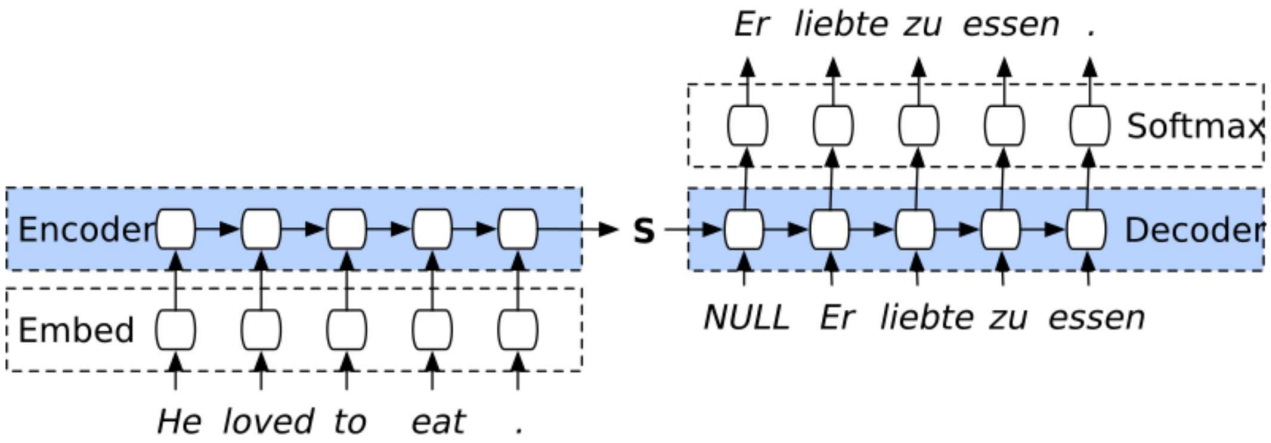
Attention Mechanisms and Neural Machine Translation

