Neural Machine Translation for German-English Translation

Thomas Nilsson
Course: 02456 Deep Learning
DTU Compute, Technical University of Denmark

Neural Machine Translation

Neural machine translation (NMT): is behind the big recent leap in Machine Translation. NMT takes a deep learning approach to aligning sentences and extracting dependencies between them.

Seq2Seq: Predicting a *target* sequence given a *source* sequence. In this case translating a German sentence to English

Scoring: BLEU [1] is a method for automatic evaluation of Machine Translation.
The BLEU-score compares a candidate sentence to its reference by counting the N-gram overlaps, up to *N=4*. There are a few ways the BLEU score can be calculated through smoothing. The vanilla BLEU score is a weighted product of the number of N-gram overlaps as well as a brevity penalty applied to the candidate sentence.

$$BLEU = min\left(1, \frac{len(candidate)}{len(reference)}\right) \cdot \left(\prod_{n=1}^{4} P_n\right)^{1/4}$$

Dataset and data processing

Multi30k [2]: A collection of German-English pairs describing images found on Flickr. Available through `TorchText.`

Vocabulary: Multi30k contains **31,000** sentence pairs. Only words occurring more than twice were used. This results in a German vocabulary of **7855** words, and an English vocabulary of **5893** words. All out of vocabulary words are mapped to the `<unk>` token.
All sentence have a starting tag `<sos>` and an ending tag `<eos>` appended to them, in order to delimit the sentence.

Batching and padding: Sentence pairs are organized into batches. Sentences are padded with a `<pad>` token such that all sentences within a batch have the same length. This is necessary for the batch to be represented as a matrix.

