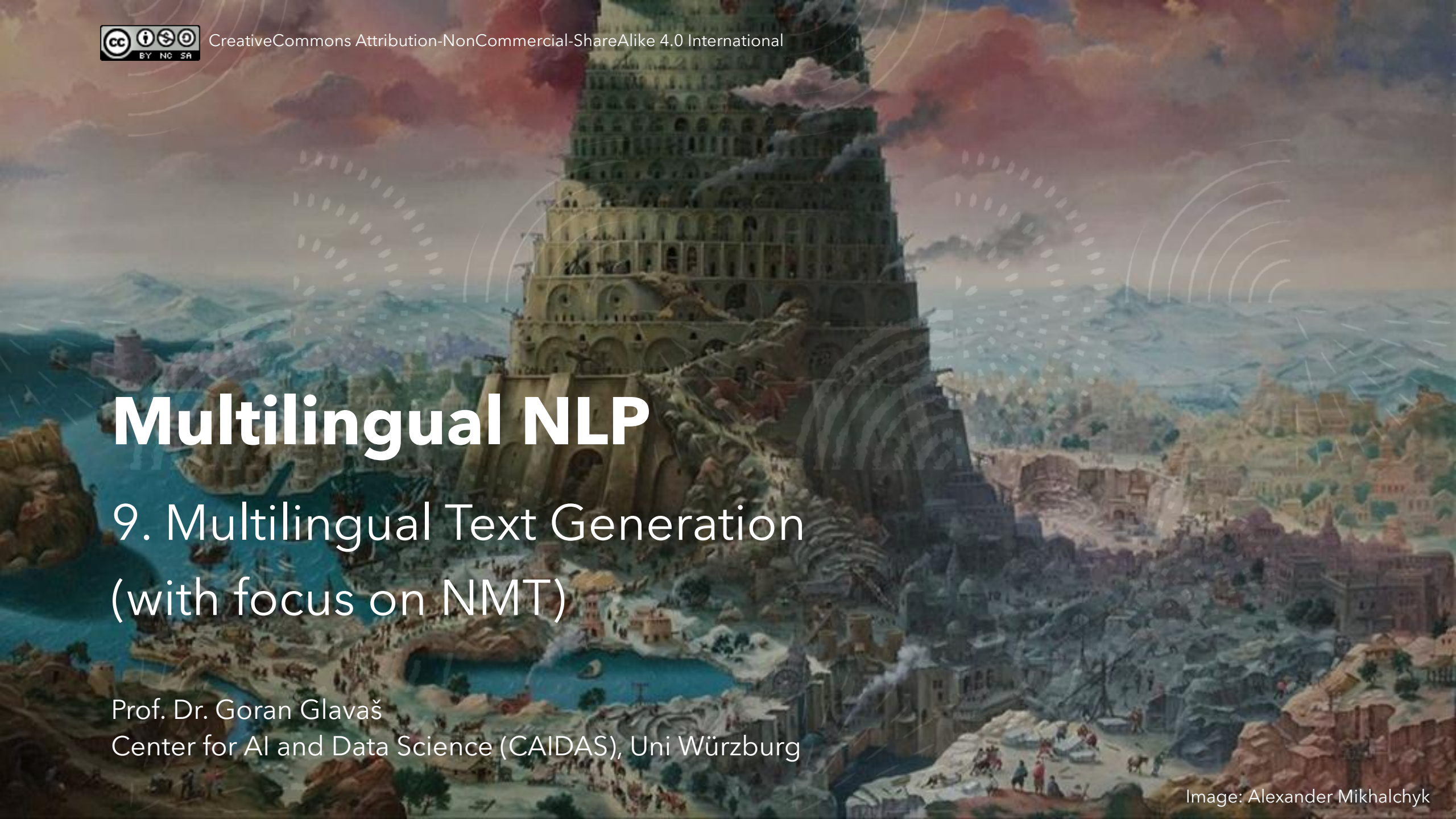


Multilingual NLP

9. Multilingual Text Generation (with focus on NMT)

Prof. Dr. Goran Glavaš
Center for AI and Data Science (CAIDAS), Uni Würzburg

Image: Alexander Mikhalchyk

After this lecture, you'll...

- Learn about text generation (with Transformers)
- Understand core principles of neural machine translation (NMT)
- Know differences between encoder-decoder and decoder-only models
- Learn about decoding strategies for language generation
- Be familiar with methods for (multilingual) text generation evaluation

Content

- **Text Generation**
- Encoder-Decoder vs. Decoder-Only Models
- Decoding Strategies
- Multilingual Machine Translation
- Evaluating Text Generation

Text Generation

- So far, we have mostly dealt with **language „understanding"**
  - Given an input text, predict something for it:
    - Assign a label (class or score) to the sequence or tokens

- **Text generation**: tasks that require generation of text that in some aspect conforms to the provided (text) input
  - Retrieval of existing text for the given input is not text generation
  - What „conforming" means is task-dependent (e.g., different for MT and text summarization)
  - But we always assume that text to be generated has to be grammatical (i.e., „not broken")

Text Generation

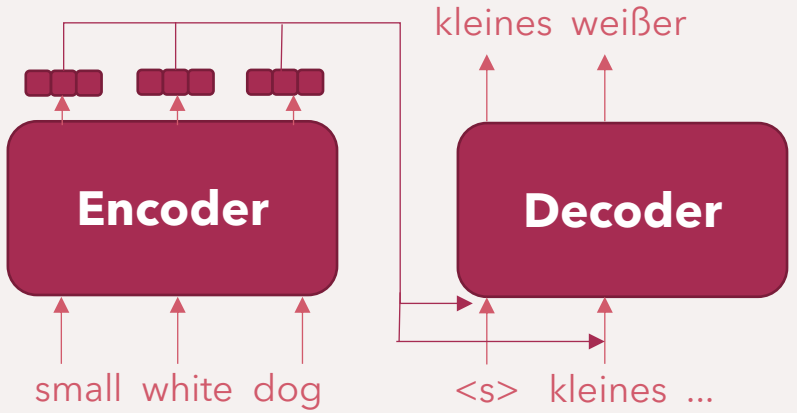
- Traditionally, the most prominent generation tasks:
- Machine translation
  - Generated text in the target language must semantically match the input text in the source language
- Summarization
  - Generated summary must contain the most important information from the input and not be redundant (length limit)
- Dialog („Conversational AI")
  - Generated text must represent a meaningful reply to the last user utterance as well as the entire conversation history

Text Generation

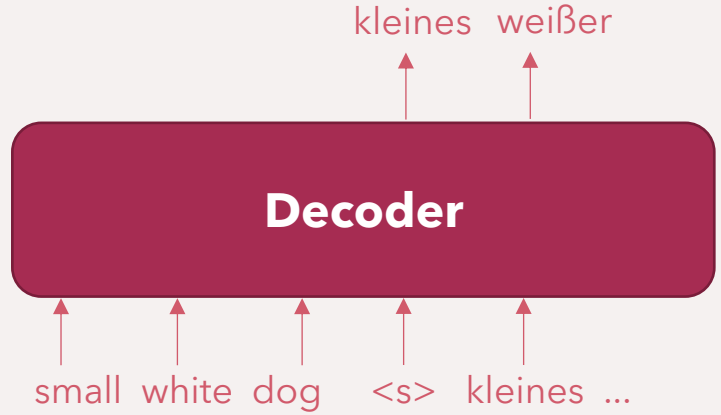
- Some other generation tasks:
  - Simplification
    - Generated text must convey the same information as the input text but with simpler (lexically, syntactically) language
  - Data-to-Text
    - Input not text, but some structured data (with certain semantics), generated text must reflect correctly data semantics
- Text generation and multilinguality
  - MT is inherently multilingual (at least bilingual)
  - Cross-lingual summarization: summary in different language from input
  - Other tasks can be cast monolingually (one model for each language) or multilingually (one model for two or more languages)

Text Generation Models

- **Encoder-Decoder** models
  - Encoder is a separate neural network from decoder: it summarizes/aggregates the input text
  - Decoder generates the output text by „consuming" the output of the Encoder
    - It also takes into account all previously generated tokens



- **Decoder-only** models
  - Only one neural network (decoder)
  - Input text is simply given as preceding context to the model
  - Suffices when it's a large pretrained model (LLM)



Content

- Text Generation
- **Encoder-Decoder vs. Decoder-Only Models**
- Decoding Strategies
- Multilingual Machine Translation
- Evaluating Text Generation

Encoder-Decoder Models

Kalchbrenner, N., & Blunsom, P. (2013, October). Recurrent continuous translation models. In *Proceedings of the 2013 conference on empirical methods in natural language processing* (pp. 1700-1709).