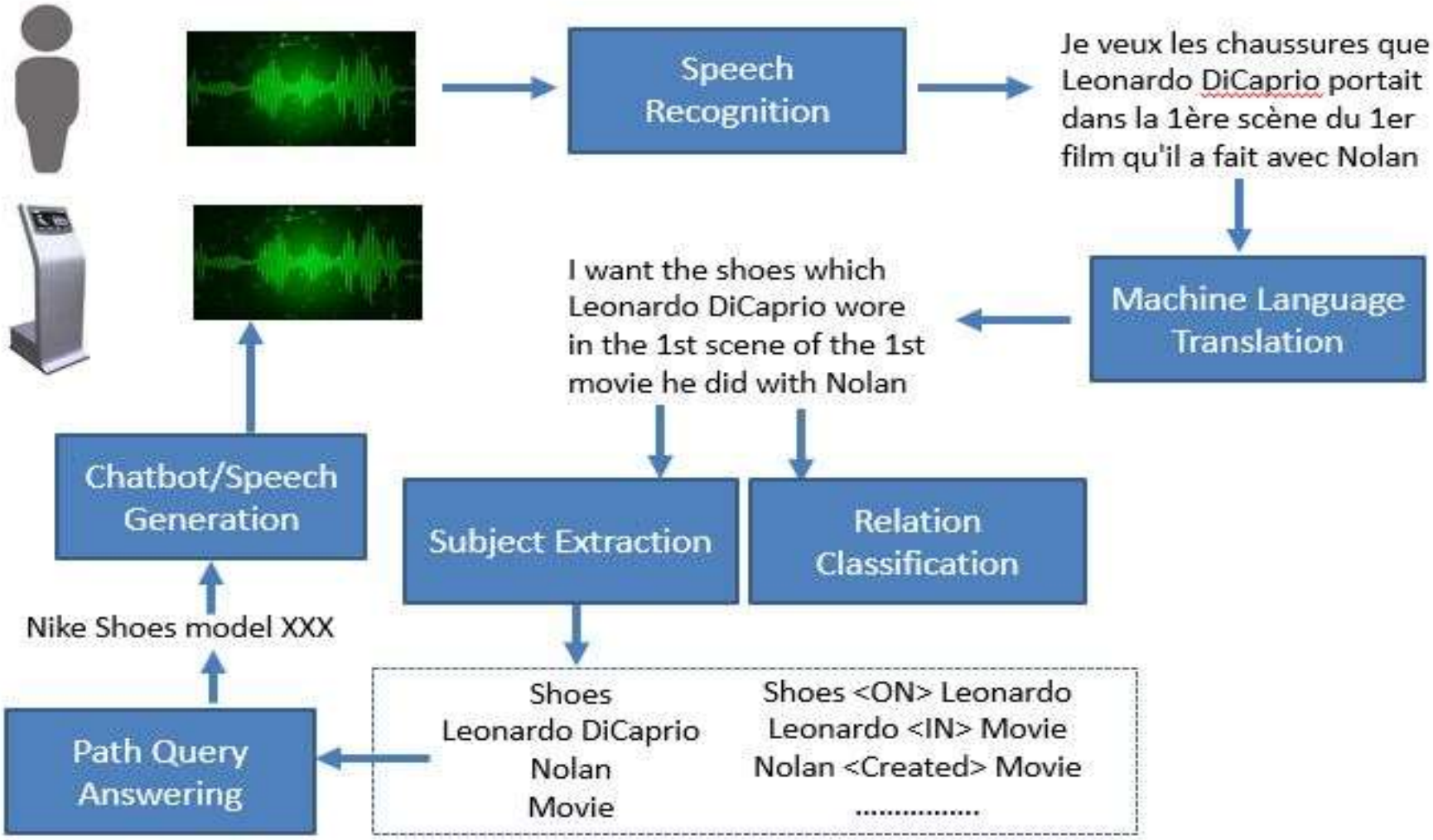
Harsha (Harsh) Gandikota

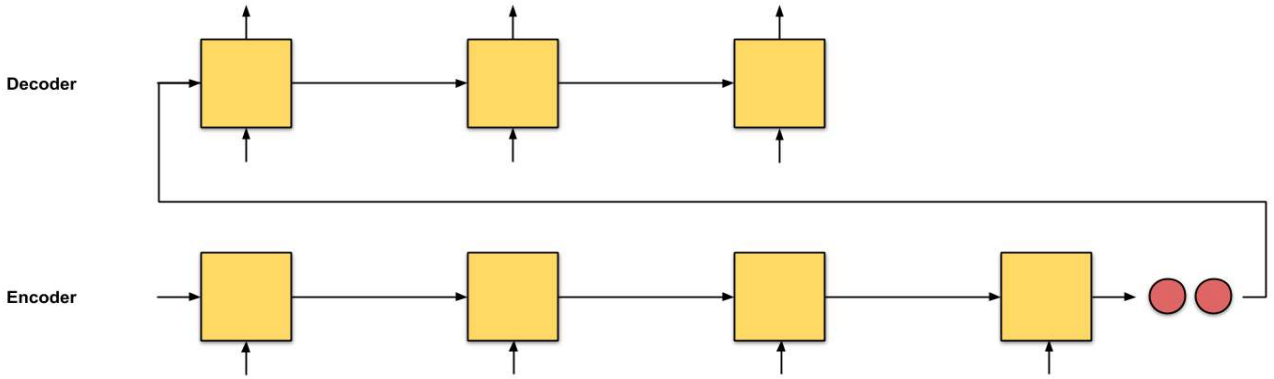
A run down of seq2seq models



Seq2Seq models constitute a common framework of converting the sequences of one domain space to another domain space.

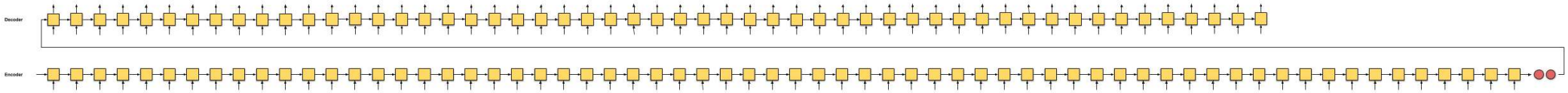
Seq2Seq converts input sequences to a hidden state that represents the entire input sequence in a fixed dimensionality which is used to derive the appropriate output sequence.

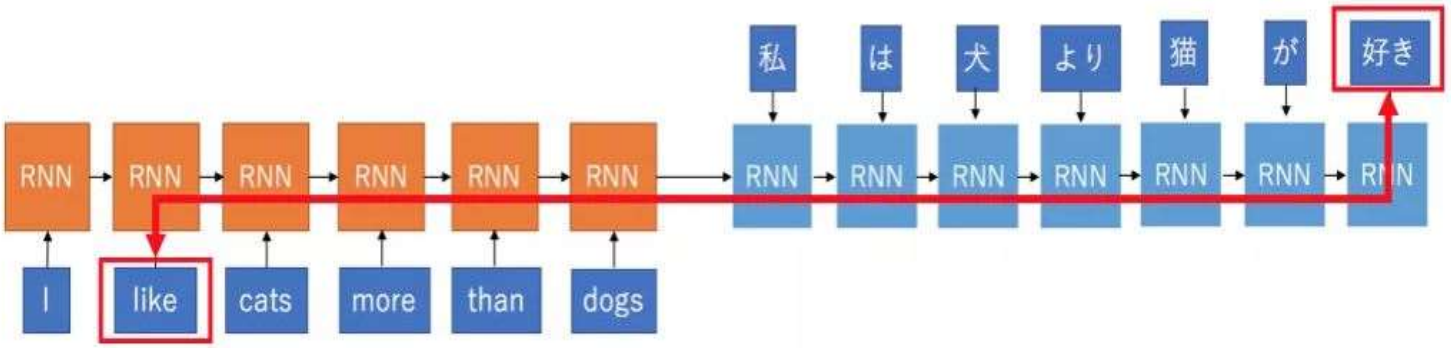

Speech Recognition
Je veux les chaussures que Leonardo DiCaprio portait dans la 1ère scène du 1er film qu'il a fait avec Nolan
Machine Language Translation
I want the shoes which Leonardo DiCaprio wore in the 1st scene of the 1st movie he did with Nolan
Subject Extraction
Relation Classification
Chatbot/Speech Generation
Nike Shoes model XXX
Path Query Answering
Shoes
Leonardo DiCaprio
Nolan
Movie
Shoes <ON> Leonardo
Leonardo <IN> Movie
Nolan <Created> Movie
.....



