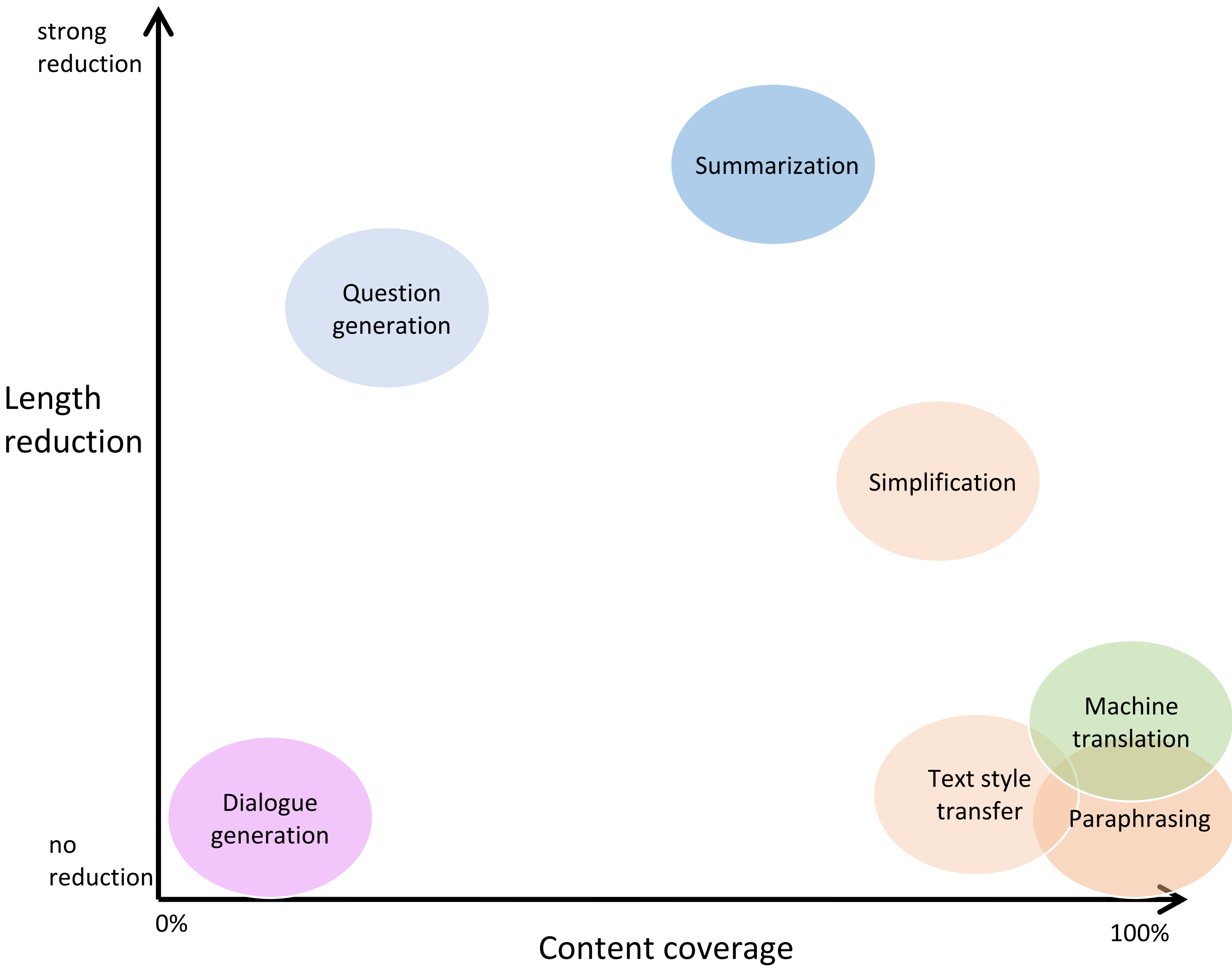


Discussion: Diversity of tasks

- **Similarity among all tasks**:
  - Input: Sequence of text
  - Output: Sequence of text with altered features

- **Differences and challenges**:
  - Input length (long document vs. short message)
  - (In)Formality (e.g. in dialogue)
  - Multilinguality
  - Closeness/Coverage of original text

Discussion: Closeness / Coverage of original text



Features of a good Seq2Seq output

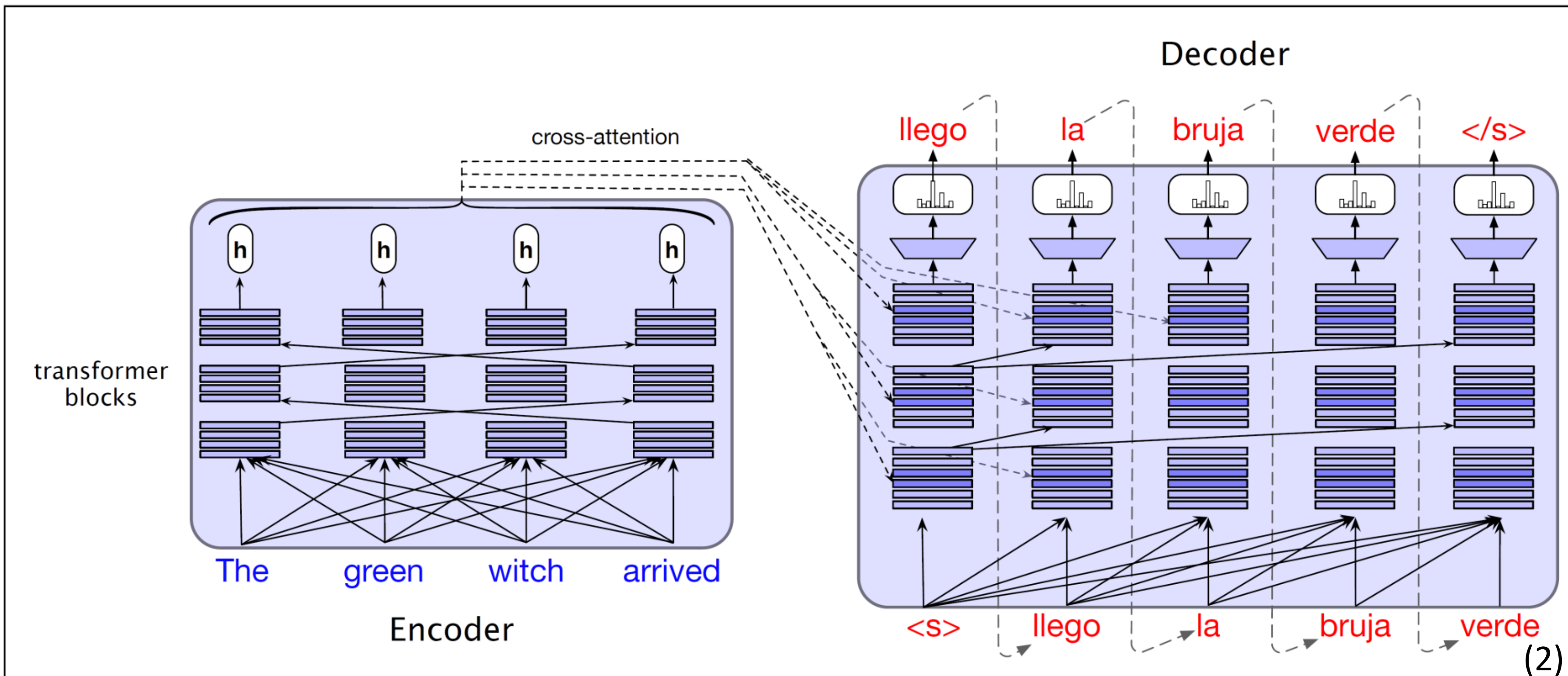
- **Fluency**: Quality of target language text
  - Grammatically correct
  - Coherent
  - Considers language characteristics (e.g. syntax, lexical divergences)

