Chapter 11

Sequence-to-sequence Learning: Part 1

Natural language tasks often require mapping one arbitrary-length sequence into another. Unlike language modeling, where the goal is predicting the next word, here the focus is on tasks such as machine translation. The central idea is to take an input sentence in one language (e.g., English) and generate a corresponding sentence in another language (e.g., German). This requires models capable of handling variable-length inputs and outputs.

The primary approach involves encoder-decoder architectures, also called seq2seq models. The encoder processes the input sequence and compresses its meaning into a vector representation, while the decoder unfolds this representation into a target sequence.

Preparing the Dataset

A bilingual parallel corpus of English and German phrases is used. The data is in a tab-separated format with three columns: English phrase, German translation, and attribution. Only the first two are needed. The dataset contains over 227,000 examples, ranging from single-word pairs ("Go." → "Geh.") to sentences of fifty words or more.

Since the full dataset is large, a subset of 50,000 samples is randomly selected for efficiency. Two special tokens are added to the German translations:

sos (start of sentence) at the beginning

eos (end of sentence) at the end

For instance, "Grüß Gott!" becomes "sos Grüß Gott! eos." These tokens are critical when the model later generates translations autonomously.

Vocabulary and Sequence Length Analysis

Vocabulary statistics are computed by counting word frequencies. Only words appearing at least ten times are considered. Results show about 2,238 frequent English words and 2,497 frequent German words. Frequent tokens include names ("Tom"), common verbs ("ist"), pronouns ("Ich," "du"), and the special sos/eos tokens.

Sequence length statistics reveal that English sentences have a median length of 6 words, while German sentences have a median of 8 words. To accommodate variations, the maximum sequence lengths are fixed at 19 for English and 21 for German.

Encoder-Decoder Architecture

The seq2seq model is structured into two parts:

1. Encoder — consumes the source sentence and produces a compact vector representation (often called a context vector or "thought vector").
2. Decoder — starts from this context vector and generates the target sentence word by word.

Figure 1 illustrates this architecture, showing how English inputs are compressed into a context vector that drives German output generation.

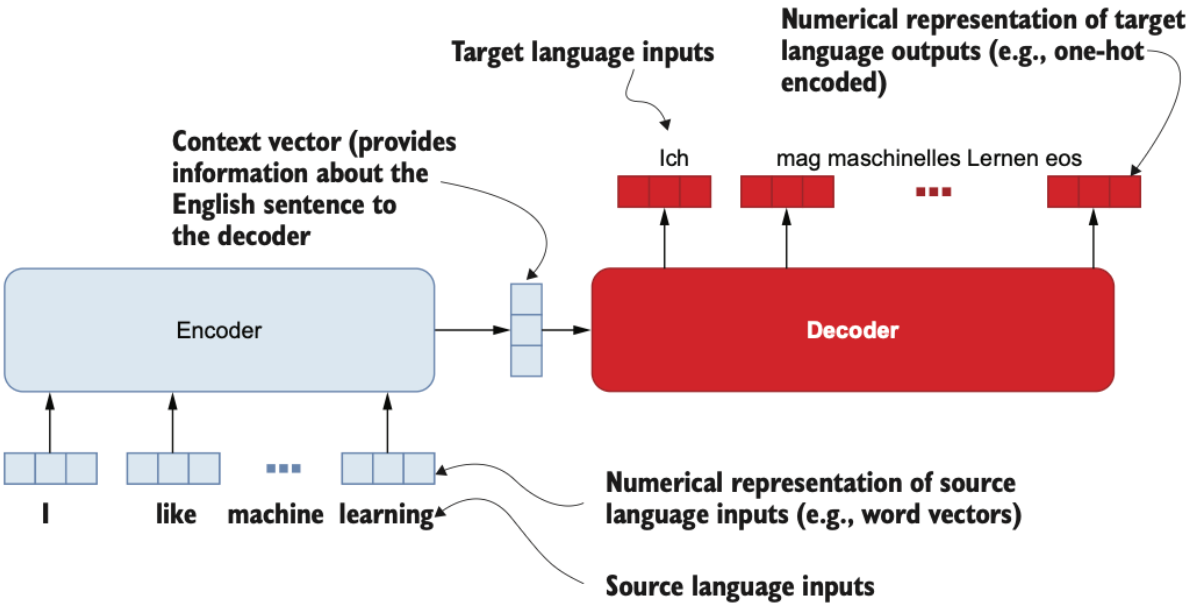

Figure 1 High-level components of the encoder-decoder architecture in the context of machine translation

The encoder is implemented using a bidirectional GRU network, which processes input both forward and backward. This design captures dependencies that may appear later in a sentence (e.g., distinguishing between "bank" as a financial institution or riverbank).

The decoder is also GRU-based but unidirectional, since it must generate output step by step. It includes dense layers that project the GRU outputs into probability distributions over the target vocabulary. Figure 2 depicts this, with recurrent layers followed by shared dense layers across time steps.