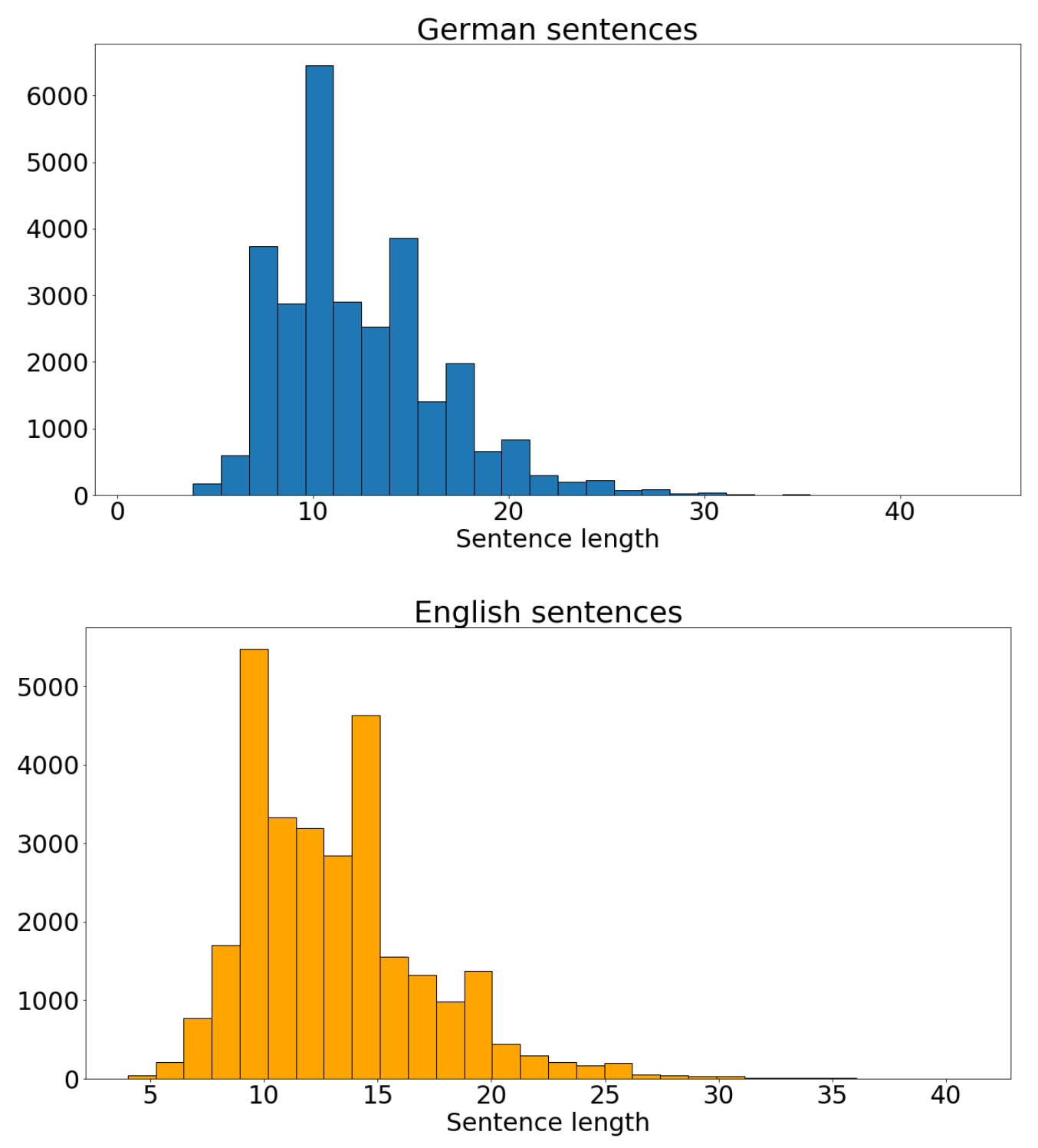


Figure 1: Sentence length distributions for the German- and English sentences. The dataset contains sentences of many different lengths, which may impact the BLEU scoring of the results since longer sentences are very hard to translate word for word compared to a reference.

RNN Encoder-Decoder

Main idea: Proposed by Cho et al. 2014 [3] The model encodes a variable-length sequence into a fixed-length context vector and then decodes this vector back into a variable-length sequence.

Encoder: Iterates over the input sequence and at each time step modifies the encoder hidden state which represents the sequence read thus far, which is a fixed-size representation.

Decoder: Unpacks the context vector and produces a token for each time step the network runs. At each timestep an output token of the target language (en) is produced. Stop unpacking, either when generating the `<eos>` token, or after a fixed number of time-steps.

Bottleneck problem: All the information in the sentence is encapsulated by the final hidden state, which leads to early tokens being forgotten. The **attention mechanism [4]** solves this by looking at the alignment of the current decoder hidden state and each encoder hidden state. If two hidden states are aligned (large dot product) then the are likely semantically related.

