- In this paper, two simple and effective attentional mechanisms for neural machine translation were proposed: the global approach which always looks at all source positions and the local one that only attends to a subset of source positions at a time.We test the effectiveness of our models in the WMT translation tasks between English and German in both directions.

- The global attention approach gives a significant boost of +2.8 BLEU, making the proposed model slightly better than the base attentional system of Bahdanau et al. (2015) (row RNNSearch).

- When using the input feeding approach, another notable gain of +1.3 BLEU was observed.

- The local attention model with predictive alignments (row local-p) proves to be even better, giving a further improvement of +0.9 BLEU on top of the global attention model.

- **What is Global Attention ?**

  In global attention, the model considers all encoder hidden states when producing each word in the output (target sentence).

  At each decoding time step t, the decoder:

  1. Takes its current hidden state h□

  2. Compares it to all encoder hidden states h̄□ to compute a score

  3. Uses softmax over these scores to compute an alignment vector a□(s)

  4. Computes a context vector c□: a weighted sum of all encoder hidden states

Formula:

$$a_t(s) = \frac{\exp(\mathrm{score}(h_t, \bar{h}_s))}{\sum_{s'} \exp(\mathrm{score}(h_t, \bar{h}_{s'}))}$$

$$c_t = \sum_{s} a_t(s) \cdot \bar{h}_s$$

- **What is Local Attention ?**

Global attention is expensive (especially for long sequences). **Local attention** restricts the attention to a **small window** of encoder states around a predicted position.

1. Predict a position p_t in the source to focus on
2. Look at a window $[p_t - D, p_t + D]$
3. Compute attention weights only in that window
4. Compute context vector c_t as before

Two Variants:

a) Monotonic (local-m):

- Assume target aligns roughly linearly with source
- Set pt = t, i.e., source word t aligns with target word t

b) Predictive (local-p):

- Dynamically predict pt with a neural network:

$$p_t = S \cdot \mathrm{sigmoid}(v_p^T \tanh(W_p h_t))$$

(S = source sentence length)

- Then, apply a **Gaussian distribution** around p□ to weight attention:

$$a_t(s) = \mathrm{align}(h_t, \bar{h}_s) \cdot \exp\left(-\frac{(s - p_t)^2}{2\sigma^2}\right)$$

- **Input-Feeding Approach**

  In global/local attention, each alignment is made independently — but translation is **sequential** and prior decisions matter.

  Proposed Solution

  Feed the **previous attentional context** *c*□ back into the decoder **at the next time step**:

  - At time t+1, input = [previous context vector *c*□ ; word embedding]
  - Helps the model **remember past alignments**, mimicking the "coverage" in traditional MT

  Advantages:

  - Helps decoder keep track of which parts of the source have been covered
  - Creates a **deeper recurrent network** — better learning