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, *27*.

- Initial Encoder-Decoder Models (for MT) were based on recurrent components: encoder & decoder an LSTM or GRU
- Seminal work (see above) introduced **neural MT (NMT)** as a viable alternative to traditional **statistical MT (SMT)**

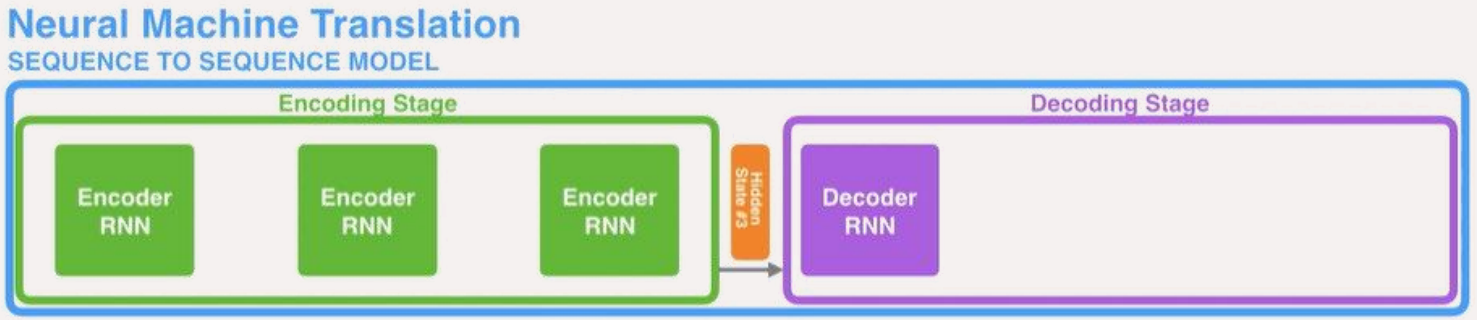


Image from: https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/

Encoder-Decoder Models

Bahdanau, D., Cho, K. H., & Bengio, Y. (2015, January). Neural machine translation by jointly learning to align and translate. In 3rd International Conference on Learning Representations, ICLR 2015.

- Initial NMT models had problems with translations of long sequences
  - Encoder compressed the entire input (no matter how long) into a single fixed-size vector
- Bahdanau et al. introduced **cross-attention**
  - Encoding of each token becomes available to the decoder

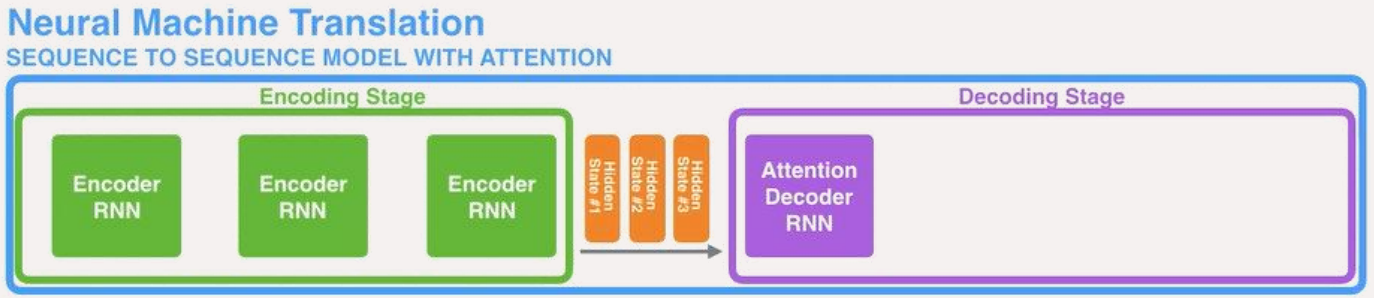
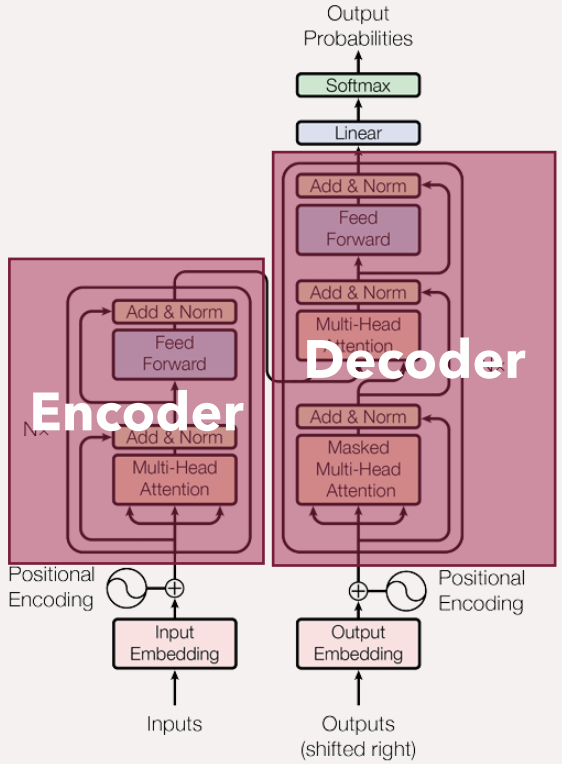


Image from: https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/

Transformer (Encoder-Decoder)

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems (NeurIPS).

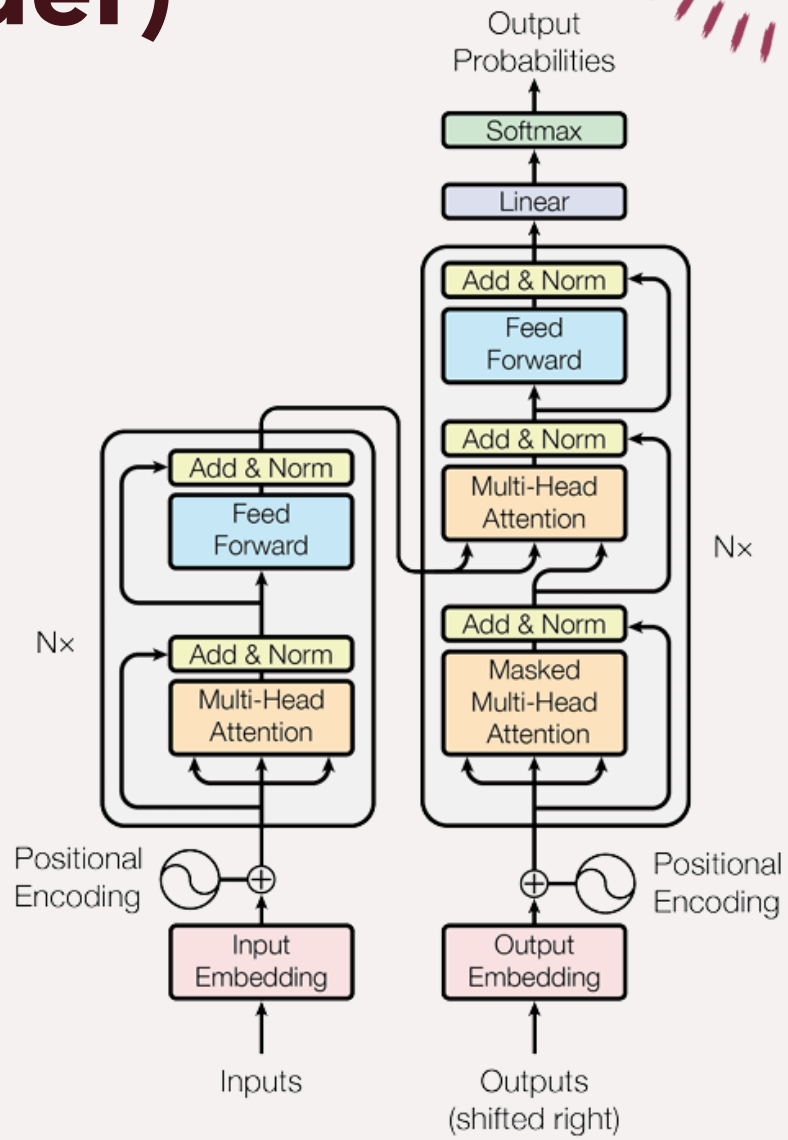
- **Transformer** as proposed by Vaswani et al. is an encoder-decoder model
  - Introduced for NMT
- Basically removes the recurrent components from the Bahdanau's model
- The **decoder** relies on both self-attention and cross-attention mechanisms



Transformer (Encoder-Decoder)

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems (NeurIPS).