- **Adequacy**: Preservation of exact meaning
  - No information missing
  - No repetition
  - No unwanted content

=> Hard to judge everything at once!

Recap | Encoder-decoder architecture

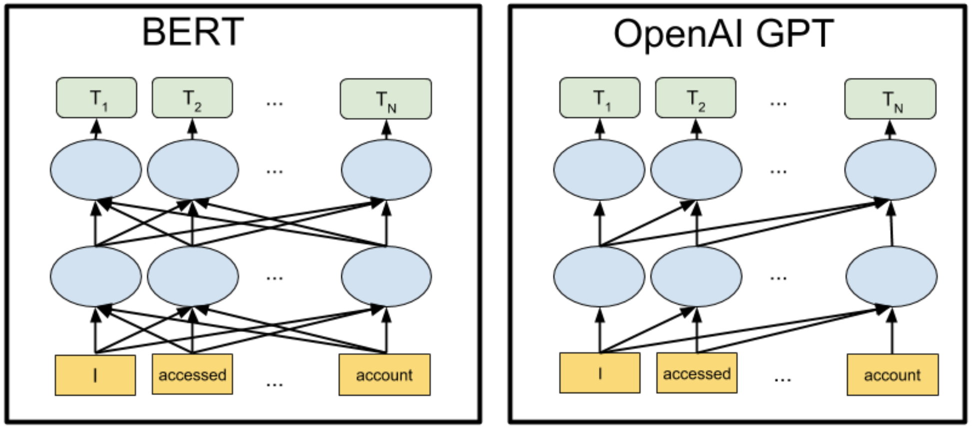
- **Encoder**: generates contextualized representation of input
- **Decoder**: autoregressively generates output
- **Cross-attention**: Decoder attends to source during generation



BART – Bidirectional Auto-Regressive Transformers

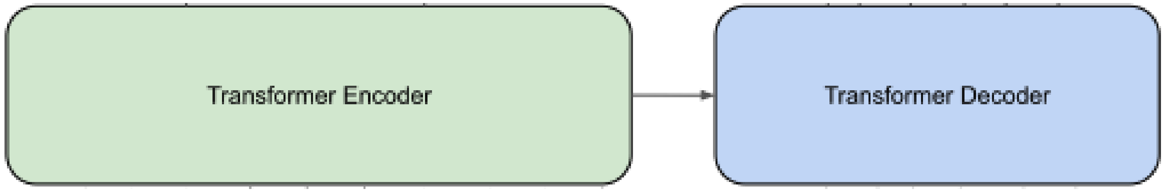
- BERT-like **bi-directional encoder**
  - Good at classification and encoding
- GPT-like **uni-direction decoder**
  - Good at text generation

- Trained by corrupting text with an arbitrary noise function, and learning a model to construct the original text.



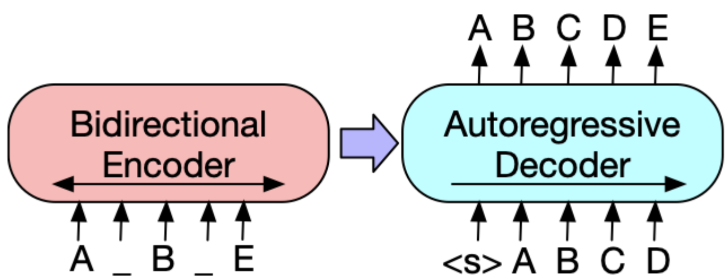
(2)

BART – Architecture



(a) BERT: Random tokens are replaced with masks, and the document is encoded bidirectionally. Missing tokens are predicted independently, so BERT cannot easily be used for generation.

(b) GPT: Tokens are predicted auto-regressively, meaning GPT can be used for generation. However words can only condition on leftward context, so it cannot learn bidirectional interactions.

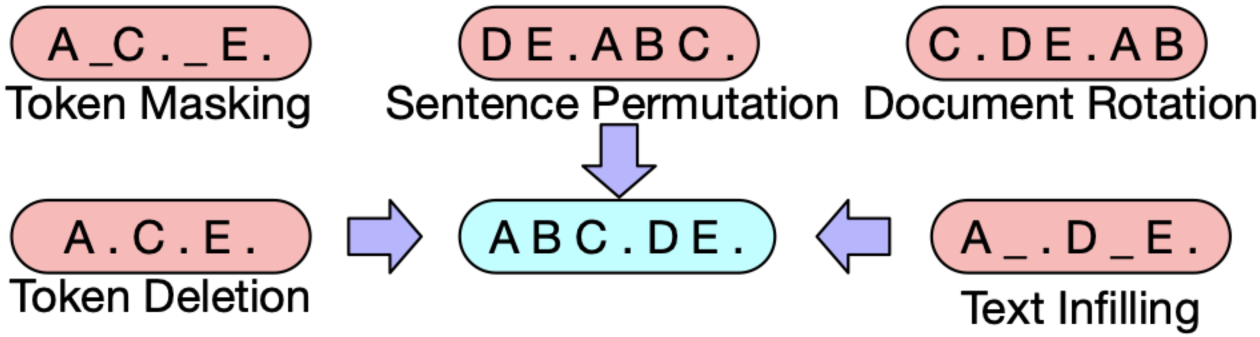


(c) BART: Inputs to the encoder need not be aligned with decoder outputs, allowing arbitary noise transformations. Here, a document has been corrupted by replacing spans of text with mask symbols. The corrupted document (left) is encoded with a bidirectional model, and then the likelihood of the original document (right) is calculated with an autoregressive decoder. For fine-tuning, an uncorrupted document is input to both the encoder and decoder, and we use representations from the final hidden state of the decoder.