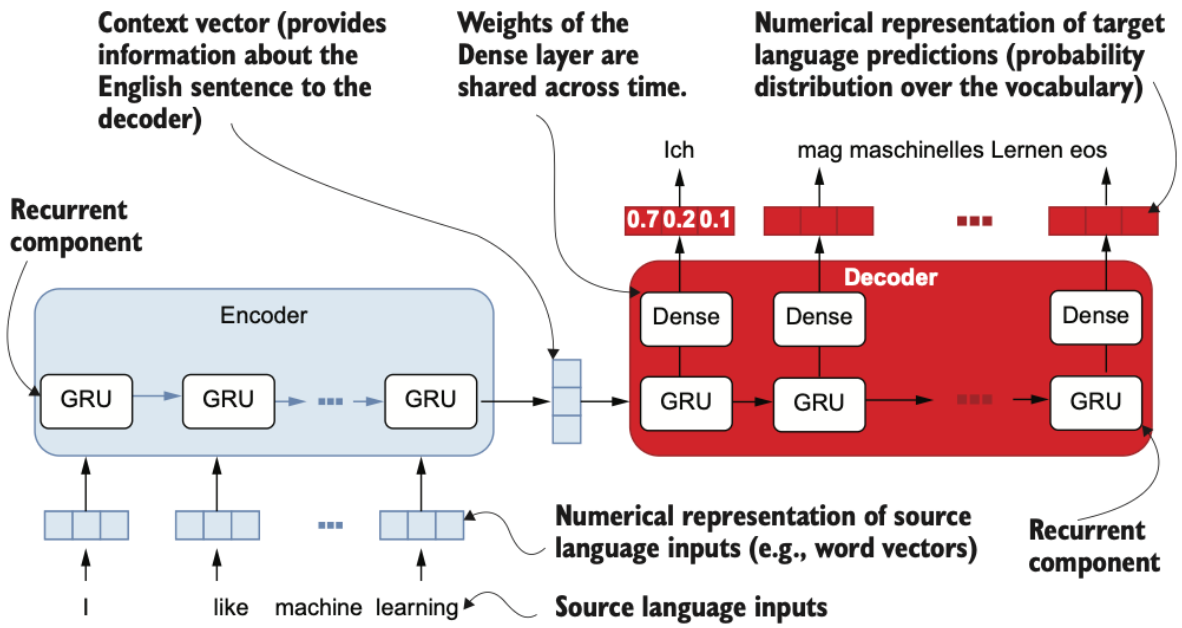

Figure 2 Specific components in the encoder and decoder modules. The encoder has a GRU layer, and the decoder consists of a GRU layer followed by one or more Dense layers

Integrating Text Vectorization

Traditionally, preprocessing converts text to numeric IDs before feeding data into the model. Instead, this design uses TextVectorization layers directly within the model. These layers tokenize text, map tokens to IDs, and handle out-of-vocabulary tokens with a special [UNK] ID. They also support output modes such as integer IDs, binary indicators, counts, or TF-IDF values.

Separate vectorizers are trained for English and German corpora. Once fitted, they can transform raw input strings like “run” or “how are you” into padded ID sequences. This makes the system more end-to-end, requiring only raw strings as input.

Building the Encoder

The encoder pipeline consists of:

- Input layer: receives raw English text strings.
- Vectorization layer: converts text to token IDs.
- Embedding layer: maps IDs to 128-dimensional word vectors.
- Bidirectional GRU: processes embeddings in both directions, producing the context vector as output.

The context vector is the last hidden state, encapsulating the meaning of the entire sentence. The encoder is wrapped as a Keras model.

Building the Decoder

The decoder receives both the German input sequence (shifted by one position for training) and the encoder's context vector. Its layers are:

- Input + vectorizer: processes the German sequence.
- Embedding layer: converts IDs to word vectors.
- GRU (256 units): initialized with the encoder's context vector as its hidden state.
- Dense hidden layer (512 units, ReLU): processes GRU outputs.
- Final dense layer with softmax: produces probability distributions over the German vocabulary.

This structure allows teacher forcing during training: the decoder is guided with the true previous word rather than its own prediction. Figure 3 shows how the encoder and decoder combine to form the full model.