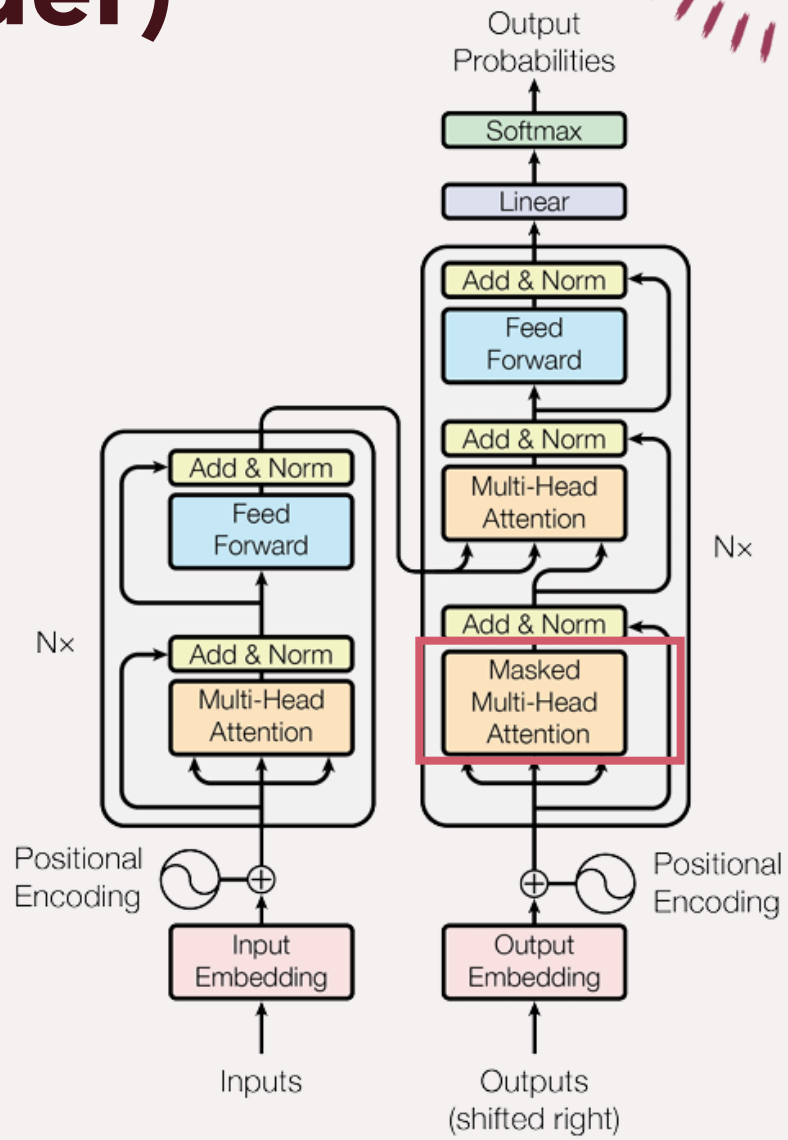
- Layers of the **decoder** have three sublayers
  - Multi-head self-attention*
  - Multi-head cross-attention
  - Feed-forward layer
    - Exactly the same as in encoder-only Transformer (covered in Lecture 4)



Transformer (Encoder-Decoder)

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems (NeurIPS).

- *Multi-head self-attention of decoder operates differently than in the encoder
- We have to prevent tokens from attending over their future tokens.
  - Q: Why?
- Future token masking
  1. Compute self-attention normally
  2. Adjust attention scores for future tokens before softmax



Transformer (Encoder-Decoder)

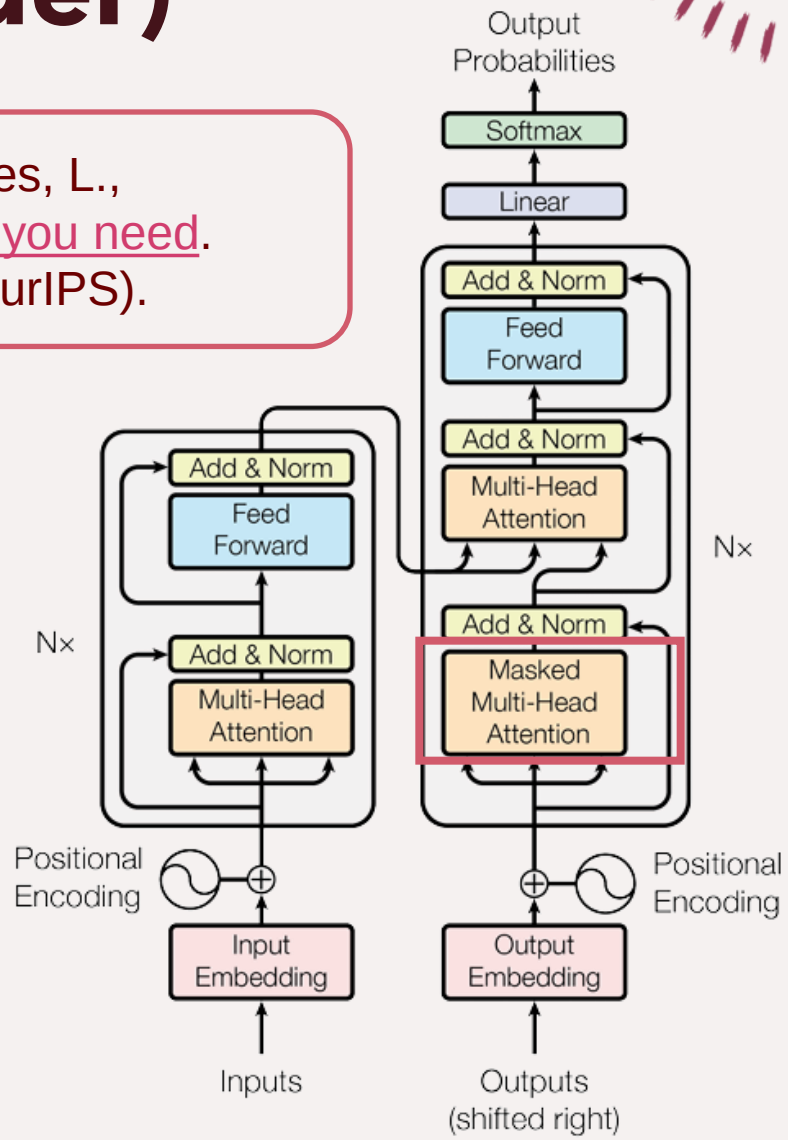
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems (NeurIPS).

- Future token masking

1. Compute self-attention scores between all pairs of target sequence tokens
   - Query matrix: $\mathbf{Q} = \mathbf{X}\,\mathbf{W}^{\mathbf{Q}}$, $\mathbf{W}^{\mathbf{Q}} \in \mathbb{R}^{d \times k}$
   - Key matrix: $\mathbf{K} = \mathbf{X}\,\mathbf{W}^{\mathbf{K}}$, $\mathbf{W}^{\mathbf{K}} \in \mathbb{R}^{d \times k}$