(2)

BART – Transformers Pre-training

- **Token deletion**: random tokens are deleted from input, model must decide which positions are missing inputs.
- **Token infilling**: Inspired by Span-BERT – teach the model to predict how many tokens are missing from a span
- **Sentence permutation:** Document is divided into sentences based on full-stops, and sentences are shuffled in a random order
- **Document rotation:** Token chosen uniform, document is rotated so that it begins with that token. ("I have a pen" → "pen I have a")



(2)

Seq2Seq naming convention

- **Seq2Seq is overloaded**:
  - Group of tasks (in contrast to e.g. classification)
  - Model architecture (mostly encoder-decoder)

I asked ChatGPT for a seq2seq output

→ Seq2seq task with LLM

I used BART for summarization

→ Seq2seq task and model

I used a seq2seq approach

→ Seq2seq model

Dow we still need Seq2Seq-specific models??

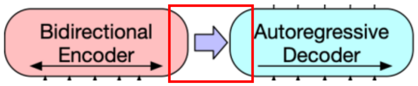
- LLMs can solve any seq2seq task!

- **But ..**
  - LLMs are inefficient (very large, expensive,..)
  - LLMs are harder to control and tend to hallucinate
  - Seq2Seq models are more task-specific
    → smaller model sufficient
  - Benefits of cross-attention in encoder-decoder models

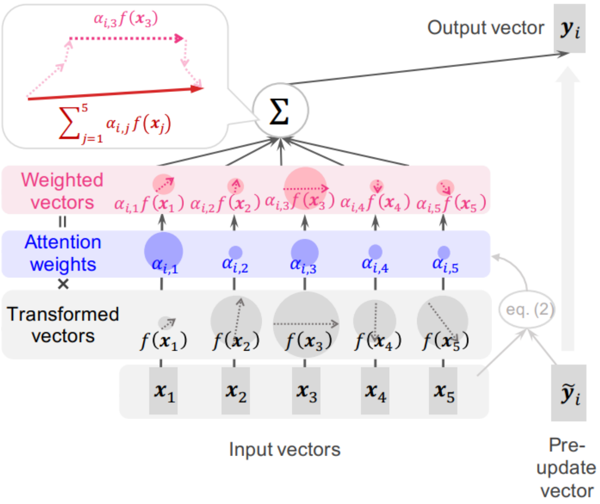
Outline

- Seq2seq definition and tasks
- **Exploiting the Seq2Seq architecture**
  - Summarization
  - Low-resource machine translation
- Factuality and hallucinations

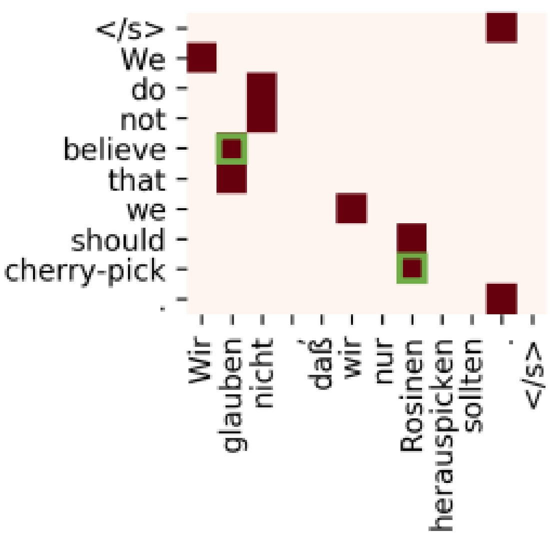
Cross-attention



- Cross-attention shows the influence of input tokens on output tokens:



→ **Model-based explanation** for free!



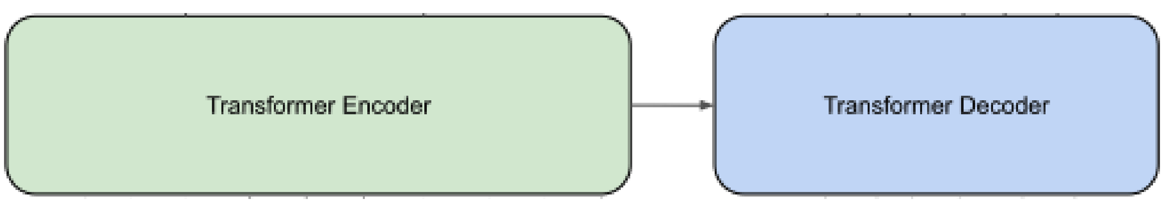
(5)

Outline

- Seq2seq definition and tasks
- Exploiting the Seq2Seq architecture
  - **Summarization**
  - Low-resource machine translation
- Factuality and hallucinations

PEGASUS

- Short for: **P**re-training with **E**xtracted **GA**p-sentences for Abstractive **SU**mmarization**S**
- Pre-training objective tailored for abstractive summarization tasks
- Hypothesis: fine-tuning performance is improved by choosing a pre-training self-supervised objective close to the final downstream task