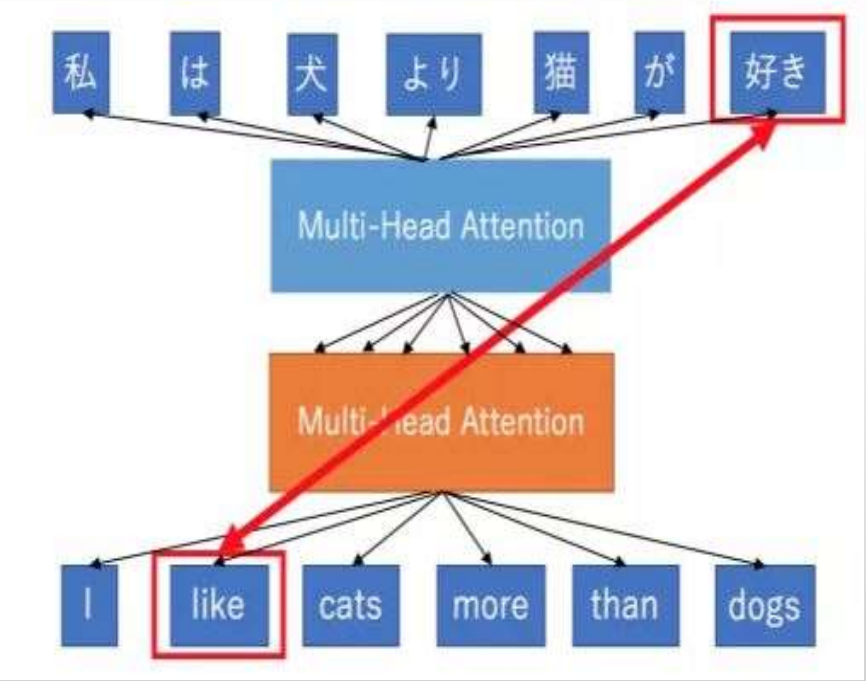
Dilemma of the seq2seq model

- Seq2Seq models tend to forget and fail to capture dependencies for longer sequences.
- Suffer from vanishing gradient and exploding gradient problem.
- Execute sequentially and cannot be parallelized.





How Attention mechanisms solve the Seq2seq dilemma



Seq2Seq models use only a single vector to represent the sequence where as Attention-based models use multiple vectors to represent the sequence

Attention provides a way for the decoder to consider information from every encoder hidden state.

Neural Machine Translation : Intuition seq2seq

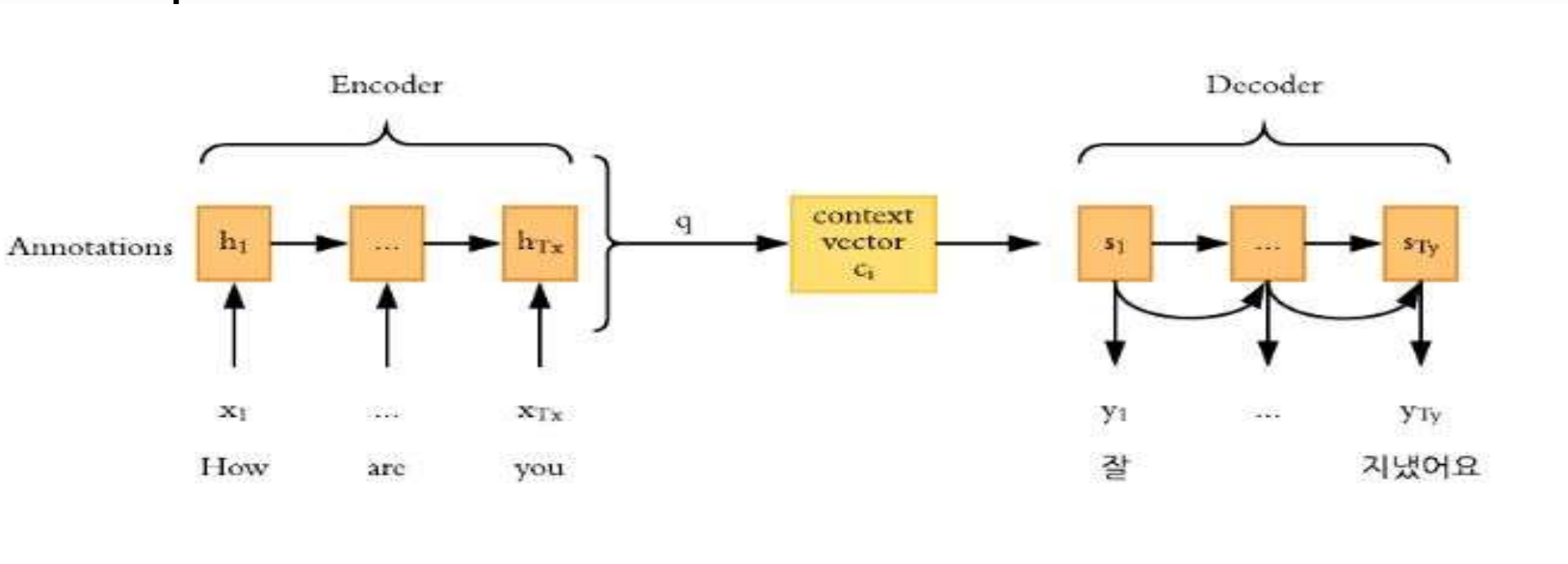
> **Intuition: seq2seq**
>
> A translator reads the German text from start till the end. Once done, he starts translating to English word by word. It is possible that if the sentence is extremely long, he might have forgotten what he has read in the earlier parts of the text.

Neural Machine Translation : Intuition
seq2seq + Attention

> **Intuition: seq2seq + attention**
>
> A translator reads the German text *while writing down the keywords* from the start till the end, after which he starts translating to English. While translating each German word, he makes use of the keywords he has written down.