Compute $\mathbf{A} = \frac{\mathbf{Q}\mathbf{K}^{T}}{\sqrt{k}}$

- This is the unnormalized self-attention matrix



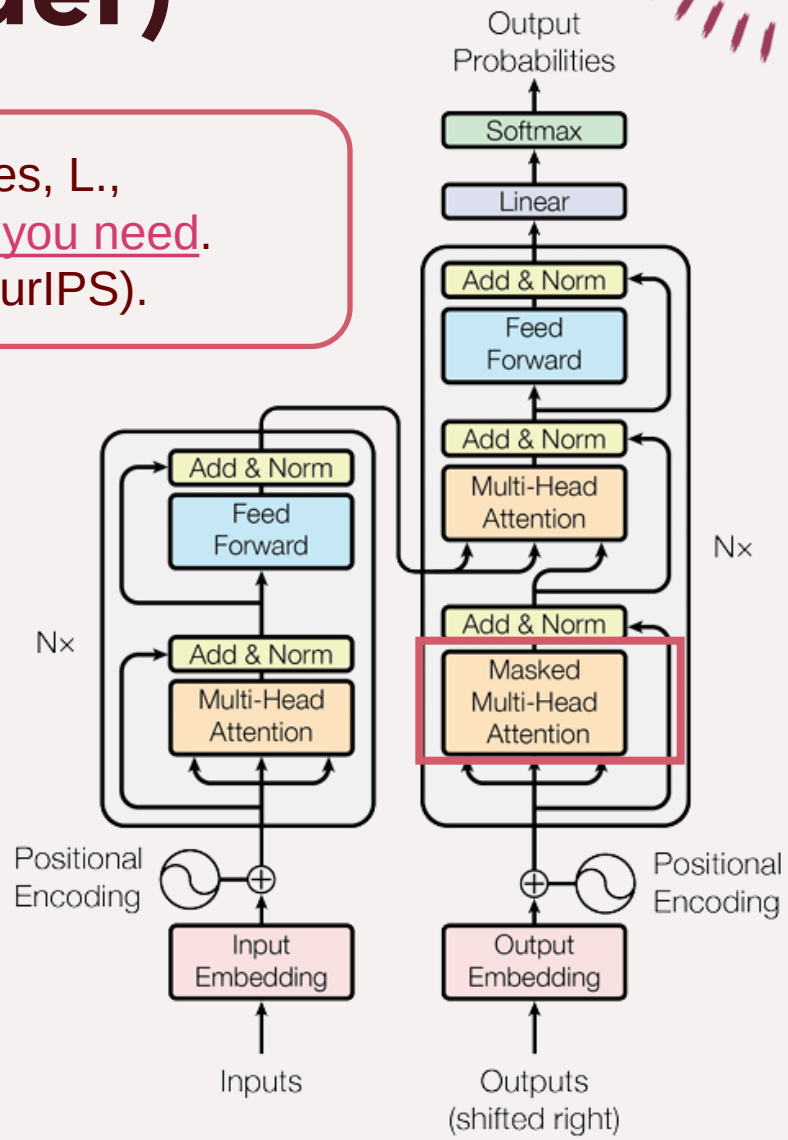
Transformer (Encoder-Decoder)

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems (NeurIPS).

- Future token masking

2. Set the values in the upper right triangle of the $\mathbf{A} = \frac{\mathbf{QK}^T}{\sqrt{k}}$ matrix to –infinity. Q: Why?

	<s>	kleiner	weißer	Hund
<s>	a_{11}	-∞	-∞	-∞
kleiner	a_{21}	a_{22}	-∞	-∞
weißer	a_{31}	a_{32}	a_{33}	-∞
Hund	a_{41}	a_{42}	a_{43}	a_{44}

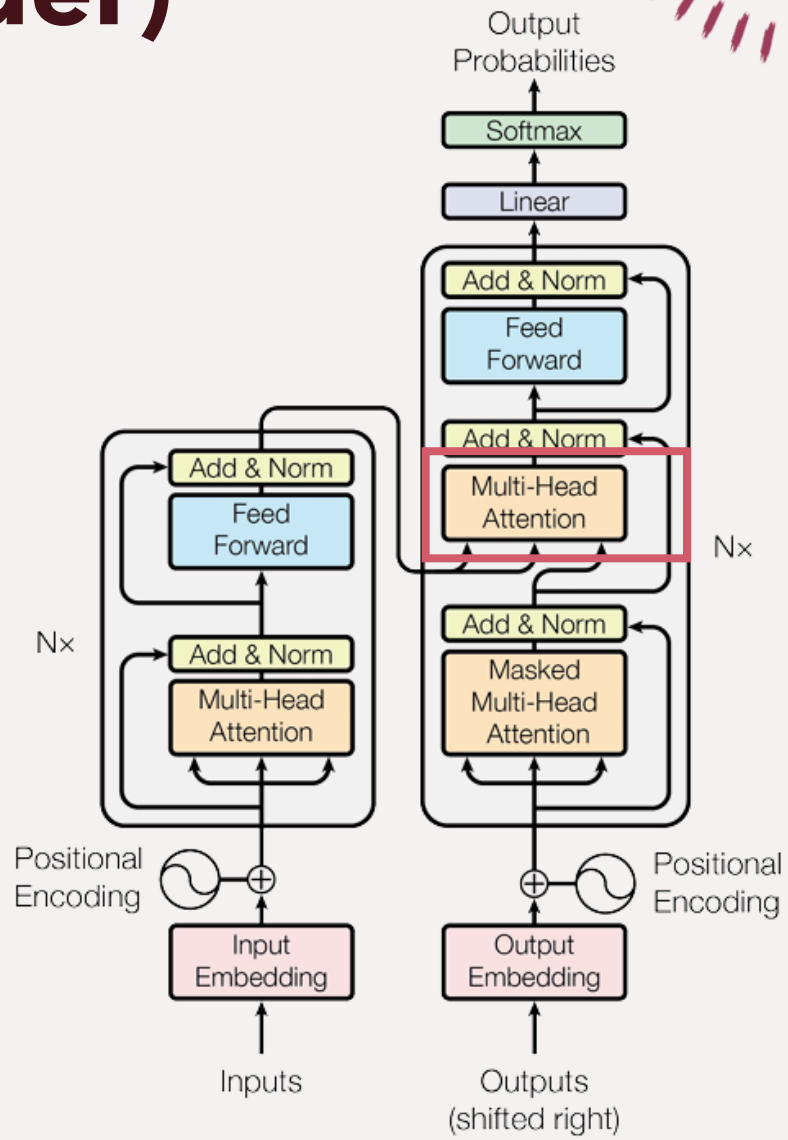


Transformer (Encoder-Decoder)

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems (NeurIPS).

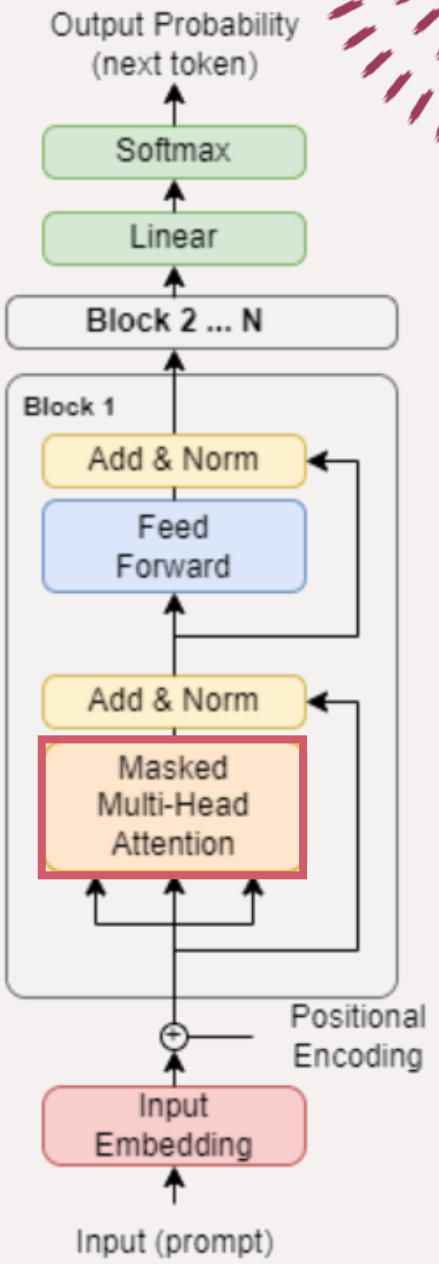
- Multi-head cross-attention
  - Target tokens attend over source tokens
  - Keys and values $\mathbf{K}_{src}$ and $\mathbf{V}_{src}$ computed for source tokens
  - Queries $\mathbf{Q}_{trg}$ computed from target tokens

$$\mathbf{Z} = \text{softmax}(\frac{\mathbf{Q}_{trg}\mathbf{K}_{src}^{T}}{\sqrt{k}})\mathbf{V}_{src}$$



Decoder-Only Transformer

- **Decoder-only** Transformer is very similar to encoder only Transformer (covered in Lecture 4)
  - All tokens (input and output) are treated equally
    - No cross-attention,
    - Only self-attention
  - But we're training for generation
    - At generation, tokens can't see future tokens
    - So, we use masked self-attention



Pretraining Transformer Models

- In Lecture 4, we've seen that we pretrain encoder-only Transformers via bidirectional masked LM-ing
  - BERT & co. → good for language understanding tasks
  - Not suitable for language generation (out of the box)

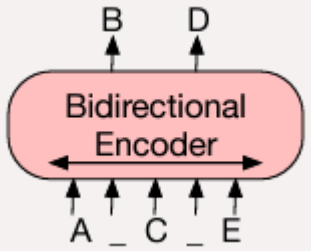
- Q: How do we pretrain decoder-only Transformers?
  - Via **autoregressive LM-ing:** predict next word in text
  - GPT, Mistral, Claude, Llama, DeepSeek, ...