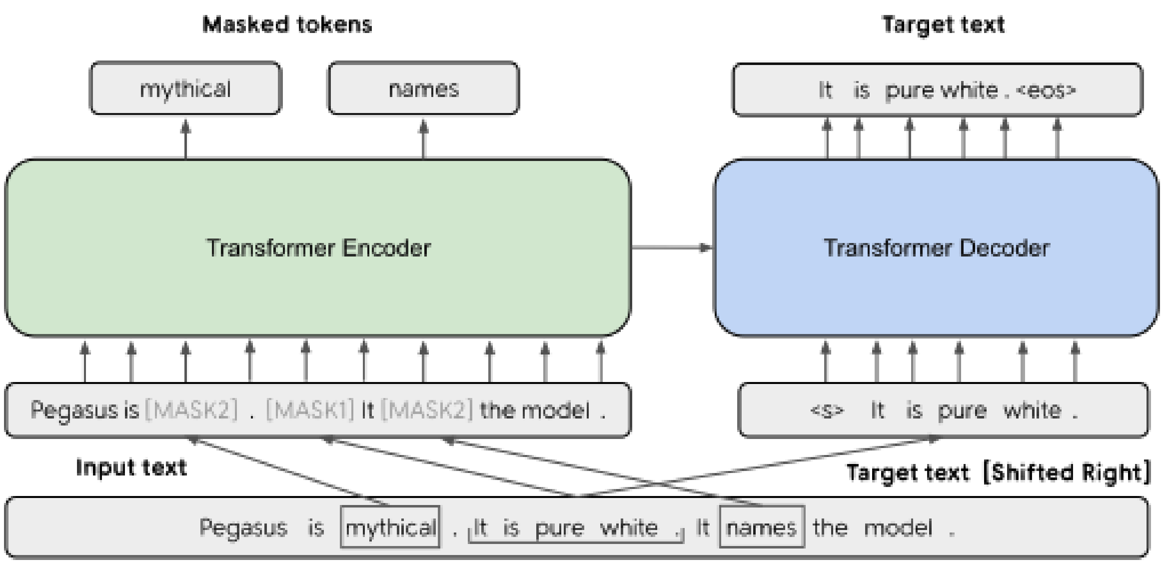


(3)

PEGASUS - Pre-training

- **Gap sentence generation (GSG)**
  - Training input: a document with missing sentences
  - Training output: missing sentences concatenated together

  - **Gap sentence generation (GSG)** and masked language model (MLM) were experimented during pre-training
    → only GSG improved performance



(3)

PEGASUS – Pre-training

- **Gap sentence generation (GSG) – Sentence selection**
  - **Random:** Uniformly select m -sentences at random
  - **Lead:** Select first m sentences
  - **Principal:** select top-m scored sentences according to importance; sentences are scored independently and top m selected

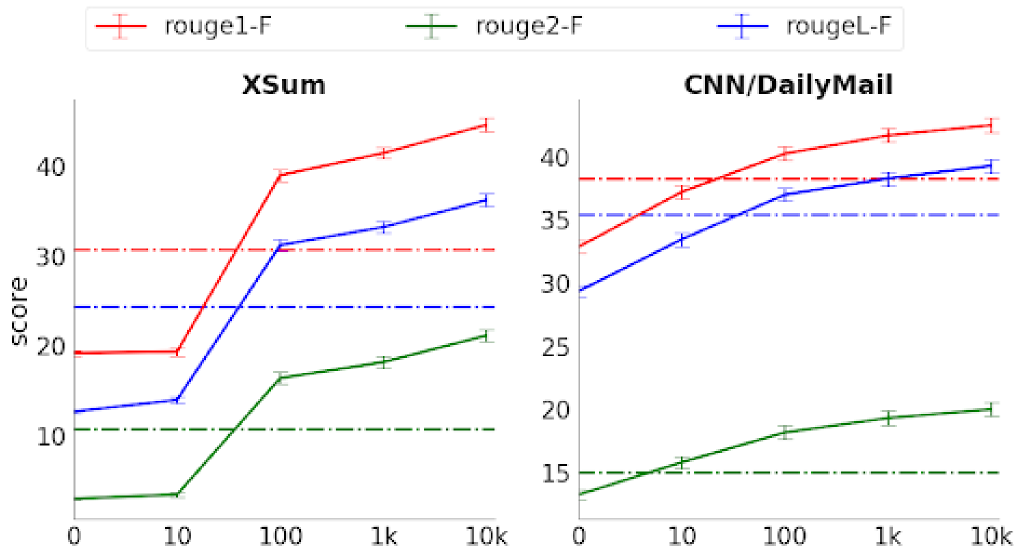
$$s_i = rouge(x_i, D\{x_i\}), \forall i$$

INVITATION ONLY We are very excited to be co-hosting a major drinks reception with our friends at Progress. This event will sell out, so make sure to register at the link above. Speakers include Rajesh Agrawal, the London Deputy Mayor for Business, Alison McGovern, the Chair of Progress, and Seema Malhotra MP. Huge thanks to the our friends at the ACCA, who have supported this event. The Labour Business Fringe at this year's Labour Annual Conference is being co-sponsored by Labour in the City and the Industry Forum. Speakers include John McDonnell, Shadow Chancellor, and Rebecca Long-Bailey, the Shadow Chief Secretary to the Treasury, and our own Chair, Kitty Ussher. Attendance is free, and refreshments will be provided.

(3)

PEGASUS - Downstream

- Model was fine-tuned on mupltiple datasets (Xsum, CNN/DailyMail,..)
- It showed good performance with less examples for fine-tuning



Fine-tuning with limited supervised examples. The solid lines are $\text{PEGASUS}_{\text{LARGE}}$ fine-tuned on 0 (zero shot), 10, 100, 1k,10k examples. The dashed lines are $\text{Transformer}_{\text{BASE}}$ models, equivalent in capacity as $\text{PEGASUS}_{\text{BASE}}$ and trained using the full supervised datasets, but with no pre-training

(3)

Outline

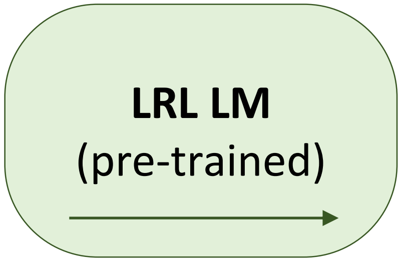
- Seq2seq definition and tasks
- Exploiting the Seq2Seq architecture
  - Summarization
  - **Low-resource machine translation**
- Factuality and hallucinations

Language model as prior

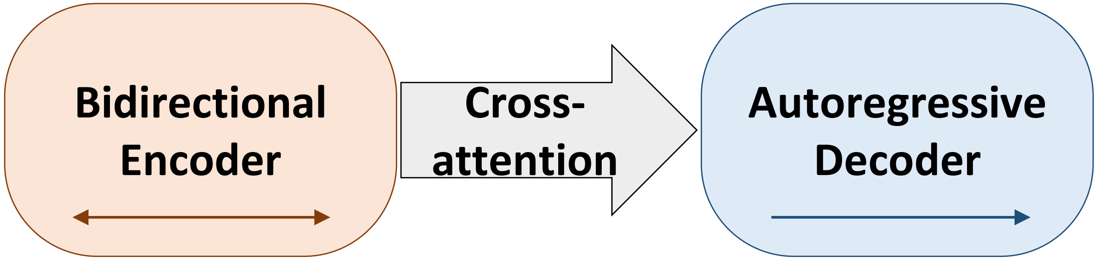
- Situation: only a few parallel data available
  - → end-to-end training won't work

- Can we use monolingual data from low-resource language (LRL)?