Neural Machine Translation : Previous Model Architecture

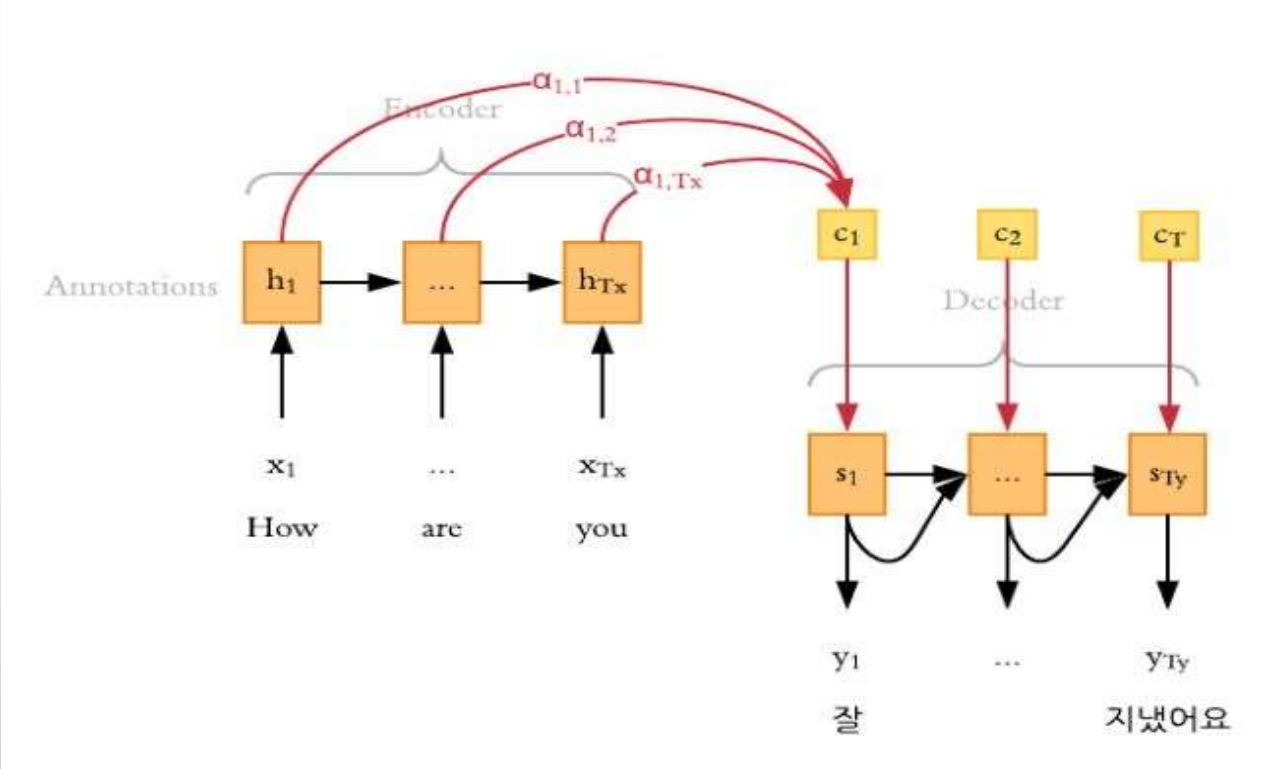
- Translation models used to compress the entire input sequence into a fixed length vector, which proved ineffective for large input sequences.



Neural Machine Translation : Proposed Model Architecture

- When generating a new word, the decoder would ‘search’ the input sequence that is the most relevant to the current word being generated.

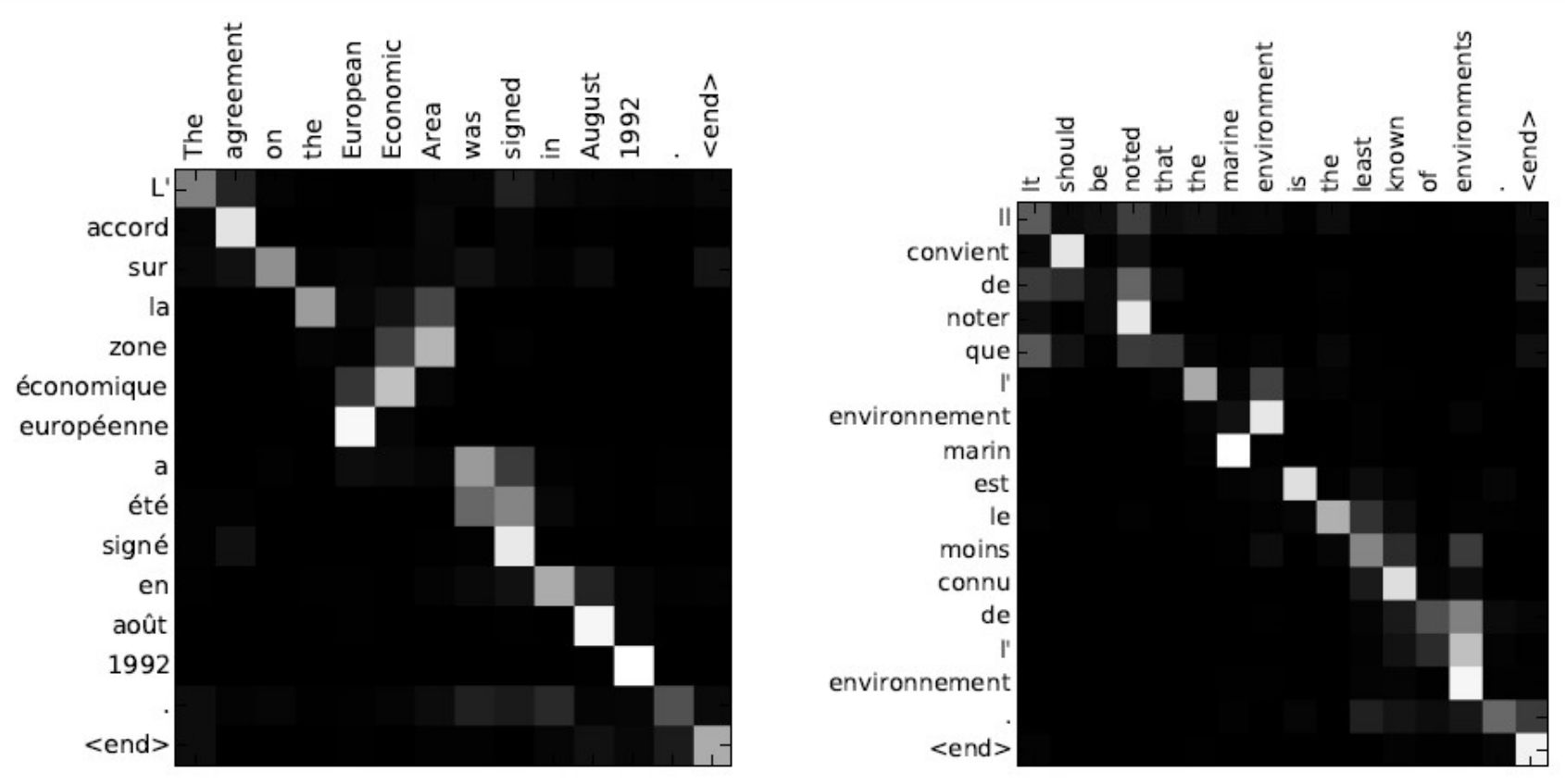
This relieves the model from compressing the entire sequence into a single vector.