- Q: How do we pretrain encoder-decoder Transformers?
  - Q: What would be the pretraining objective?
  - Q: What would such pretraining be useful for?

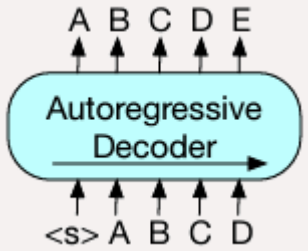
Pretraining Encoder-Decoder Models

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... & Zettlemoyer, L. (2020). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp. 7871-7880).

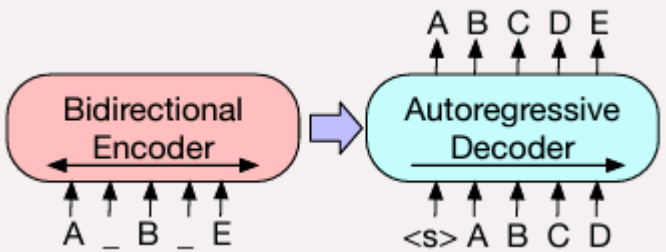
- BART: pretraining an encoder-decoder Transformer by means of various denoising self-supervised pretraining objectives
- Different from both autoregressive (GPT) and masked (BERT) LM-ing



BERT



GPT

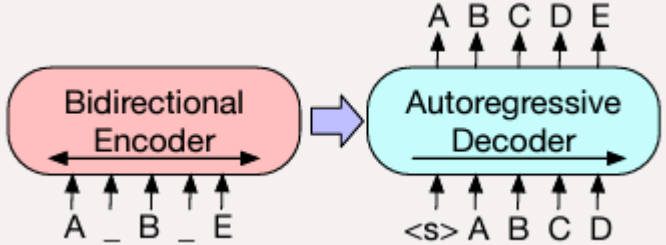


BART

Pretraining Encoder-Decoder Models

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... & Zettlemoyer, L. (2020). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp. 7871-7880).

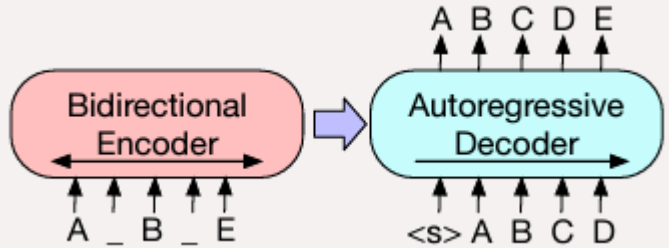
- BART allows for arbitrary corruption of input:
  - Token masking (same as BERT)
  - Token deletion
  - Text infilling (aka span masking)
    - Whole span replaced with one [MASK] token
  - Sentence permutation (within document)
  - Document rotation
    - Around a randomly selected token
  - ...



Fine-Tuning Encoder-Decoder Models

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... & Zettlemoyer, L. (2020). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp. 7871-7880).

- Fine-tuning BART
  1. Text generation tasks – primary use case
     - Normal sequence-to-sequence training
  2. Sequence/token classification tasks
     - (Same) input fed to both enc and dec
     - Output of dec (last layer) fed to classifier
     - For seq. class: special token appended to the end of the input sequence

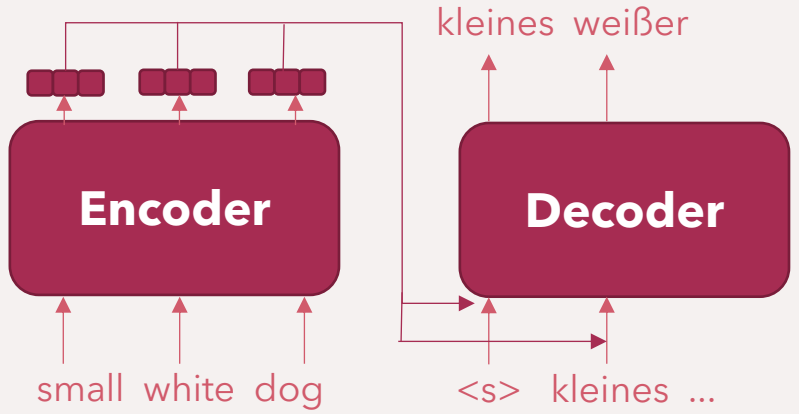


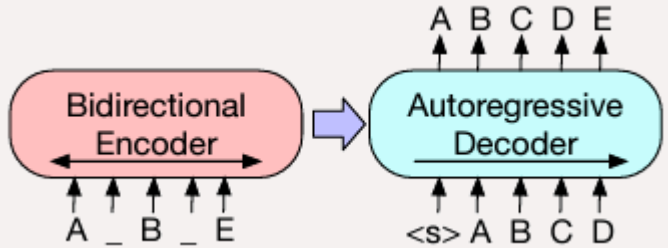
- Better results if input to enc is corrupted (as in pretraining)

Pretraining Encoder-Decoder Models

Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., ... & Zettlemoyer, L. (2020). Multilingual denoising pre-training for neural machine translation. Transactions of the Association for Computational Linguistics, 8, 726-742.

- Multilingual BART (mBART)
  - BART pretraining on concatenated corpora from 25 languages
  - The complete multilingual Encoder-Decoder Transformer is pretrained
  - After pre-training, fine-tuned with parallel data for MT



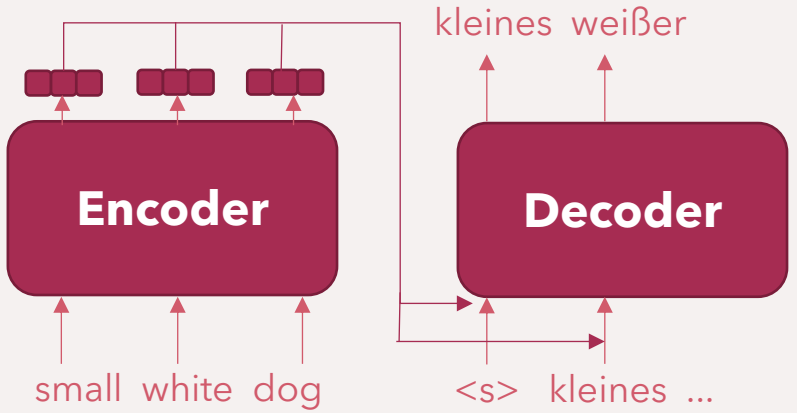


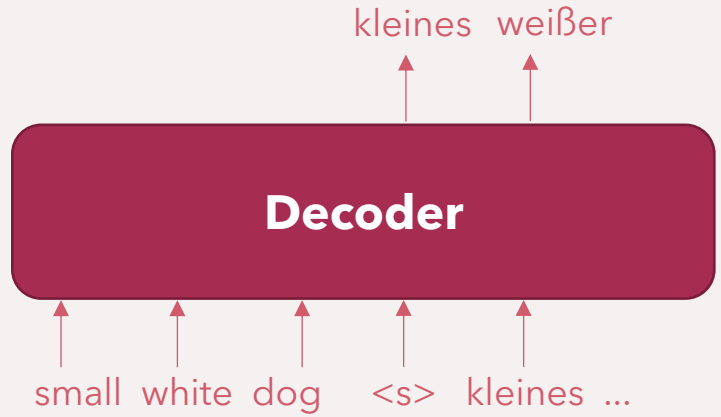
Content

- Text Generation
- Encoder-Decoder vs. Decoder-Only Models
- **Decoding Strategies**
- Multilingual Machine Translation
- Evaluating Text Generation

Decoding at Inference

- At inference, tokens generated one by one
  - Applies to both decoder-only and encoder-decoder models
  - Q: is taking the most likely token at each step the best strategy?
    - Q: will it lead to globally most likely generation?
  - Q: Can we do exact (full, complete) search and evaluate probability of all sequences under the model?