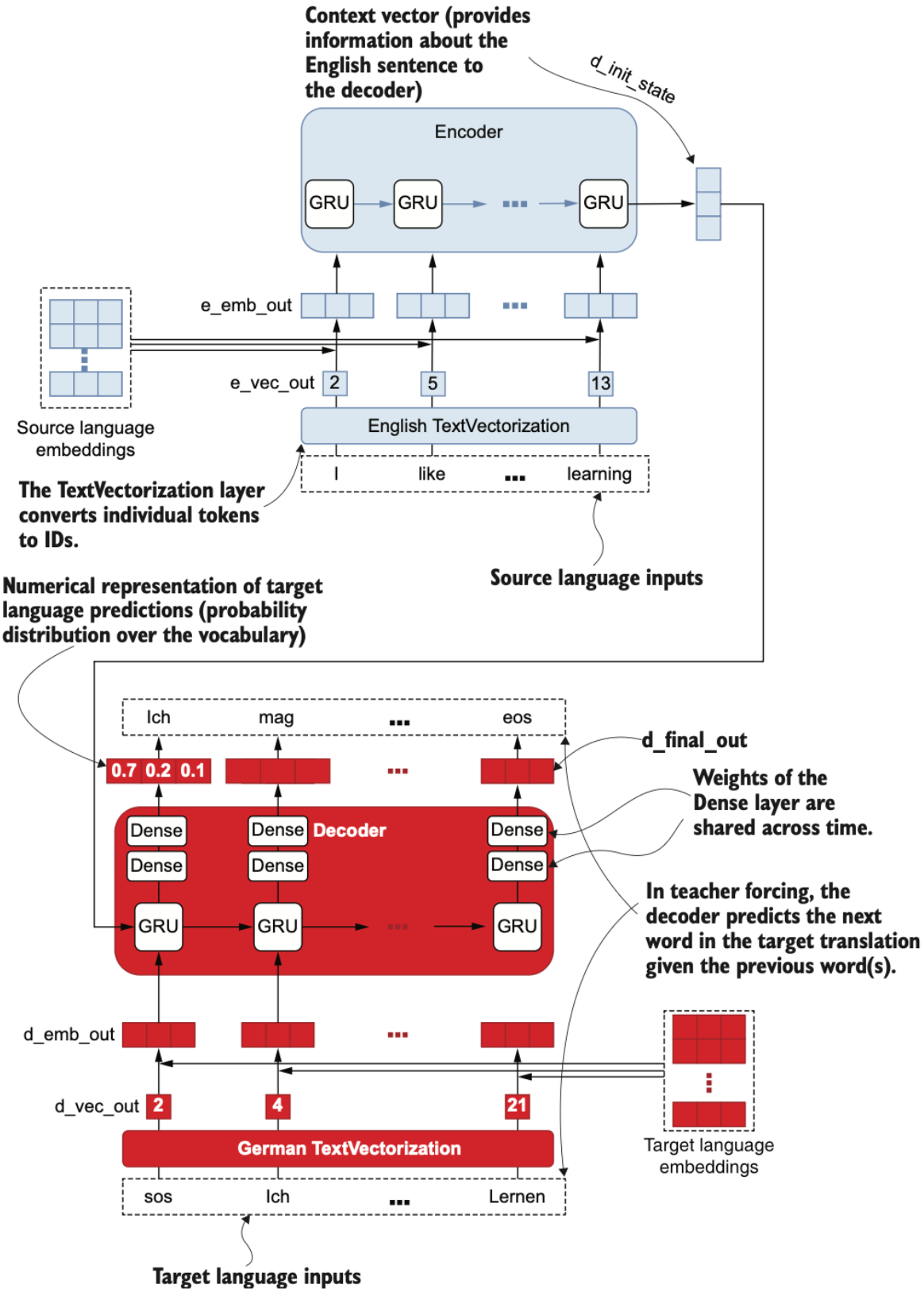
Context vector (provides information about the English sentence to the decoder)
d_init_state
Encoder
GRU
GRU
GRU
e_emb_out
e_vec_out
2
5
13
Source language embeddings
English TextVectorization
I
like
learning
The TextVectorization layer converts individual tokens to IDs.
Source language inputs
Numerical representation of target language predictions (probability distribution over the vocabulary)
Ich
mag
eos
d_final_out
0.7 0.2 0.1
Dense
Dense
Decoder
Weights of the Dense layer are shared across time.
GRU
In teacher forcing, the decoder predicts the next word in the target translation given the previous word(s).
d_emb_out
d_vec_out
2
4
21
German TextVectorization
Target language embeddings
sos
Ich
Lernen
Target language inputs

Figure 3 The implementation of the final sequence-to-sequence model with the focus on various layers and outputs involved

Model Compilation

The complete seq2seq model accepts two inputs (English and German sequences) and outputs predicted German tokens. It is compiled with:

- Loss: sparse categorical cross-entropy
- Optimizer: Adam
- Metric: accuracy

The model summary shows about 2.5 million trainable parameters.

Training Process

Data preparation involves splitting sentences into:

- Encoder inputs: English sequences.
- Decoder inputs: German sequences without the last token.
- Decoder outputs: German sequences without the first token.

For example:

- Input: “I want a piece of chocolate cake.”
- German target: “Ich möchte ein Stück Schokoladenkuchen.”
- Decoder input: [“Ich,” “möchte,” “ein,” “Stück”]
- Decoder output: [“möchte,” “ein,” “Stück,” “Schokoladenkuchen”].

The data is shuffled each epoch to improve training.

Evaluation Metrics

Two main metrics are used:

- Accuracy — measures exact word-by-word matches.
- BLEU (Bilingual Evaluation Understudy) — evaluates translation quality by comparing n-grams with reference translations.

BLEU corrects shortcomings of precision by penalizing repetitive outputs and rewarding longer meaningful matches. For example, although “the cat the cat the cat the” has high unigram precision, BLEU assigns it a much lower score than a fluent, semantically correct sentence.