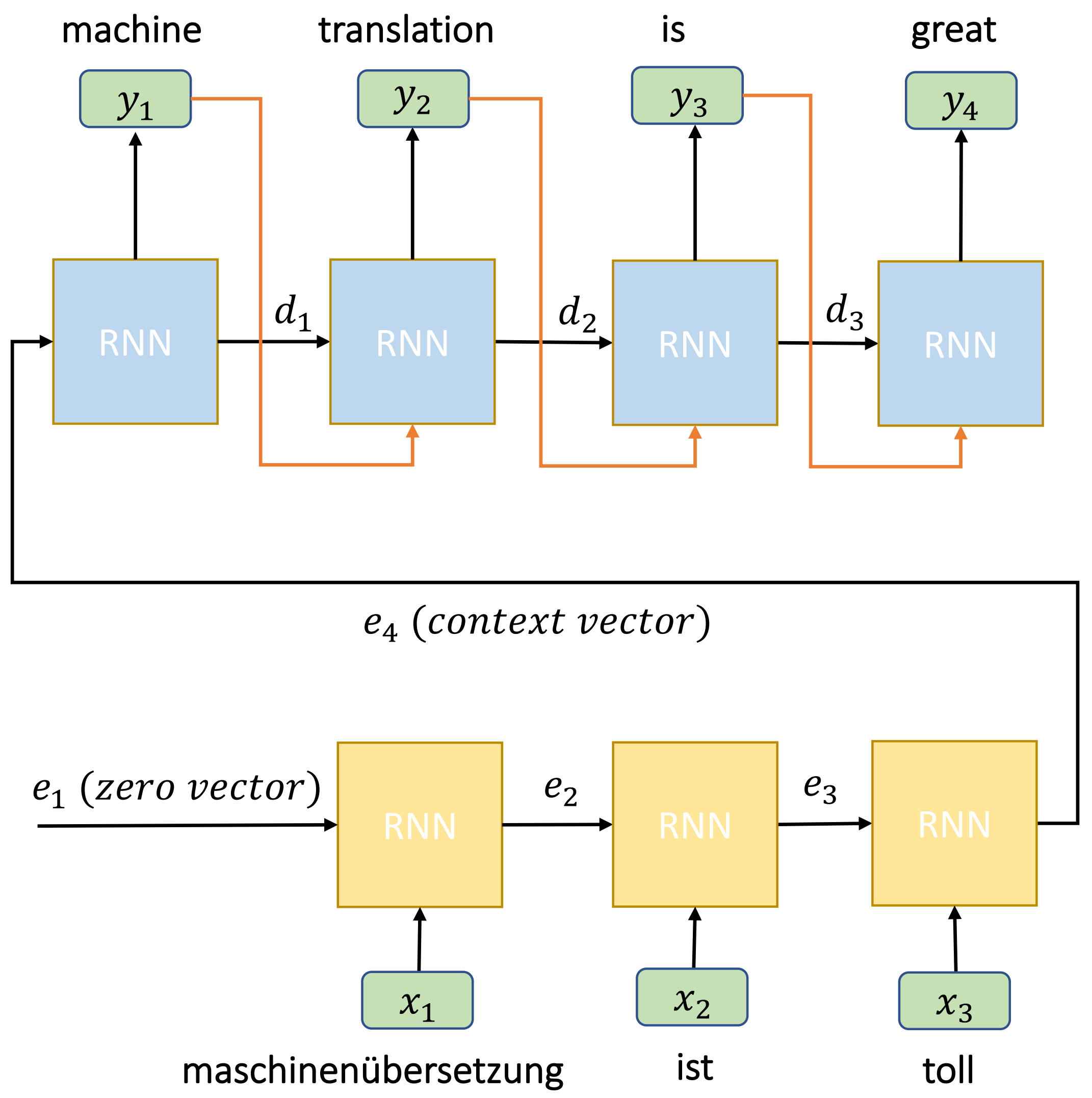


Figure 2: The RNN Encoder-Decoder model architecture for a German-English sentence pair which contains a different number of tokens. For decoding the next token, the previous decoder output is used, this is illustrated with the orange arrows. Teacher forcing may alternatively be used, where the target token at a given timestep is fed to the decoder.

Transformer

Main idea: In 2017 it was proposed by Google Brain and Google Research in 2017 [5] to scrap the recurrence, and instead let attention solve perform the translation alone. The Transformer model is heavily parallelized since there is no sequential bottleneck like in RNNs and is therefore much faster to train.

Self-attention: Calculates attention score between sequence elements, i.e. an attention score is calculated between each element in the sequence. This attention mechanism is employed in parallel, and in total 8 attention-heads are used in the original paper for the Transformer.

Parallel input: Due to the input being read all at once, the input order must be provided. This is the **positional encoding** for which sine and cosine are used. In addition, **masked-attention** is used in the decoder to prevent revealing future words to the model, at the current 'time step'.