- Given:

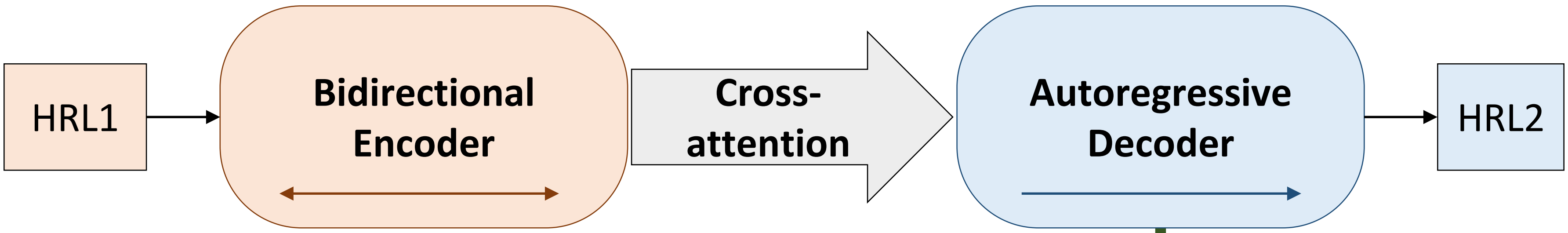


- How can we train?

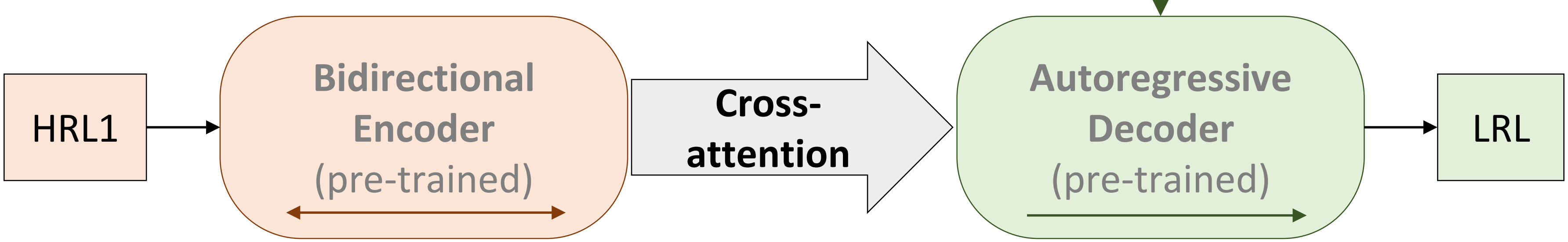


Language model as pre-trained decoder: Deep fusion

- BART model pre-trained for translation of high-resource languages (HRL)

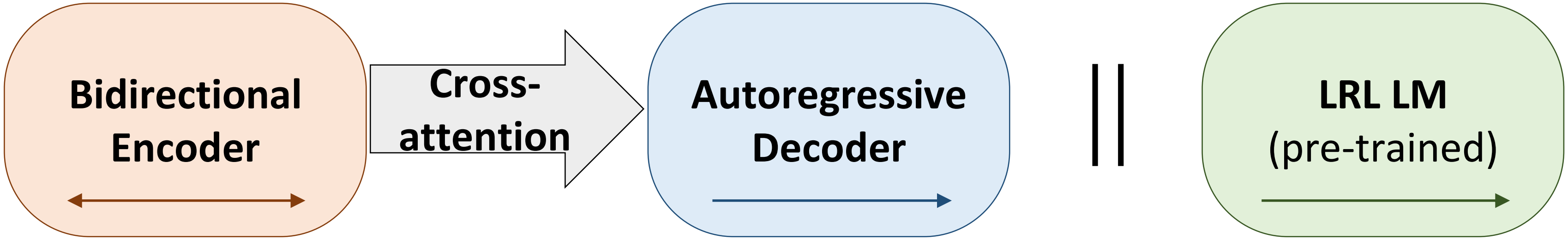


- Idea: Replace decoder with language model in low-resource language (LRL)
- Train only cross-attention on aligned data



Language model as prior

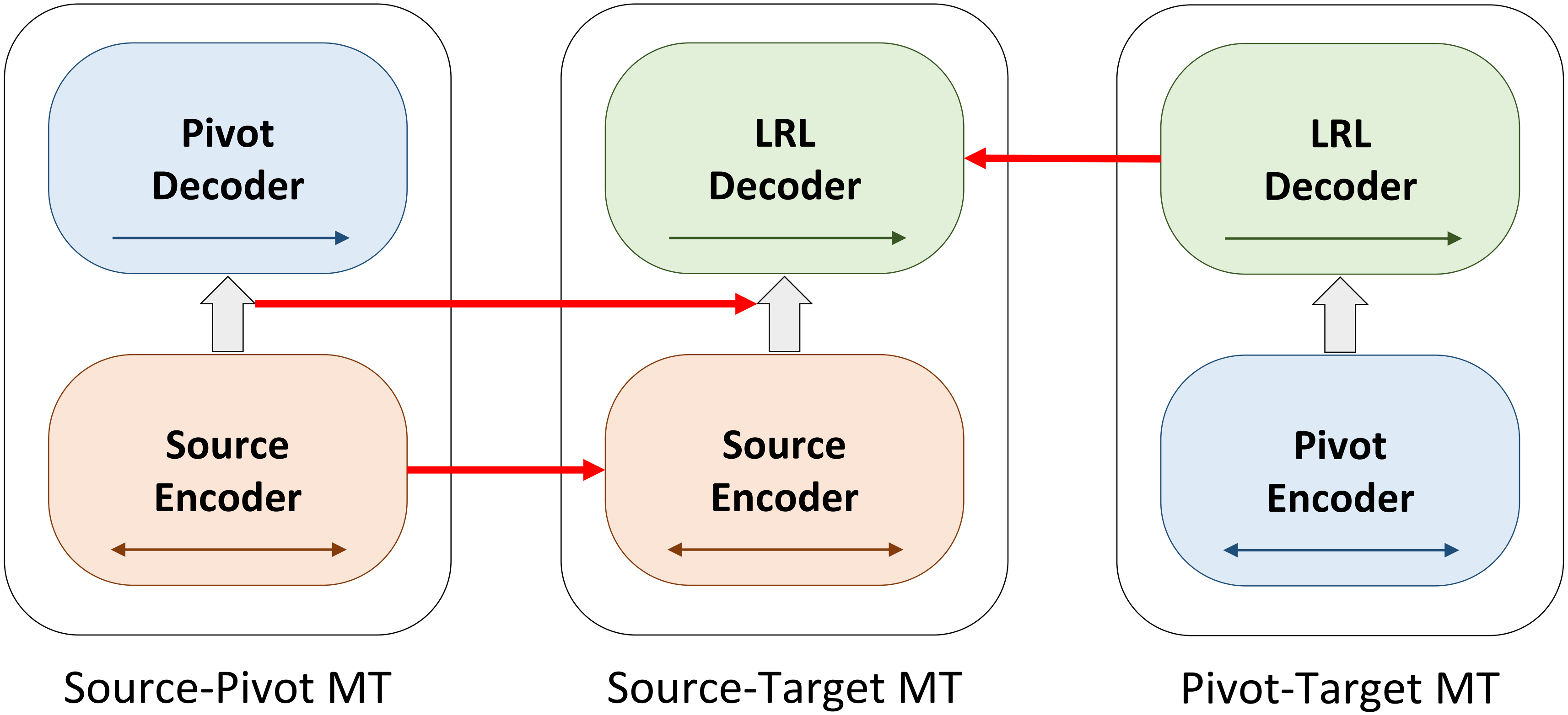
- Idea: LM’s distribution of words describes a language
    - MT decoder’s distribution should be similar to LM’s
    - → LM as prior distribution
    - → minimize Kullback-Leibler divergence between distributions in training