Decoding at Inference

- Decoding problem: given a language (generation) model

$$P(y_i \mid y_1, y_2, \ldots, y_{i-1}; x, \theta)$$

- Find the sequence of tokens $y_1 \ldots y_T$ with the largest $P(y_1 \ldots y_T \mid x)$
  - x is the input text (e.g., source language sentence in MT)
  - θ are the parameters of the text generation model

$$y'_1 \cdots y'_T = \textit{argmax}_{y1 \ldots yT} P(y_1 \ldots y_T \mid x, \theta)$$
$$= \textit{argmax}_{y1 \ldots yT} P(y_1 \mid x) * P(y_2 \mid y_1; x, \theta) * \ldots * P(y_T \mid y_1, y_2, \ldots, y_{T-1}; x, \theta)$$

- Exact (full, complete) search: assume target vocabulary of size $|V|$
  - T tokens → compute the above probability for $|V|^T$ sequences
  - Intractable!

Decoding Strategies

- **Greedy decoding**: select the most likely token at each step

$$y'_1 = argmax_{y1}\, P(y_1 \mid x, \theta)$$
$$y'_2 = argmax_{y2}\, P(y_2 \mid y'_1; x, \theta)$$
$$\ldots$$
$$y'_T = argmax_{yT}\, P(y_T \mid y'_1, y'_2, \ldots, y'_{T-1}; x, \theta)$$

- While the model (e.g., decoder-only Transformer) itself considers the entire preceeding context at each generation step, decoding doesn't
  - Generally leads to repetitive generations, especially for longer sequences
  - Less of an issue for larger decoders (LLMs) that can semantically accurately represent long preceding sequences

Decoding Strategies

- **Random sampling**: at each step, select the token by randomly selecting from the probability distribution of that step

  $y'_i \rightarrow$ randomly sample from $P(y_i \mid y'_1, y'_2, \ldots, y'_{i-1}; x, \theta)$

  - More likely tokens according to the model (θ) have higher chance of being sampled → still produces (most often) meaningful text
    - But less repetitive than with greedy decoding

- Top-k random sampling: sample over only the k most likely tokens according to $P(y_i \mid y'_1, y'_2, \ldots, y'_{i-1}; x, \theta)$
  - Tradeoff between quality of generation and repetition
  - Q: What is top-1 random sampling equivalent to?

Decoding Strategies

- **Beam Search**: A heuristic search algorithm that expands the most promising sequences found so far
  - At any step (generated sequence length) keeps only k best solutions of that length
  - k, the „beam width", defines the „breadth" of the search

- Step 1: keep k best of |V| possible choices for y_1
- Step 2: keep k best of k * |V| possible choices for y_1y_2
  - For each of the k most likely tokens y_1 from step 1, we evaluate all |V| tokens for y_2
- ...
- Step T: keep k best (or one) of k * |V| possible choices for $y_1y_2...y_{T-1}y_T$
  - For each of the k most likely sequences $y_1y_2...y_{T-1}$ from step T-1, we evaluate all |V| tokens for y_T

Decoding Strategies

Beam Search

- Step *i*: keep k of k * |V| possible choices for $y_1 y_2 \ldots y_{i-1}$
  - For each of the k most likely sequences $y_1 y_2 \ldots y_{i-1}$ from step i-1, we evaluate all |V| tokens for y_i

- Q: What is the score we compute for evaluated sequence $y_1 \ldots y_i$?
  - $P(y_1 \mid x) * P(y_2 \mid y_1; x,\theta) * \ldots * P(y_i \mid y_1, y_2, \ldots, y_{i-1}; x,\theta)$
    - For long(er) sequences this could lead to underflow
  - Thus apply log: $\log P(y_1 \mid x) + \log P(y_2 \mid y_1; x,\theta) + \ldots + \log P(y_i \mid y_1, y_2, \ldots, y_{i-1}; x,\theta)$

- Q: How many sequences does beam search with width k evaluate?
  - Step 1: |V|, Step 2…T: k*|V| → k*|V|*(T-1) + |V| ≈ k*|V|*T
  - k*|V|*T still much smaller than $|V|^T$ (exact/full search)

Decoding Strategies

Beam Search: illustration (k = 2)

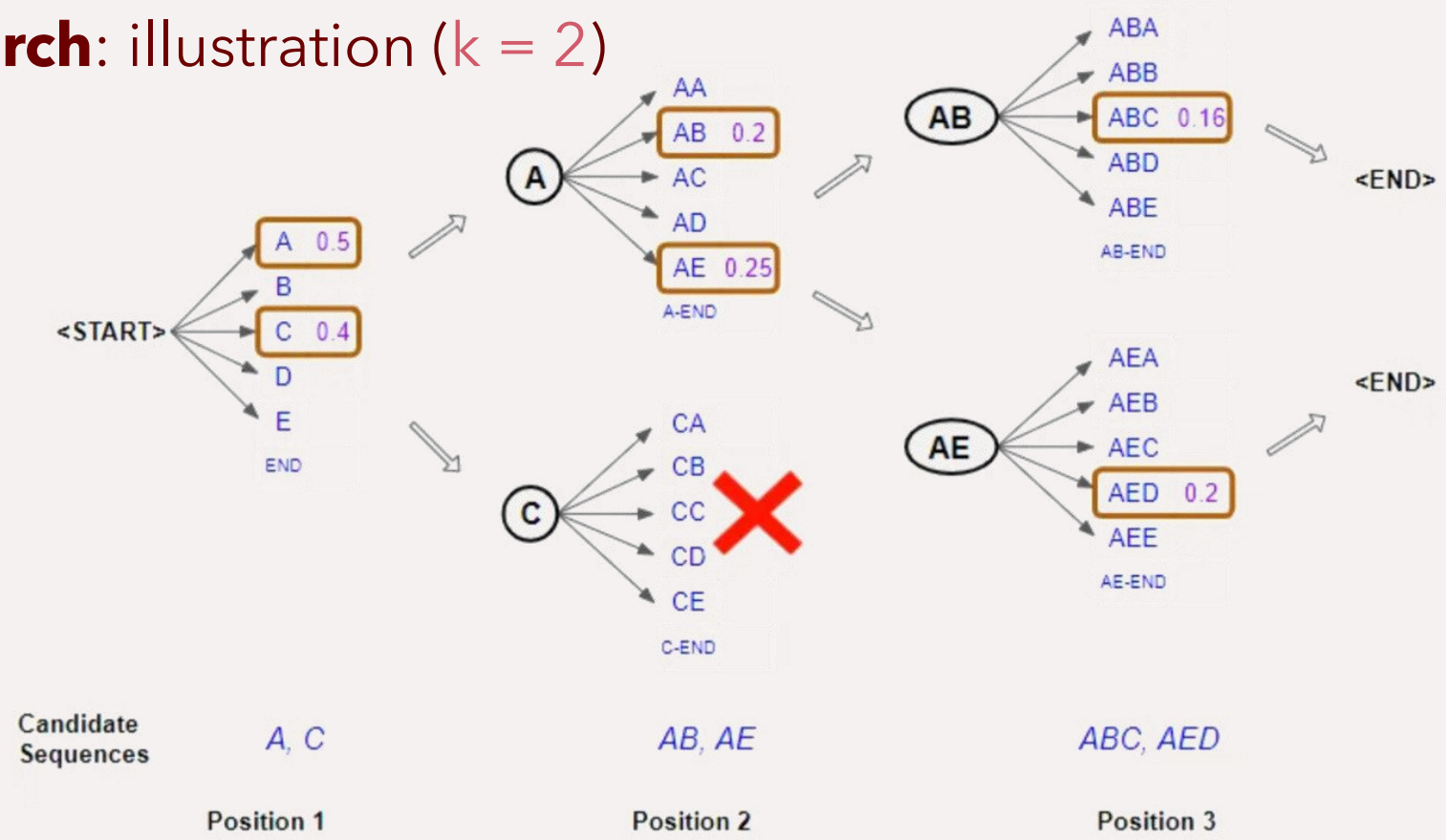


Image from: https://towardsdatascience.com/foundations-of-nlp-explained-visually-beam-search-how-it-works-1586b9849a24

Content

- Text Generation
- Encoder-Decoder vs. Decoder-Only Models
- Decoding Strategies
- **Multilingual Machine Translation**
- Evaluating Text Generation