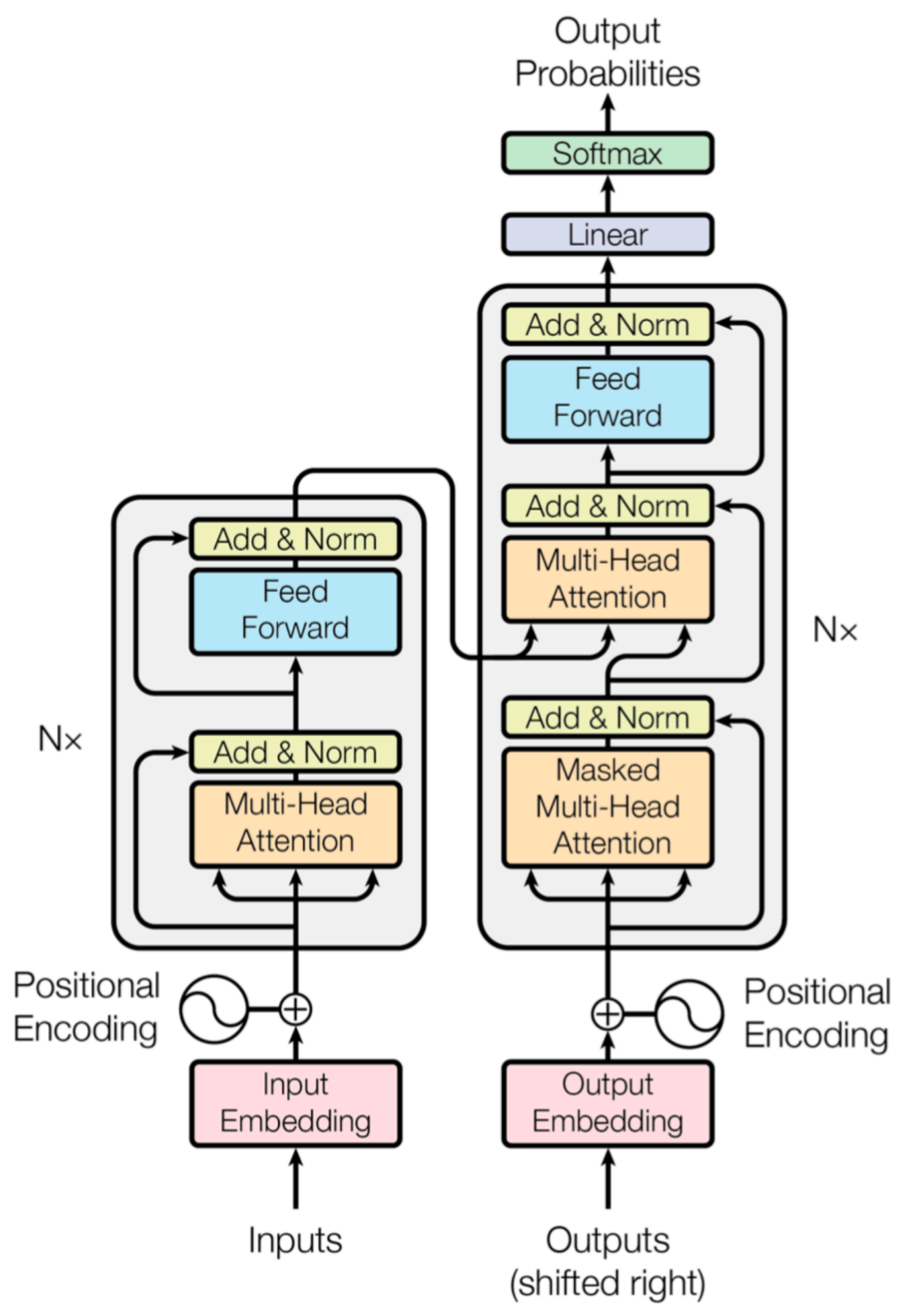


Figure 3: Transformer. Encoding and decoding happens all at once. The Transformer uses several encoder-decoder layers which is marked by the 'Nx' label, N meaning the number of stacked layers.