- Decoder is allowed to deviate from LM, e.g. for rare/unseen words

Transfer learning: Pivot languages

- Idea: Intermediate language (pivot) for translation pair
  - Existing parallel data for source-pivot and pivot-LRL
  - Parallel data mostly for non-English parent language
  - Otherwise: Multiple pivot languages combined



Outline

- Seq2seq definition and tasks
- Exploiting the Seq2Seq architecture
    - Summarization
    - Low-resource machine translation
- **Factuality and hallucinations**

Faithfulness and Hallucinations

- **Quality criterion adequacy**: Preservation of exact meaning
- **Faithfulness**: Staying consistent and truthful to the source

  → All facts in translation are grounded in source
- **Hallucination**: Output (parts) that are nonsensical or unfaithful to source
  - **Intrinsic**: misrepresented information contradicting the source text

    Source: “*John became an older brother because Mary gave birth to a girl.*”

    Summarization candidate: “*Mary gave birth to a boy.*”

    Source: “*John became an older brother because Mary gave birth to a girl.*”

    Summarization candidate: “*John gave birth to a girl.*”
  - **Extrinsic**: Generating information not contained in source

    Source (German): “*Dieses Haus ist in einer großen Stadt.*”

    Translation candidate: “*This house is in the big city close to the ocean.*”
- => How to measure consistency?

Faithfulness and Factuality

- **Faithfulness**: Staying consistent and truthful to the source

  → All facts in translation are grounded in source

- **Factuality**: Staying consistent and truthful to world knowledge

Source (German): "*Dieses Haus ist in einer großen Stadt.*"

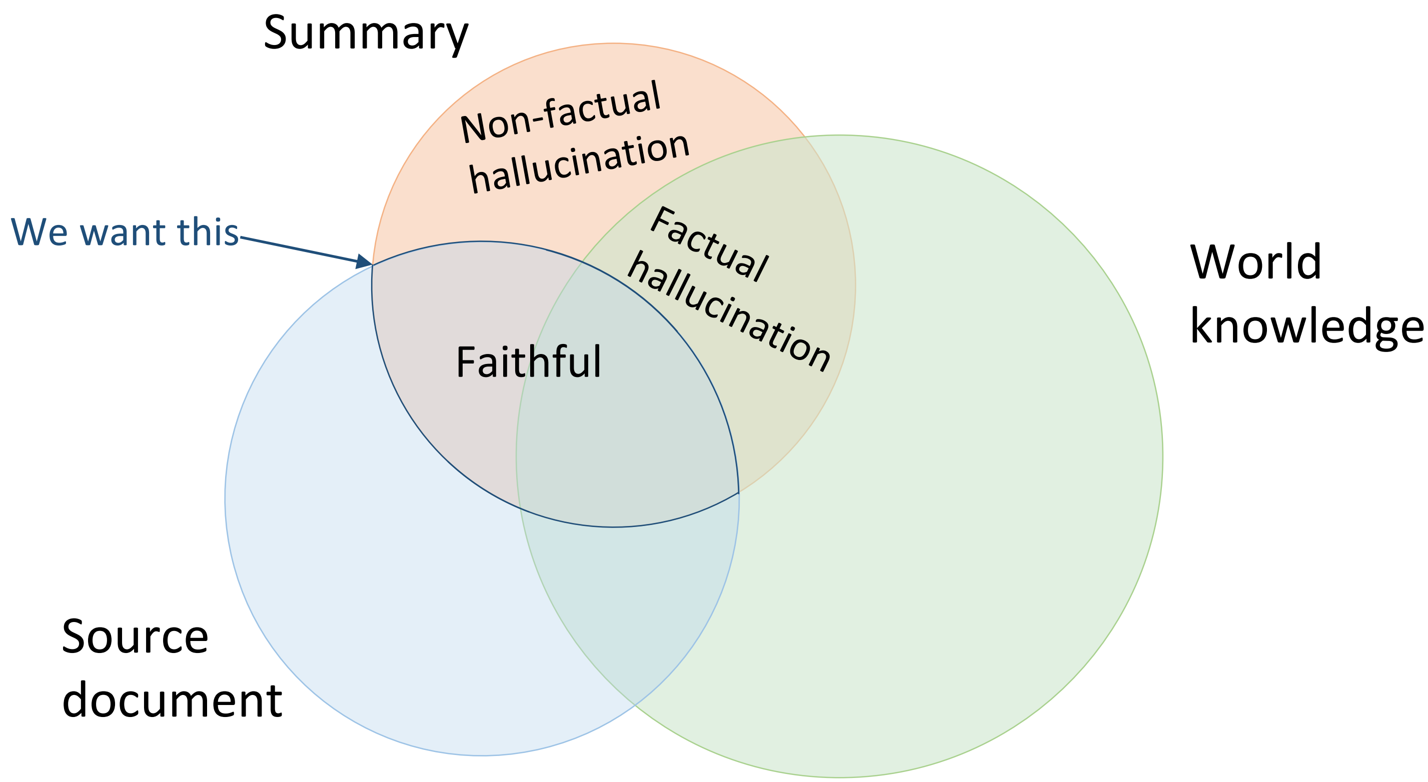
Translation candidate 1: *"This house is in a big city where many people live in."* → **not faithful, but factual**