Neural Machine Translation : Alignment and then Translation

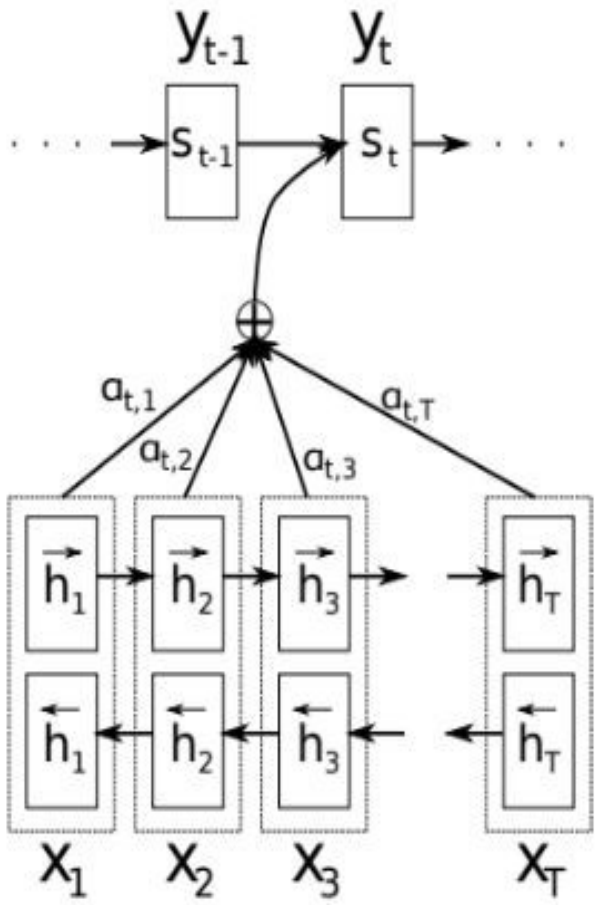
- Alignment in Neural Machine Translation:

  How well inputs around position j match with outputs at position i.



Neural Machine Translation: Context Vectors

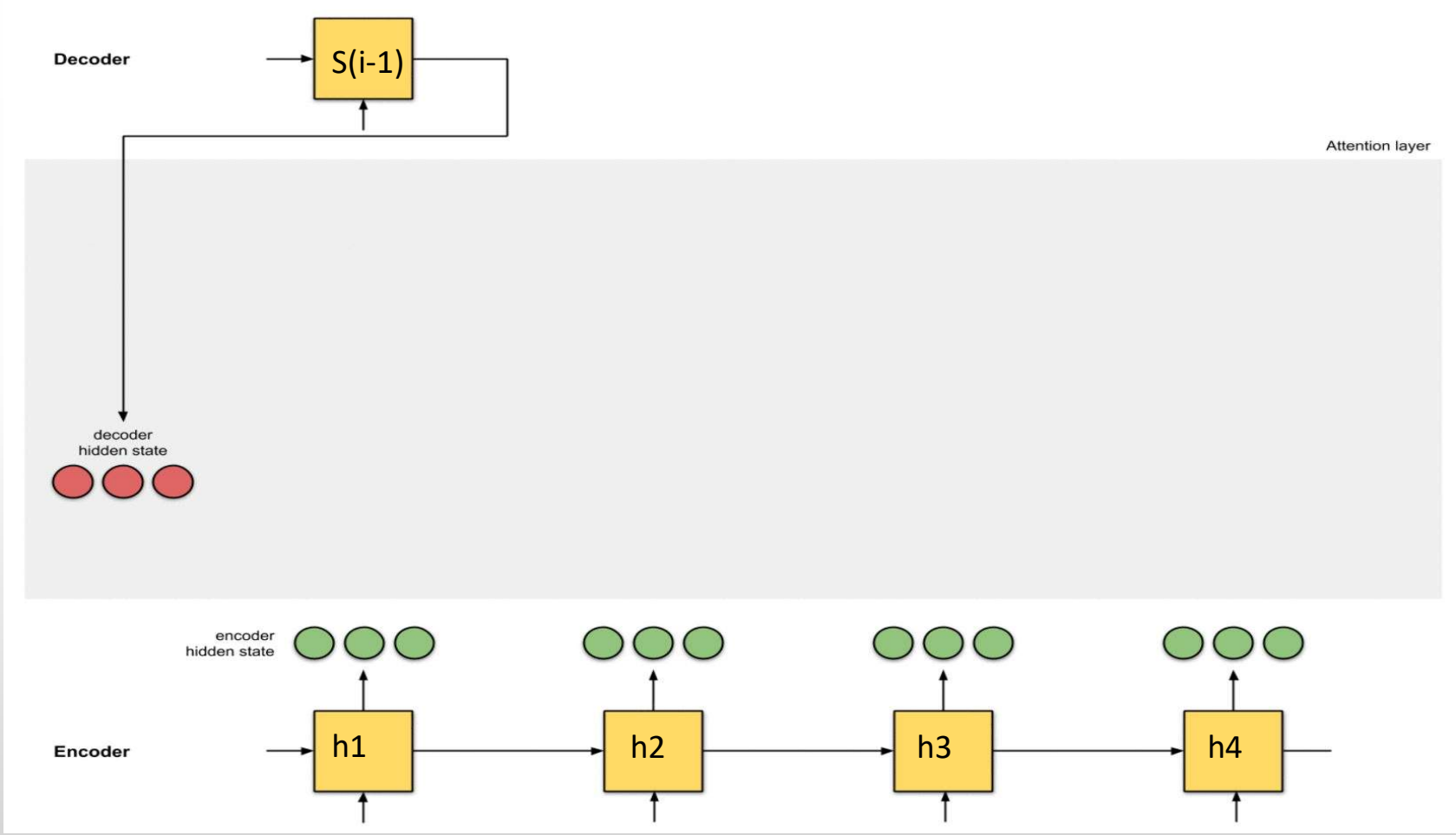
When generating a new word, the decoder considers a ‘context vector’ to decide what position to pay attention to.