Training

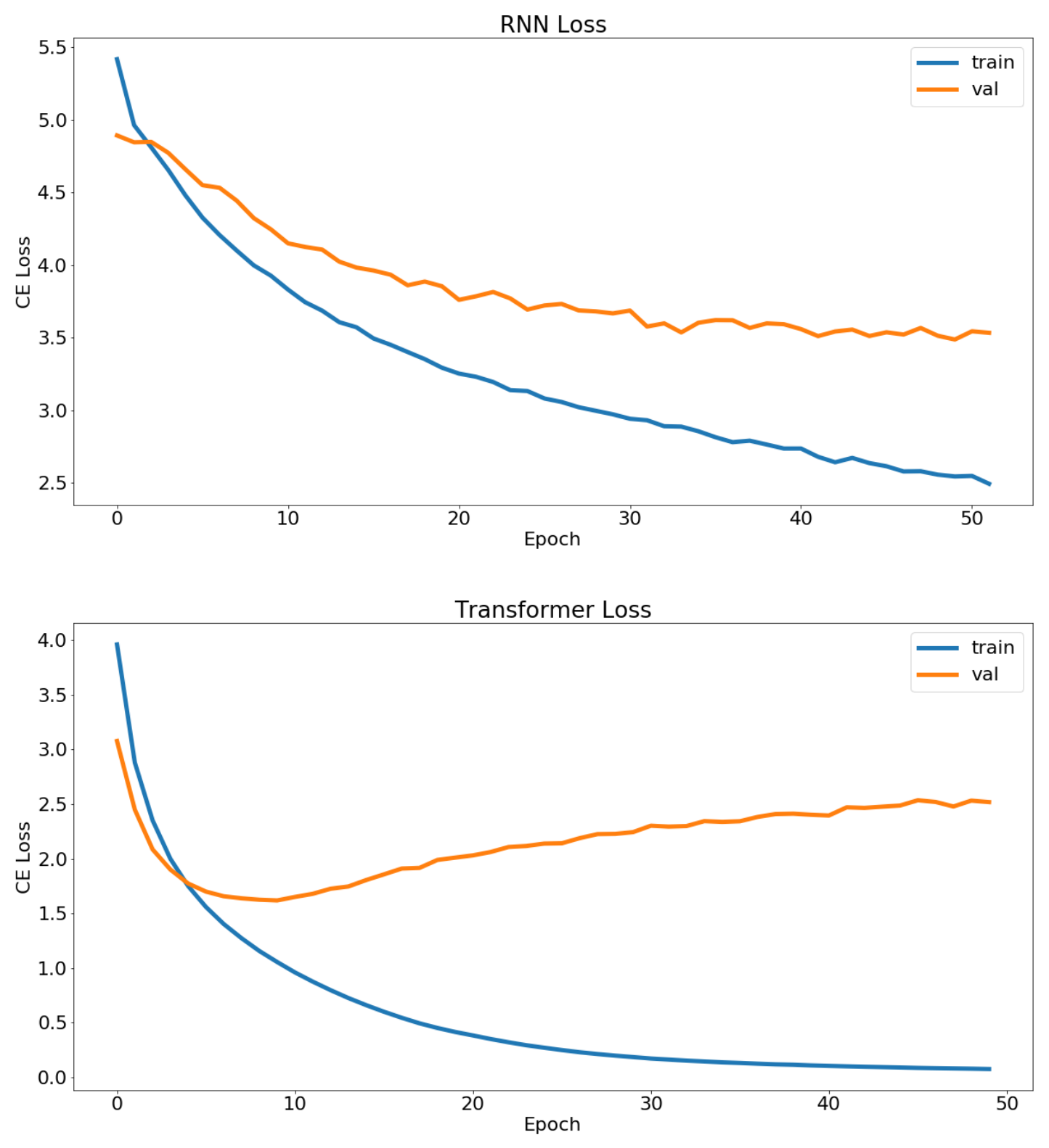
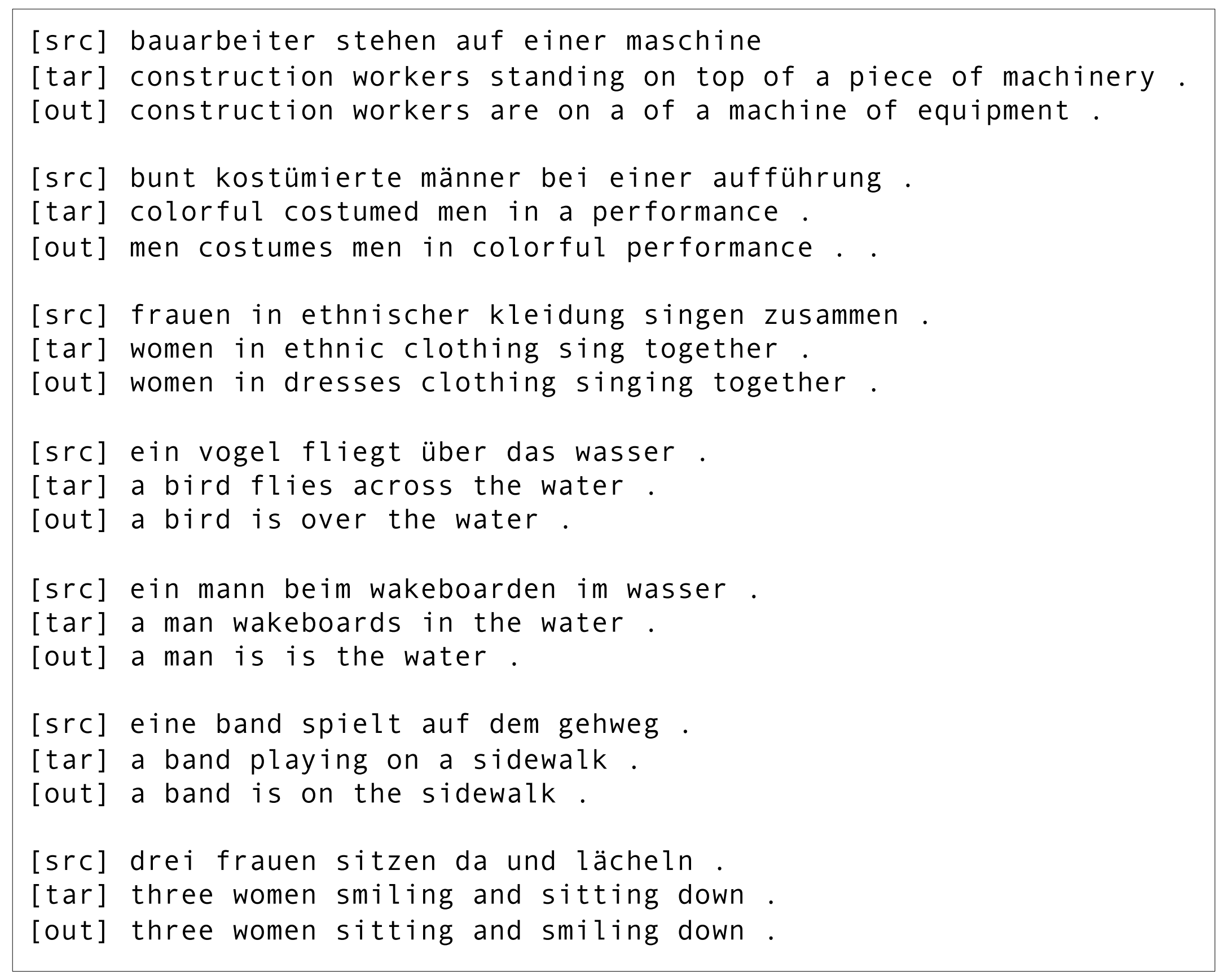