Source (German): "*Dieses Haus ist in einer großen Stadt mit fliegenden Autos.*"

Translation candidate 1: *"This house is in a big city with flying cars."* → **faithful, but not factual**

- Evaluating factuality needs knowledge base/world knowledge representation
  → See lecture about knowledge enhancement

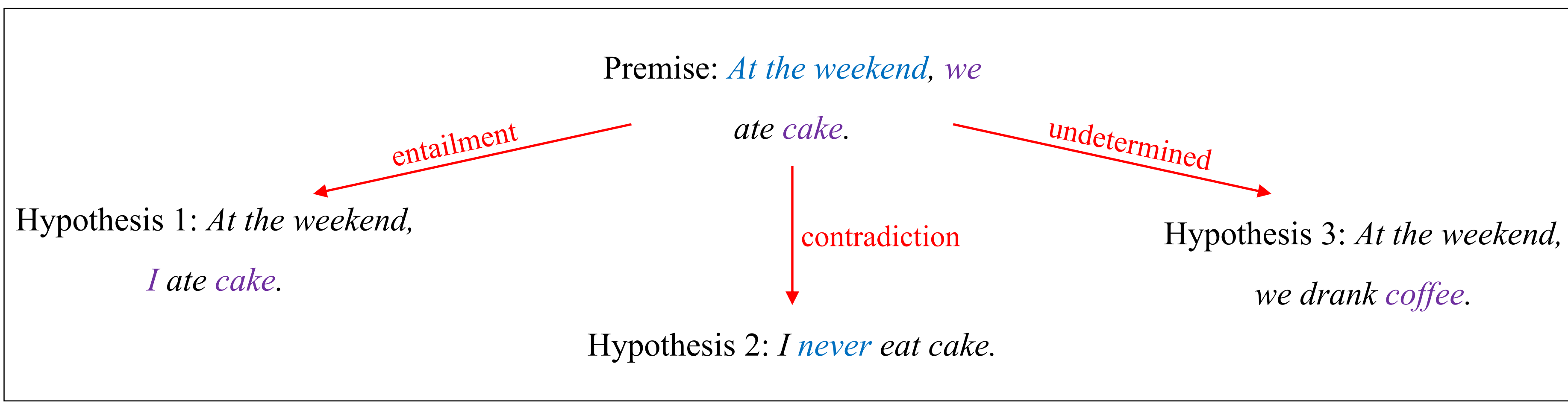
Faithfulness and Factuality



- Evaluating factuality needs knowledge base/world knowledge representation
  → See lecture about knowledge enhancement

Textual Entailment (TE)

- Idea borrowed from Natural Language Inference:
  Given a premise - Classify a hypothesis as
  - True → *Entailed in premise*
  - False → *Contradicts premise*
  - Neutral → *Undetermined*



Textual Entailment (TE) for seq2seq evaluation

- Apply pre-trained entailment classifier on output:
  - Premise = source, Hypotheses = candidates
  - All candidates must be entailed in source

- For machine translation, classifier needs to understand both languages
  → Use multilingual model

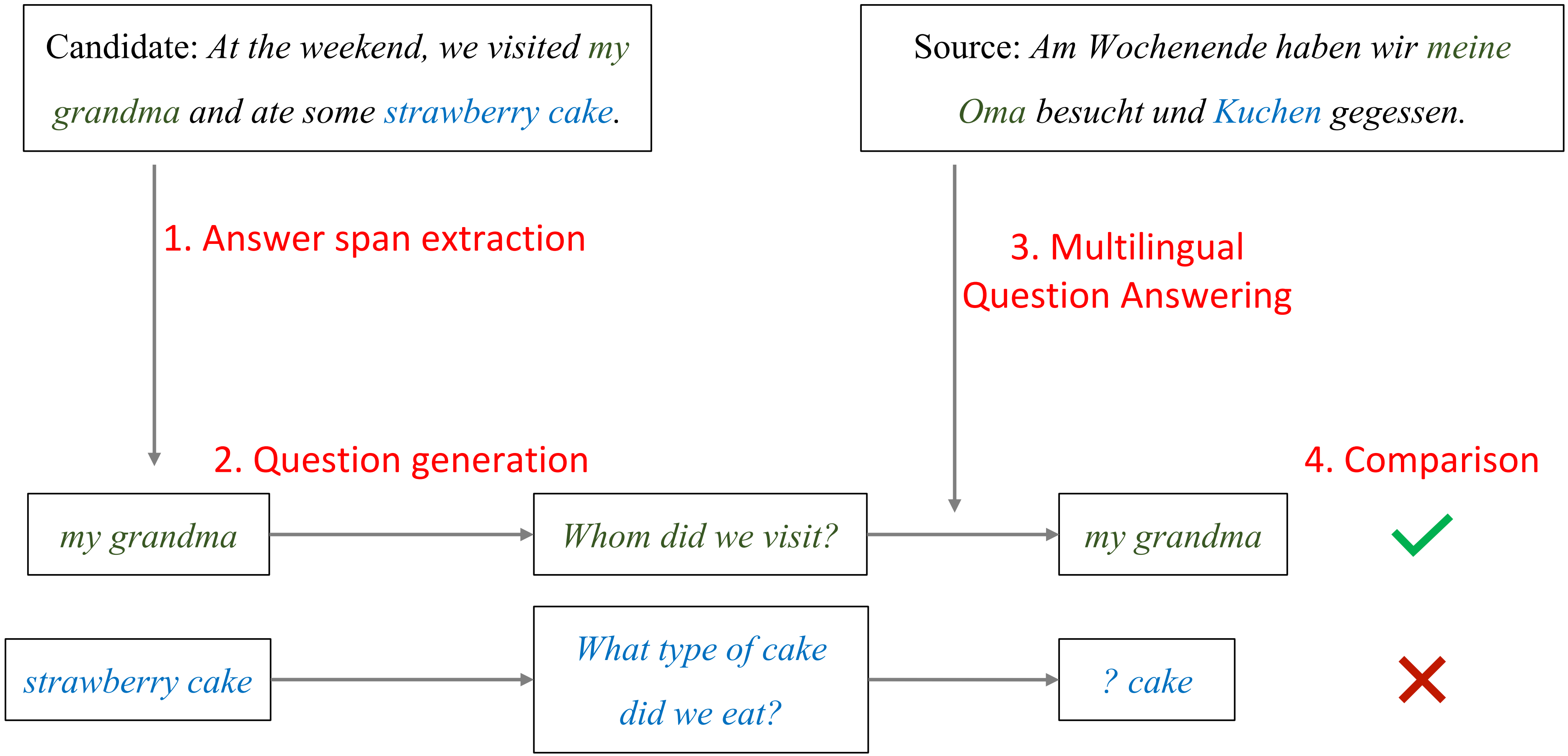
Source (German): "*Dieses Haus ist in einer großen Stadt.*"
Translation candidate 1: "*This house is in a big city where many people live in.*" → **TE: undefined**

