- The models proposed recently for neural machine translation often belong to a family of encoder–decoders and encode a source sentence into a fixed-length vector from which a decoder generates a translation.

- Unlike the traditional phrase-based translation system (see, e.g., Koehn et al., 2003) which consists of many small sub-components that are tuned separately, neural machine translation attempts to build and train a single, large neural network that reads a sentence and outputs a correct translation.

- Most of the proposed neural machine translation models belong to a family of encoder–decoders (Sutskever et al., 2014; Cho et al., 2014a), with an encoder and a decoder for each language, or involve a language-specific encoder applied to each sentence whose outputs are then compared (Hermann and Blunsom, 2014).

- An encoder neural network reads and encodes a source sentence into a fixed-length vector. A decoder then outputs a translation from the encoded vector. The whole encoder–decoder system, which consists of the encoder and the decoder for a language pair, is jointly trained to maximize the probability of a correct translation given a source sentence.

- A potential issue with this encoder–decoder approach is that a neural network needs to be able to compress all the necessary information of a source sentence into a fixed-length vector. This may make it difficult for the neural network to cope with long sentences, especially those that are longer than the sentences in the training corpus.

- The most important distinguishing feature of this approach from the basic encoder–decoder is that it does not attempt to encode a whole input sentence into a single fixed-length vector. Instead, it encodes the input sentence into a sequence of vectors and chooses a subset of these vectors adaptively while decoding the translation.

- The paper proposes a novel neural machine translation architecture that extends the standard encoder–decoder by introducing a soft attention mechanism. It extended the basic encoder–decoder by letting a model (soft-)search for a set of input words, or their annotations computed by an encoder, when generating each target word. This frees the model from having to encode a whole source sentence into a fixed-length vector, and also lets the model focus only on information relevant to the generation of the next target word. This has a major positive impact on the ability of the neural machine translation system to yield good results on longer sentences. Unlike with the traditional machine translation systems, all of the pieces of the translation system, including the alignment mechanism, are jointly trained towards a better log-probability of producing correct translations.

- This was done through the usage of attention mechanism (decoding phase) and bidirectional RNN (encoding phase).

- The encoder uses a bidirectional RNN to capture both past and future context in the source sentence, generating richer annotations (hidden states) for each input word.

- **Attention Mechanism:**

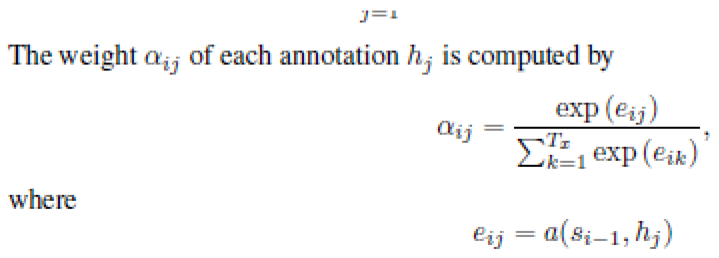
  They **extend the encoder–decoder model with a soft attention mechanism**, which allows the decoder to:

  1. **Look back ("attend") to all encoder hidden states** (annotations of source words).

  2. **Dynamically compute a context vector (c□)** for each target word, based on a weighted average of encoder states.

  3. **Use the current decoder hidden state (h□) to compute alignment scores** with each source position, creating a **soft alignment** vector a□.

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j.$$



$j=1$

The weight α_{ij} of each annotation h_j is computed by

$$\alpha_{ij} = \frac{\exp\left(e_{ij}\right)}{\sum_{k=1}^{T_x} \exp\left(e_{ik}\right)},$$

where

$$e_{ij} = a(s_{i-1}, h_j)$$

This process helps the model:

- **Focus only on relevant parts of the input** for generating each output word.

- **Avoid information bottlenecks** caused by a single fixed-length representation.

-