Figure 4: Training and validation loss for the RNN- and the Transformer model, respectively. The RNN-based model performs quite poorly, and the Transformer overfits after 10 epochs, but has a much lower loss than the RNN model.

Results - Qualitative



```
[src] bauarbeiter stehen auf einer maschine
[tar] construction workers standing on top of a piece of machinery .
[out] construction workers are on a of a machine of equipment .

[src] bunt kostümierte männer bei einer aufführung .
[tar] colorful costumed men in a performance .
[out] men costumes men in colorful performance . .

[src] frauen in ethnischer kleidung singen zusammen .
[tar] women in ethnic clothing sing together .
[out] women in dresses clothing singing together .

[src] ein vogel fliegt über das wasser .
[tar] a bird flies across the water .
[out] a bird is over the water .

[src] ein mann beim wakeboarden im wasser .
[tar] a man wakeboards in the water .
[out] a man is is the water .

[src] eine band spielt auf dem gehweg .
[tar] a band playing on a sidewalk .
[out] a band is on the sidewalk .

[src] drei frauen sitzen da und lächeln .
[tar] three women smiling and sitting down .
[out] three women sitting and smiling down .
```

Figure 5: Examples of translated sentences from the validation set, translated by the Transformer model.

Results - Quantitative

Model	Params	Min. Val Loss	Time per epoch	BLEU
RNN-Att	7.27M	3.48	290 s	9.83
Transformer	54.2M	1.62	95 s	28.56
Transformer-Benchmark [6]	?	?	?	35.41

References

[1] Papineni et al., BLEU: a Method for Automatic Evaluation of Machine Translation, ACL, 2002
[2] Elliott et al., Multi30K: Multilingual English-German Image Descriptions. CoRR, 2016
[3] Cho et al., Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation, CoRR, 2014
[4] Bahdanau et al., Neural Machine Translation by Jointly Learning to Align and Translate, CoRR, 2014
[5] Vaswani et al., Attention Is All You Need, CoRR, 2017
[6] Transformer Multi30k Reference, 35.41 BLEU, https://github.com/dmlc/dgl/tree/master/examples/pytorch/transformer