Multilingual Machine Translation

- Traditionally, (S)MT models were trained for concrete language pairs
  - Dedicated MT model for each language pair and translation direction
- **Multilingual MT**: one model that supports multiple languages and translation directions
- Q: Why multilingual MT models?
  - Training MT models requires parallel data
  - For many language pairs there is no (sufficiently large) parallel data
  - Multilingual MT training can lead to positive cross-lingual transfer
    - Generalization of MT for „unseen" language combinations (no parallel data), and even „unseen" languages (unseen in MT training)
    - E.g., we have plenty EN-ES and ES-QU parallel data, but no EN-QU
  - Hardware (memory) limitations at inference: one vs. many models

Multilingual Machine Translation

- Pivot translation: multilingual MT before pretraining neural LMs
  - Based on pivoting: pivot language (typically EN) is the language that has parallel corpora with most other languages
  - E.g., if we have EN→X and X→EN translation models, then L1→L2 translation amounts to pipelining L1→EN and EN→L2
  - Pipeline of two translation models: errors more likely (they propagate)
- Multilingual source models (N-to-1)
  - Single model that translates N languages to one target language of interest
- Multilingual target models (1-to-N)
  - Single model that translates one source language of interest to N target languages
- Fully multilingual MT (N-to-N)
  - Single model for translation from any of the N languages to any other

Multilingual Machine Translation

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., ... & Dean, J. (2017). Google's multilingual neural machine translation system: Enabling zero-shot translation. Transactions of the Association for Computational Linguistics, 5, 339-351.

- GNMT: First noteworthy (successful) effort in N-to-N translation
- Architecture: an Encoder-Decoder (with recurrent enc and dec)
  - Inverted token order of the input source language text
  - Last token of encoder input indicates the target language (e.g., <2de> for DE)

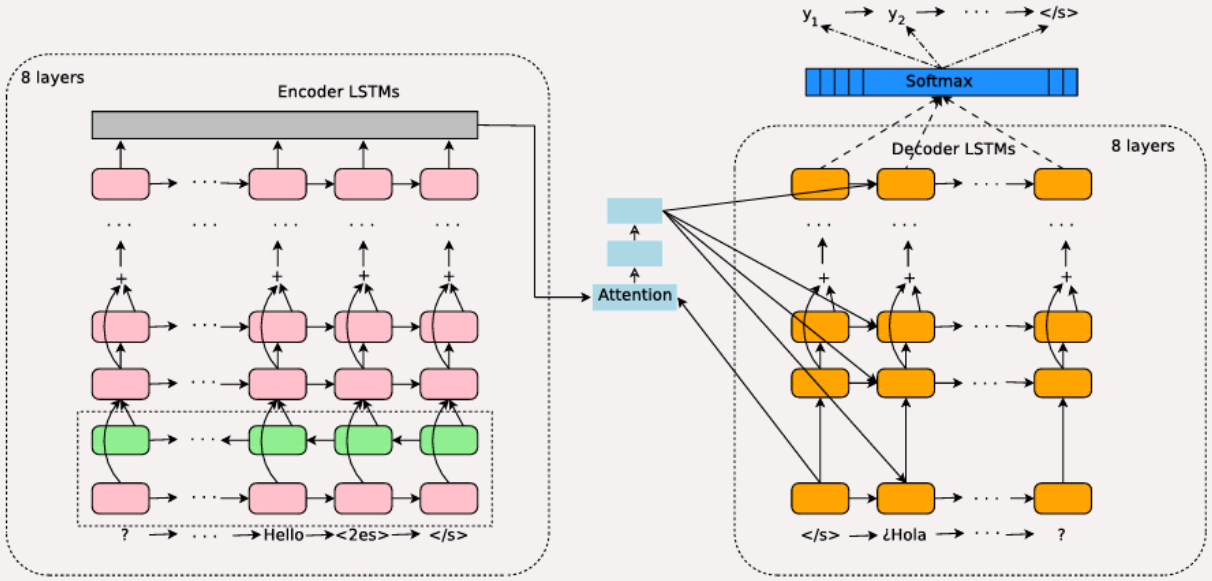
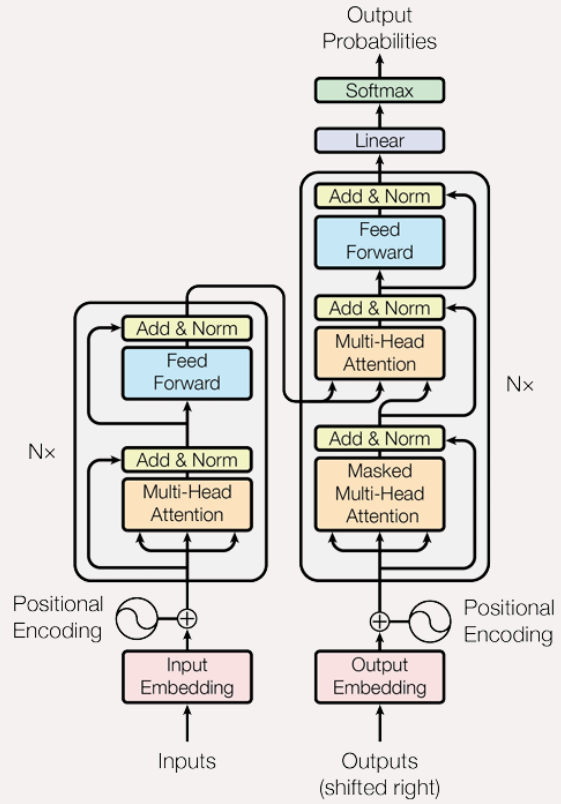


Image from Johnson et al.

Multilingual Machine Translation

Aharoni, R., Johnson, M., & Firat, O. (2019, June). Massively Multilingual Neural Machine Translation. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (pp. 3874-3884).

- Replace the recurrent encoder and decoder with the Encoder-Decoder Transformer
  - Parallel data encompassing 58 languages
  - But only EN-X and X-EN (116 pairs in total)
  - Like before, last token of encoder input indicates the target language (e.g., <2es> for ES)
    - This facilitates N-to-N translation at inference, including language pairs not seen in training (i.e., without English)
- Encoder-Decoder Transformer still trained from scratch only for MT (i.e., not pretrained in any way)



Multilingual Machine Translation

Tang, Y., Tran, C., Li, X., Chen, P. J., Goyal, N., Chaudhary, V., ... & Fan, A. (2021, August). Multilingual translation from denoising pre-training. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021 (pp. 3450-3466).

- In cross-lingual transfer for language understanding, we relied on pretrained MMTs (e.g., mBERT, XLM-R)
- Q: Could NMT also benefit if we start the MT training from some pretrained multilingual model?
- Q: From what pretrained multilingual model to start?
  - Q: mBERT, XLM-R? These are encoder-only Transformers
  - mBART: this is a multilingually pretrained Encoder-Decoder Transformer
    - Tang et al. pre-train a new mBART model for 50 languages
- So, (1) massively multilingual self-supervised pretraining of Enc-Dec + (2) massively multilingual MT training (fine-tuning)

Multilingual Machine Translation

Tang, Y., Tran, C., Li, X., Chen, P. J., Goyal, N., Chaudhary, V., ... & Fan, A. (2021, August). Multilingual translation from denoising pre-training. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021 (pp. 3450-3466).

(1) massively multilingual self-supervised pretraining of Enc-Dec
- mBART pretraining for 50 languages

(2) massively multilingual MT training (fine-tuning)
- parallel data between 50 languages

- Q: How important is pre-training (given that MT data is large)?
- Q: Multilingual vs. bilingual MT fine-tuning?
  - Positive transfer effects vs. curse of multilinguality?
  - Q: Does it depend on the „resourceness" of the language?

Multilingual Machine Translation

Tang, Y., Tran, C., Li, X., Chen, P. J., Goyal, N., Chaudhary, V., ... & Fan, A. (2021, August). Multilingual translation from denoising pre-training. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021 (pp. 3450-3466).