context vector = Summation of Alignment Vectors times each Hidden State

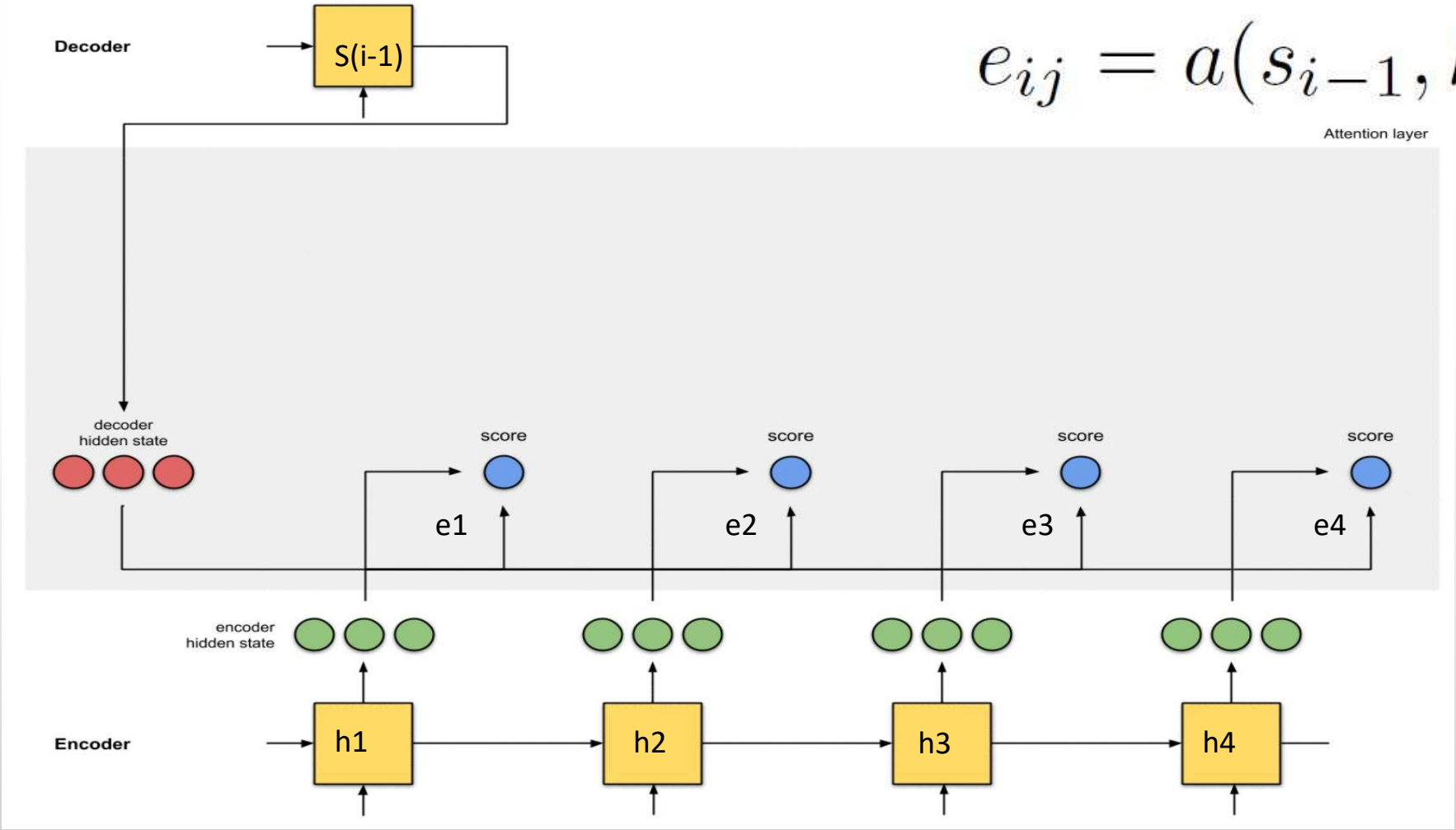
$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$$

