Minor in AI

Breaking Language Barriers

Machine Translation & Attention Mechanism

March 25, 2025

1 Machine Translation

Machine translation (MT) involves automatically translating text from one language to another. Traditional methods relied on statistical approaches, but with advancements in deep learning, sequence-to-sequence (seq2seq) models have been widely adopted. However, seq2seq models suffer from limitations in handling long-range dependencies, which led to the development of the **attention mechanism** and later, **transformers**.

2 BLEU Score

The **Bilingual Evaluation Understudy (BLEU)** score is a metric used to evaluate the quality of machine-translated text by comparing it to a set of reference translations. It is based on precision of n-grams and a brevity penalty to handle length mismatches.

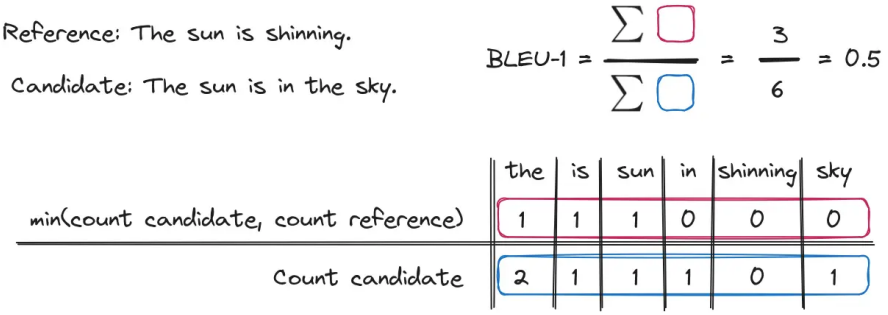

Figure 1: BLEU Score P.C.: Elastic

3 Attention Mechanism

Attention allows a model to focus on relevant parts of the input at different time steps, overcoming limitations of fixed-length encoding in RNNs and LSTMs.

Attention mechanisms help models focus on different parts of the input sequence when generating output. Traditional sequence models like RNNs and LSTMs process information sequentially, meaning earlier words might be forgotten as the sequence gets longer. Attention solves this issue by assigning different importance (weights) to different words in the input at each step of the output generation.

3.1 How Attention Works?

- Each input word gets a weight representing its relevance to the current output word.
- These weights are calculated using a scoring function.
- The model then takes a weighted sum of input representations to create a context vector.
- This context vector is used to generate the output.

3.2 Query, Key, and Value in Attention

The attention mechanism computes relevance scores between input words and output words using:

- **Query (Q)**: Represents the current word in the output sequence being generated.
- **Key (K)**: Represents each word in the input sequence.
- **Value (V)**: Represents the contextual representation of the input words.

3.3 Mathematical Formulation of Attention

Given an input sequence represented as key-value pairs (K, V) and a query Q, the attention mechanism computes a weighted sum of values based on the similarity of queries and keys. This is given by:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

where:

- $Q \in \mathbb{R}^{t \times d_k}$: Query matrix
- $K \in \mathbb{R}^{n \times d_k}$: Key matrix
- $V \in \mathbb{R}^{n \times d_v}$: Value matrix
- d_k: Dimension of the key vectors
- softmax ensures the weights sum to 1

3.4 Example: Translating "The cat is on the mat"

When translating to French, attention assigns higher weights to words in the source sentence that correspond to the target word being generated.

For instance, when generating "chat" in "Le chat est sur le tapis," attention focuses on "cat" with a high weight while other words receive lower weights.

3.5 Illustration of Attention Mechanism

Figure 2 visually represents how attention works. Given an input sentence, each word is embedded into a high-dimensional space. These embeddings are then transformed into queries, keys, and values using learned weight matrices. The dot-product similarity between queries and keys determines the attention scores, which are then passed through a softmax function to compute attention weights. These weights are used to obtain a weighted sum of values, which forms the attended output.