Question Answering for hallucination detection

- Idea: Ask the same questions to candidates and source/references
  - Answers should be similar
  - Hallucination = Answers only by candidate

- Evaluation steps:
  1. Extract possible answer spans from candidate
  2. Generate questions for these answers
  3. Answer questions with source
  4. Compare both answers

Question Answering for evaluation | Example

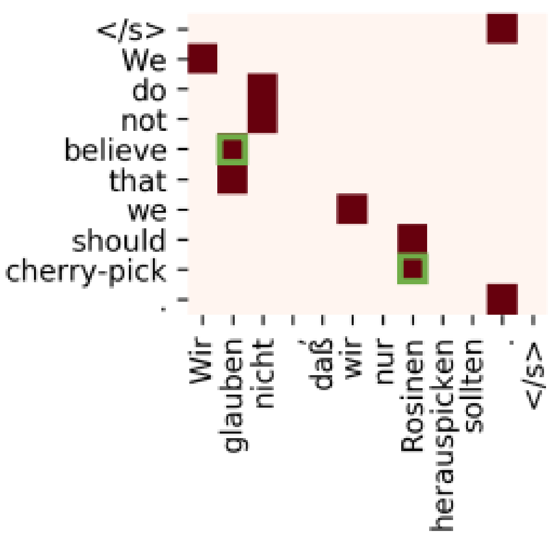


adapted from (7)

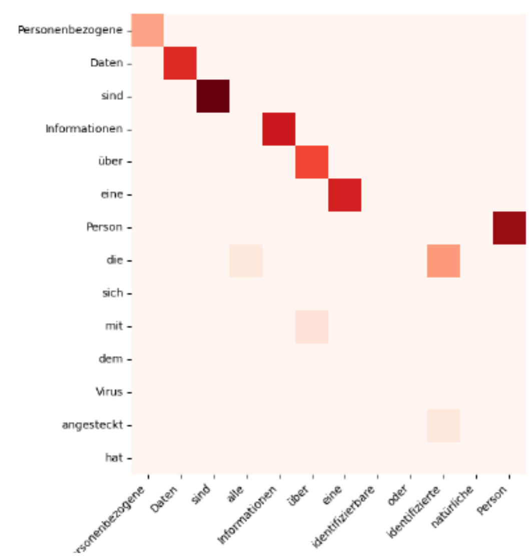
Model-based Hallucination detection

- Idea: Hallucinations are not based on source text
  → they should show a different cross-attention pattern!

- **=> Does the model “know” when it hallucinates?**



Correct sample



Sample with hallucination

(5)

Model-based Hallucination detection II

- Guerreiro et al: estimate cost of transferring source distribution to translation distribution
  - Measured with Wasserstein distance:

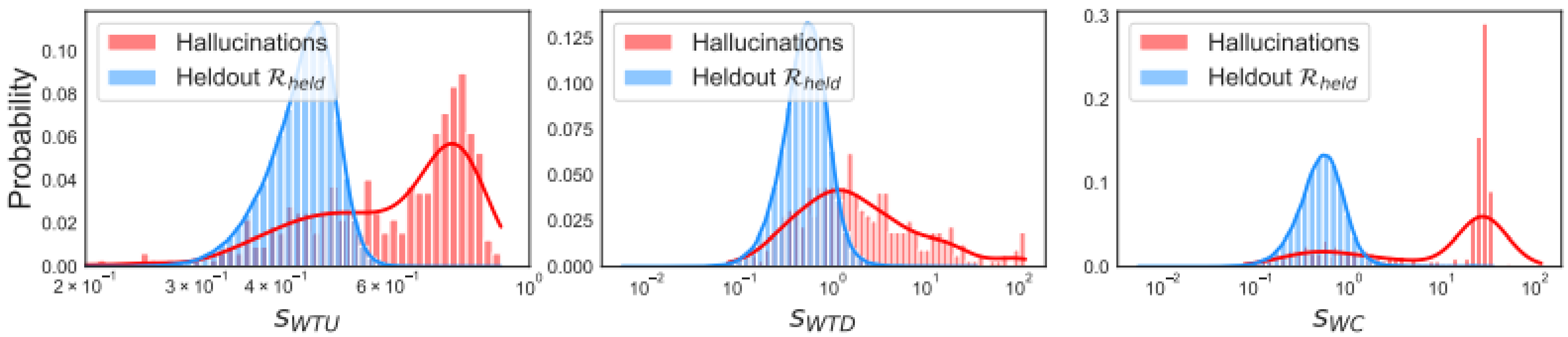


Figure 2: Histogram scores for our methods – Wass-to-Unif (left), Wass-to-Data (center) and Wass-Combo (right). We display Wass-to-Data and Wass-Combo scores on log-scale.

→ Competitive with external detection metrics

(8)

Discussion: Are there desirable hallucinations?

- Text simplification: give explanation for words

Source (German): “*Dieses Haus ist in einer großen Stadt.*”
Translation candidate 1: “*This house is in a big city, a place where many people live in.*”

→ Include factuality check

- Distinguish between more different types of hallucinations?

Final thoughts

- **Seq2Seq** = “Text in – text out”
  - Collection of tasks (with different requirements)
  - Encoder-decoder architecture to solve those tasks

- LLMs are great at solving these tasks – BUT seq2seq architecture can have benefits!

- **Hallucinations** are text snippets created without relying on the source text
  - Can be detected with external metrics..
  - .. Or based on the models themselves

Bibliography

(1) Dan Jurafsky et al. 2022. Speech and Language Processing book

(2) Lewis et al. 2020. BART

(3) Zhang et al. 2020. PEGASUS

(4) Mhaskar et al. 2022. Pivot languages in NMT

(5) Kobayashi et al. 2020. Attention is not only a weight

(6) Ji et al. 2022. Hallucination in NLG

(7) Krubiński et al. 2021. Q&A in evaluation

(8) Guerrerio et al. 2023. Model-based hallucination detection

Study Approach

Minimal

- work with the slides

Standard

- minimal approach + skim references 2 and 5

In-Depth

- standard approach + read references 2 and 5

See you next time!