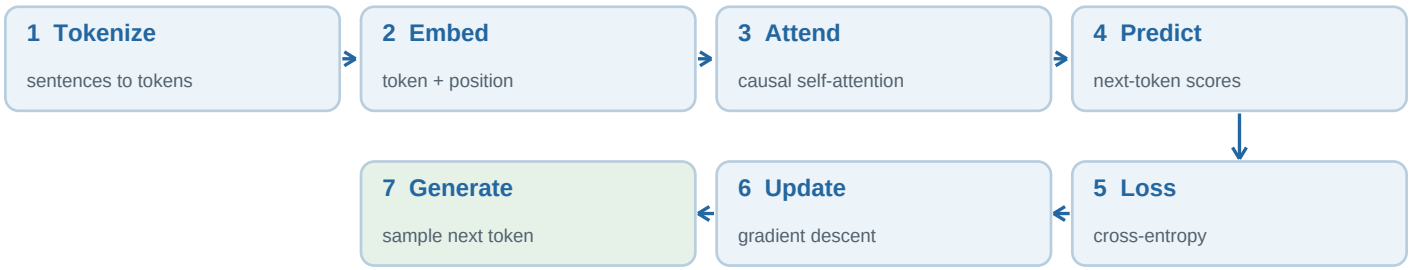
# How the Transformer Mechanism Works in AI-ANNE

**From example sentences to generated text | A compact, trainable causal self-attention model**

Prof. Dr. habil. Dennis Klinkhammer




*Training repeats steps 2–6 across shuffled windows and epochs; step 7 uses the trained parameters.*

**1 Tokenize example sentences and construct training pairs**

$$s = (w_1, \ldots, w_n, \langle \text{eos} \rangle), \quad x_t = w_t, \quad y_t = w_{t+1}$$

Lowercase words and punctuation are tokenized; padded, overlapping windows of length T provide next-token targets.

**2 Learn token and position embeddings**

$$X_t = E[\text{token}_t] + P_t, \quad E \in \mathbb{R}^{V \times d}, \quad P \in \mathbb{R}^{T \times d}$$

Trainable token vectors and learned position vectors are added; both begin with small random values.

**3 Compute causal multi-head self-attention**

$$A_h = \text{softmax}\left(\frac{X_h X_h^\top}{\sqrt{d_h}} + M\right)$$

Each head uses a slice of X, without separate Q/K/V projections. The causal mask M prevents attention to future tokens.

**4 Combine attended representations and predict tokens**

$$O = \text{concat}_h\,(A_h X_h), \quad z_t = O_t W + b, \quad p_t = \text{softmax}(z_t)$$

Attention combines earlier context; a trainable output matrix produces probabilities over V possible tokens.

**5 Measure prediction error**

$$\mathcal{L} = -\frac{1}{N} \sum_{t \in I} \log p_t[y_t], \quad I = \{t : y_t \neq \text{PAD}\}$$

Cross-entropy measures how unlikely the correct next token was; padded targets are ignored.

**6 Update trainable parameters**

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}, \quad \theta = \{E, P, W, b\}$$

Backpropagation and stochastic gradient descent update embeddings, positions, output weights and bias over repeated epochs.

**7 Generate a sentence, one token at a time**

$$w_{t+1} \sim \text{TopK/TopP}(\text{Adjust}(p(\cdot \mid w_1, \ldots, w_t), \tau))$$

A corpus-based opening starts generation; temperature, top-k/top-p and grammar rules guide sampling.

### Why it runs on modest hardware

Only four arrays are trained: **2Vd + Td + V** parameters. With default d = 14, T = 10 and vocabulary cap V = 60, that is at most **1,880 parameters**. One attention layer, short contexts and a tiny corpus allow local browser execution through Pyodide, without a GPU.




**Implementation note.** AI-ANNE is a pedagogical transformer-like model, not a full Transformer: it omits learned Q/K/V projections, feed-forward layers, layer normalization and residual connections; generation additionally applies corpus-derived grammar rules.