Step 0: Prepare Hidden States



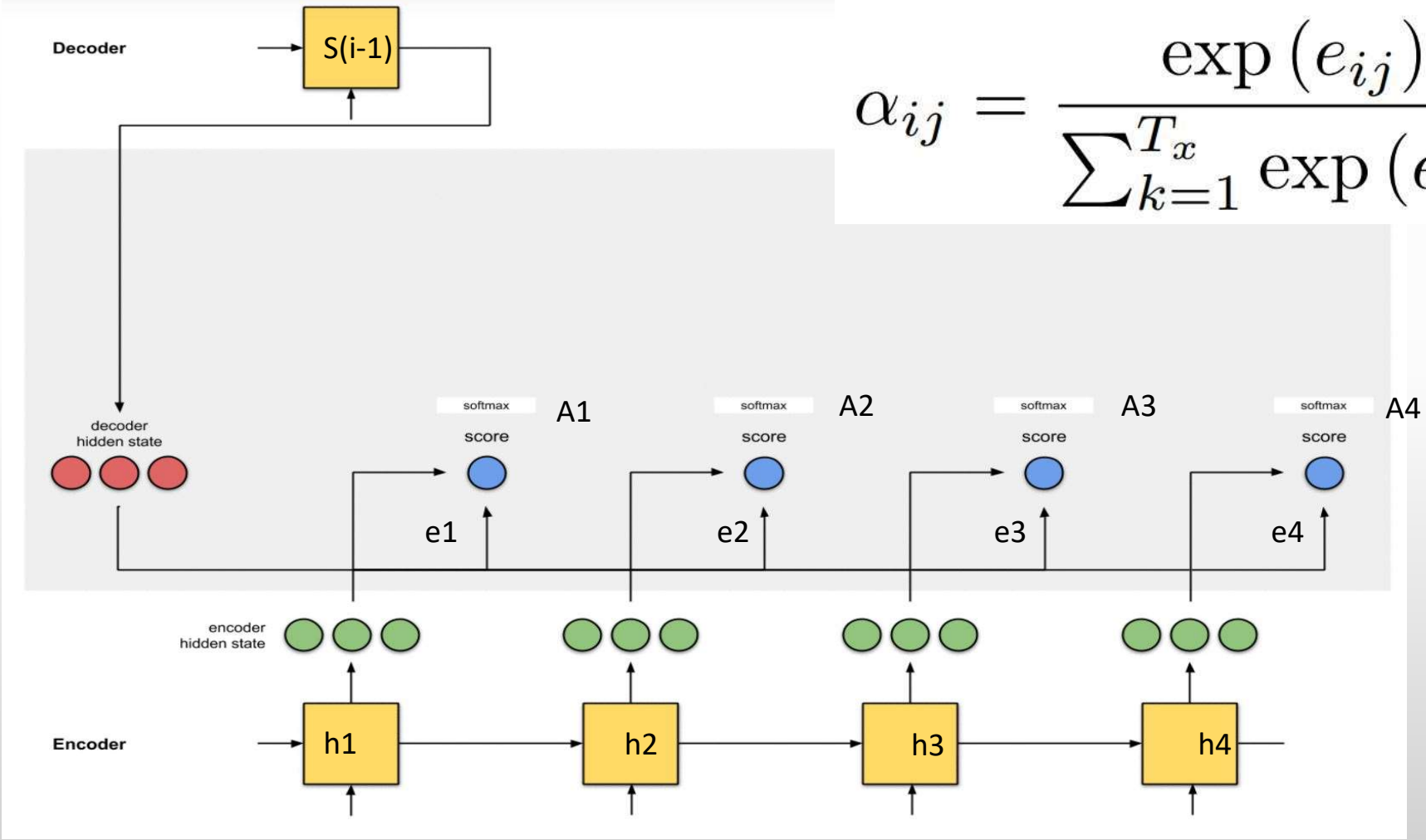
Step 1: Obtain a Score for every hidden state