Data	Multilingual FT Translation to English						Multilingual FT Translation from English					
ΔBLEU	Bilingual		Bilingual FT		Multilingual SC		Bilingual		Bilingual FT		Multilingual SC	
	M→1	M↔M	M→1	M↔M	M→1	M↔M	1→M	M↔M	1→M	M↔M	1→M	M↔M
>10M	2.4	0.2	0.7	-1.6	1.1	-0.5	-0.3	-1.5	-2.1	-3.3	0.2	0
1M-10M	6.2	4.4	2.3	0.5	1.4	0.3	1.7	0.6	-1.6	-2.7	0.2	-0.4
100k-1M	8.0	7.3	2.4	1.6	2.5	0.4	4.0	3.2	-0.4	-1.2	-0.1	-0.3
10-100K	22.3	20.7	5.5	3.8	4.4	2.3	13.5	13.7	0.1	0.32	-0.2	-0.3
4-10k	18.9	15.0	7.3	3.4	5.8	0.9	10.0	9.7	1.3	1.00	-0.7	-1.2
All	12.0	10.3	3.5	1.8	3.1	-0.1	6.3	5.8	-0.5	-1.0	-0.1	-0.4

Table 2: Multilingual Finetuning on 50 languages comparing to 3 baselines: (1) bilingual from scratch, (2) bilingual finetuning, and (3) multilingual training from scratch. Multilingual Finetuning **(a)** consistently improves over all baselines for translation into English (left), while **(b)** performs similarly over bilingual finetuning and multilingual from scratch with significant improvement over bilingual from scratch for translation from English (right). Numbers are average *BLEU difference* between multilingual finetuning models and the corresponding baselines. Per direction comparison is available in Figure 1.

Table of results from Tang et al.

Content

- Text Generation
- Encoder-Decoder vs. Decoder-Only Models
- Decoding Strategies
- Multilingual Machine Translation
- **Evaluating Text Generation**

Evaluating Text Generation

- **Crucial question**: how good is the generated text?
- Q: What does good mean? Remember our definition of text generation
  - Tasks that require generation/creation of text that in some aspect **conforms** to the provided (text) input
- Two main types of evaluation for text generation:
  1. **Reference-based** evaluation: generated text is compared against „gold standard" (i.e., manual, human) text, e.g.,
     - Human translation in MT,
     - Human-written summary in summarization
     - Q: Many different generations (e.g., translations / summaries) could be judged as good – how do we know the one provided as „reference" is the best?
     - Multi-reference evaluation: comparison against multiple reference texts

Evaluating Text Generation

- **Crucial question**: how good is the generated text?
- Q: What does good mean? Remember our definition of text generation
  - Tasks that require generation/creation of text that in some aspect **conforms** to the provided (text) input
- Two main types of evaluation for text generation:

2. **Reference-free** evaluation: there is no reference text
   - Creating references (e.g., summaries) is time-consuming and expensive
   - Reference-free evaluation measures estimate the quality of the generation by comparing it directly with the corresponding input texts
     - MT: generated L_T translation compared against the input L_S text
     - Summarization: generated summary compared against the input long text

Evaluating Text Generation

- **Evaluation measures**
  - Traditional symbolic metrics
    - Based on word-overlap between the generation and reference (in reference-based evaluation) or input (in reference-free evaluation)
    - Examples: the (in)famous BLEU for MT or ROUGE for summarization
    - Shortcoming is that generated text can be:
      - good even if it has low term overlap with the reference and
      - bad even if it has high term overlap
  - Semantic metrics
    - Compare the meaning of the generated text and the meaning of the reference (in reference-based evaluation) or input (in reference-free evaluation)

Evaluating Text Generation (MT)

- BLEU is (still) the most commonly used reference-based evaluation measure in MT
  - Product of two scores: geometric precision (gp) and brevity penalty (bp)
  - Precision: proportion of n-grams in the generation accounted for in the reference
    - Generation: *the big black big dog is black*
    - Reference: *the big dog is black*
    - Precision = 7 / 7 = 1?
  - Clipped precision: no repetition, each reference token can be matched at most once
    - For the above example: $p_1 = 5/7$
  - First component of BLEU is the geometric average of clipped precision for 1-grams (p_1), 2-grams (p_2), 3-grams (p_3), and 4-grams (p_4)
    - $gp = (p_1)^{1/4} * (p_2)^{1/4} * (p_3)^{1/4} * (p_4)^{1/4}$
  - Second component: brevity penalty
    - Accounts for shorter texts a priori being more likely to have better precision
    - bp = 1 if g (length of generation) > r (length of reference), else $e^{(1-r/g)}$

Evaluating Text Generation (MT)

- BLEU is based on token overlap
  - As such it can both penalize good generations as well as reward bad ones
  - E.g., reference: *they utilize the tool of dark color*
    good translation (but low BLEU)*: they use the dark utensil*
    bad translation (but high BLEU)*: they don't utilize the tool of dark color*

- Q: Why is BLEU then still used as the primary evaluation metric in MT?
  - Proven to have high correlation with human estimates of translation quality
  - Simple, easily interpretable, and computationally inexpensive
  - Errors at individual sentences „cancel out" at the level of evaluation corpus
    - Some good sentence translations will be (incorrectly) punished
    - Other bad translations will be (incorrectly) rewarded
    - Thus average BLEU on the corpus will be an ok estimate of translation quality ☺

Evaluating Text Generation (MT)

- Semantic evaluation: we compare semantic representations (i.e., embeddings) of generated tokens against reference tokens
  - But token alignments are not always 1-to-1, they are often M-to-N
    - E.g., „dark tool" – „utensil of dark color"
  - Assume two sets of embeddings: for generated tokens $\mathbf{G} = \{\mathbf{g}_1, \mathbf{g}_2, ..., \mathbf{g}_M\}$ and for reference/input tokens $\mathbf{R} = \{\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_N\}$
    - These can be static word embeddings, or contextualized obtained with some pre-trained LM (e.g., BERT)
    - „Unsupervised" semantic metrics typically compare the two sets of embeddings

Evaluating Text Generation (MT)

Kusner, M., Sun, Y., Kolkin, N., & Weinberger, K. (2015, June). From word embeddings to document distances. In International conference on machine learning (pp. 957-966). PMLR.

- Assume two sets of embeddings: for generated tokens $\mathbf{G} = \{\mathbf{g_1}, \mathbf{g_2}, \ldots, \mathbf{g_M}\}$ and for reference/input tokens $\mathbf{R} = \{\mathbf{r_1}, \mathbf{r_2}, \ldots, \mathbf{r_N}\}$
- Word (Earth) Mover Distance (WMD): casts the task of measuring semantic distance/similarity of two texts as the optimal transport problem between the two corresponding sets of embeddings **G** and **R**
- We have the similarity matrix $\mathbf{S} \in \mathbb{R}^{M \times N}$ that contains pairwise similarity scores
  - S_{ij} is the similarity (e.g., cosine similarity) between $\mathbf{g_i}$ and $\mathbf{r_j}$
- We're looking for an (optimal) alignment matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$:

$$\mathbf{A}^* = \text{argmax}_{\mathbf{A}} \sum_{i,j} A_{ij} S_{ij,}$$

with constraints $\sum_j A_{ij} = 1$ and $\sum_i A_{ij} = 1$

Evaluating Text Generation (MT)

Kusner, M., Sun, Y., Kolkin, N., & Weinberger, K. (2015, June). From word embeddings to document distances. In International conference on machine learning (pp. 957-966). PMLR.

- WMD: the similarity matrix $\mathbf{S} \in \mathbb{R}^{M \times N}$ contains pairwise similarity scores
  - S_{ij} is the similarity (e.g., cosine similarity) between $\mathbf{g_i}$ and $\mathbf{r_j}$
- We're looking for an (optimal) alignment matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$:

$$\mathbf{A}^* = \text{argmax}_{\mathbf{A}} \sum_{i,j} A_{ij} S_{ij,}$$

with constraints $\sum_j A_{ij} = 1$ and $\sum_i A_{ij} = 1$

- With the most efficient algorithm for solving the optimal transport problem, WMD has complexity $O(K^3 \, log K)$; K = number of unique words in both texts
  - This can be prohibitive for long texts
  - Still ok for sentence-level MT evaluation

Evaluating Text Generation (MT)

Zhao, W., Glavaš, G., Peyrard, M., Gao, Y., West, R., & Eger, S. (2020, July). On the Limitations of Cross-lingual Encoders as Exposed by Reference-Free Machine Translation Evaluation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp. 1656-1671).

- Reference-free MT evaluation with WMD
  - If no reference translation, we have to compare the generated translation in L_T against the input text in L_S
  - This means we must have a bilingual L_S-L_T word embedding space
    - Q: How do we obtain those?
    - E.g., with pretrained MMTs (e.g., mBERT or XLM-R)
  - WMD on top of mBERT embeddings good for MT evaluation for translation between closely related languages, but not for distant languages
    - Rewards „Translationese"
    - But it can be combined with a measure of likelihood of the generated text in the target language (as measured by some LM of the target language)

Image: Alexander Mikhalchyk