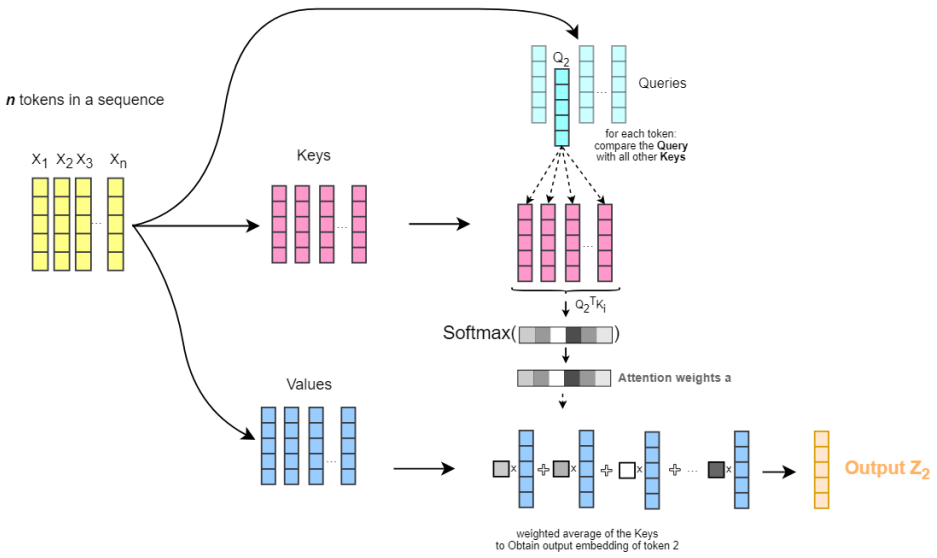

Figure 2: Illustration of Key, Query, and Value in Attention Mechanism

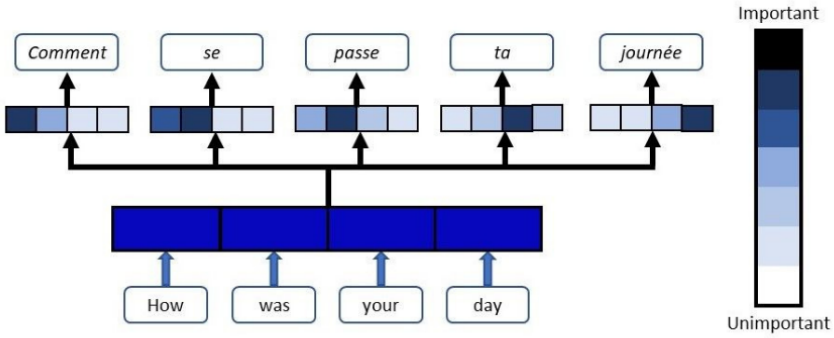

Figure 3: Attention Mechanism P.C : Teksands.AI

4 Transformers and Self-Attention

Transformers enhance performance by applying self-attention in a parallelized manner across input sequences. Instead of processing words sequentially, transformers apply attention mechanisms at multiple layers, making them more efficient.

Transformers have revolutionized natural language processing (NLP), particularly in tasks like machine translation. Unlike RNNs, transformers process entire sequences simultaneously using the self-attention mechanism.

4.1 Tokenization and Embeddings

Before feeding text into the model, words or subwords are tokenized into a sequence of indices. Each token is then converted into a vector using an embedding matrix:

$$X = E \cdot T, \tag{2}$$

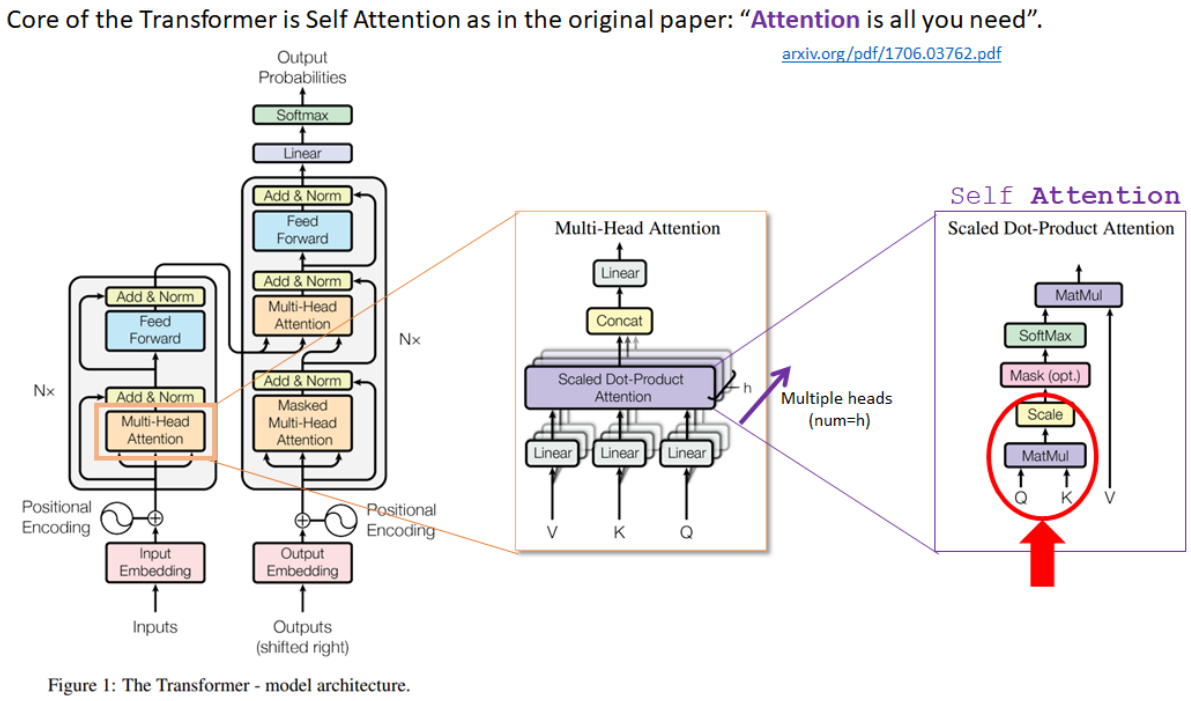

Figure 1: The Transformer - model architecture.

Figure 4: Transformer Architecture

where $E \in \mathbb{R}^{V \times d}$ is the embedding matrix, T is the tokenized input, and d is the embedding dimension.

4.2 Positional Encoding

Transformers lack recurrence, so positional encodings are added to retain order information:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d}}\right), \quad PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d}}\right), \tag{3}$$

where pos is the token position, and i is the dimension index.

4.3 Self-Attention Mechanism

Self-attention allows tokens to attend to all others. Each token is mapped to three vectors: Query (Q), Key (K), and Value (V), computed as:

$$Q = XW^Q, \quad K = XW^K, \quad V = XW^V, \tag{4}$$

where $W^Q, W^K, W^V \in \mathbb{R}^{d \times d}$ are trainable weight matrices.

Attention scores are computed using the scaled dot-product:

$$A = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V. \tag{5}$$

The softmax function ensures the sum of attention weights is 1, and scaling by $\sqrt{d}$ prevents large gradients.

4.4 Multi-Head Self-Attention

Instead of using a single attention mechanism, transformers use multiple attention heads to capture different aspects of the input:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_h)W^O \tag{6}$$

where each attention head is computed as:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{7}$$

for learnable weight matrices W_i^Q, W_i^K, W_i^V, W^O.

4.5 Feedforward Network

Each attention output is passed through a feedforward network:

$$FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2, \tag{8}$$

where W_1, W_2 and b_1, b_2 are trainable parameters.

4.6 Normalization and Residual Connections

Residual connections and layer normalization stabilize training:

$$X' = \text{LayerNorm}(X + \text{MultiHead}(X, X, X)), \tag{9}$$

$$X'' = \text{LayerNorm}(X' + \text{FFN}(X')). \tag{10}$$

5 Illustrative Example

Consider a simple translation task: translating "`I love cats`" from English to French ("`J'aime les chats`").

Step 1: Tokenization

The input sentence is tokenized as:

["`I`", "`love`", "`cats`"] $\rightarrow$ [1, 45, 302].

The target sentence is tokenized as:

["`J'aime`", "`les`", "`chats`"] $\rightarrow$ [90, 25, 400].

Step 2: Embeddings and Positional Encoding

The token indices are mapped to vectors using an embedding matrix, then positional encodings are added.

Step 3: Self-Attention Calculation

Queries, Keys, and Values are computed for each token. For example, for "love":

$$Q_{love} = X_{love}W^Q, \quad K_{love} = X_{love}W^K, \quad V_{love} = X_{love}W^V. \tag{11}$$

Attention scores determine how much "love" attends to other words:

$$A = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V. \tag{12}$$

Step 4: Multi-Head Attention and Feedforward Processing

Each token's representation is refined through multi-head attention and a feedforward network.

Step 5: Final Output

The decoder predicts the next token at each step, and softmax determines probabilities:

$$P(y|X) = \text{softmax}(XW^O). \tag{13}$$

The final output sequence is "J'aime les chats."

6 Key Takeaways

1. Attention mechanisms enhance sequence models by allowing dynamic focus on relevant words.
2. Self-attention improves efficiency by capturing global dependencies in input sequences.
3. Transformers outperform RNNs and LSTMs in machine translation by enabling parallel processing.
4. The BLEU score remains an important evaluation metric for translation models.
5. Multi-head attention enables the model to learn multiple aspects of the input context.