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

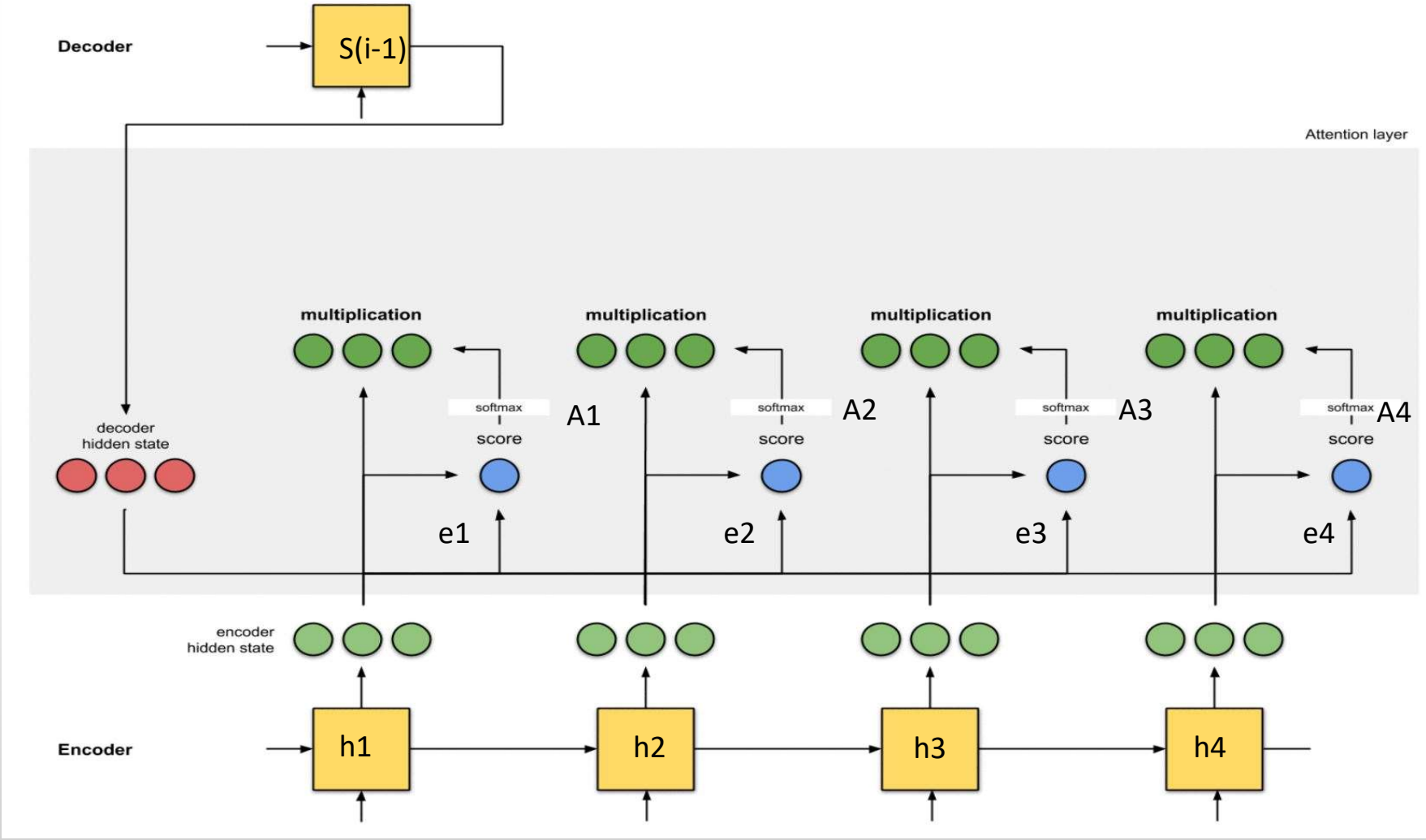


Step 2: Run Scores through Softmax Layer

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

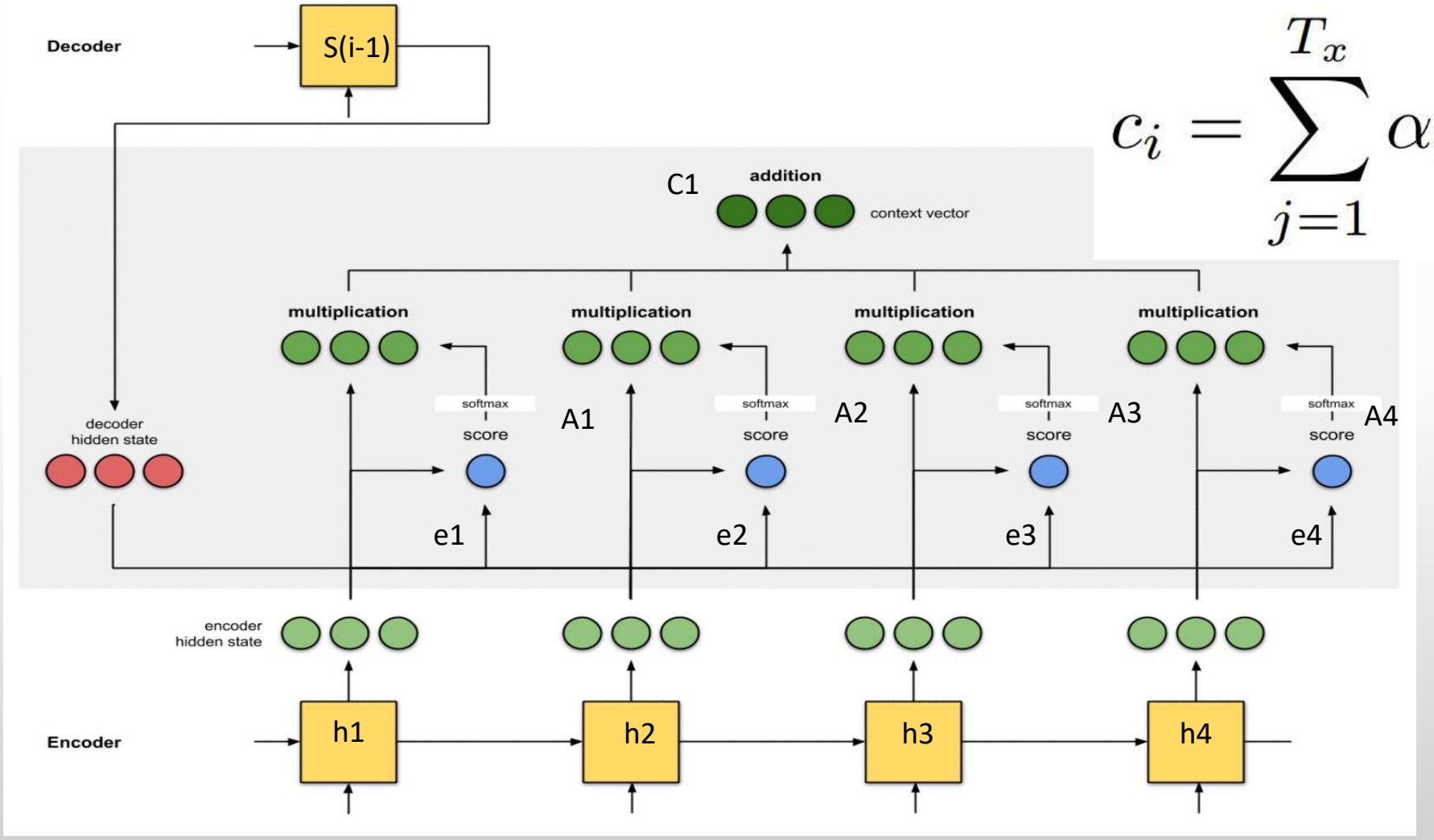


Step 3: Multiply each Encoder state by the softmaxed score

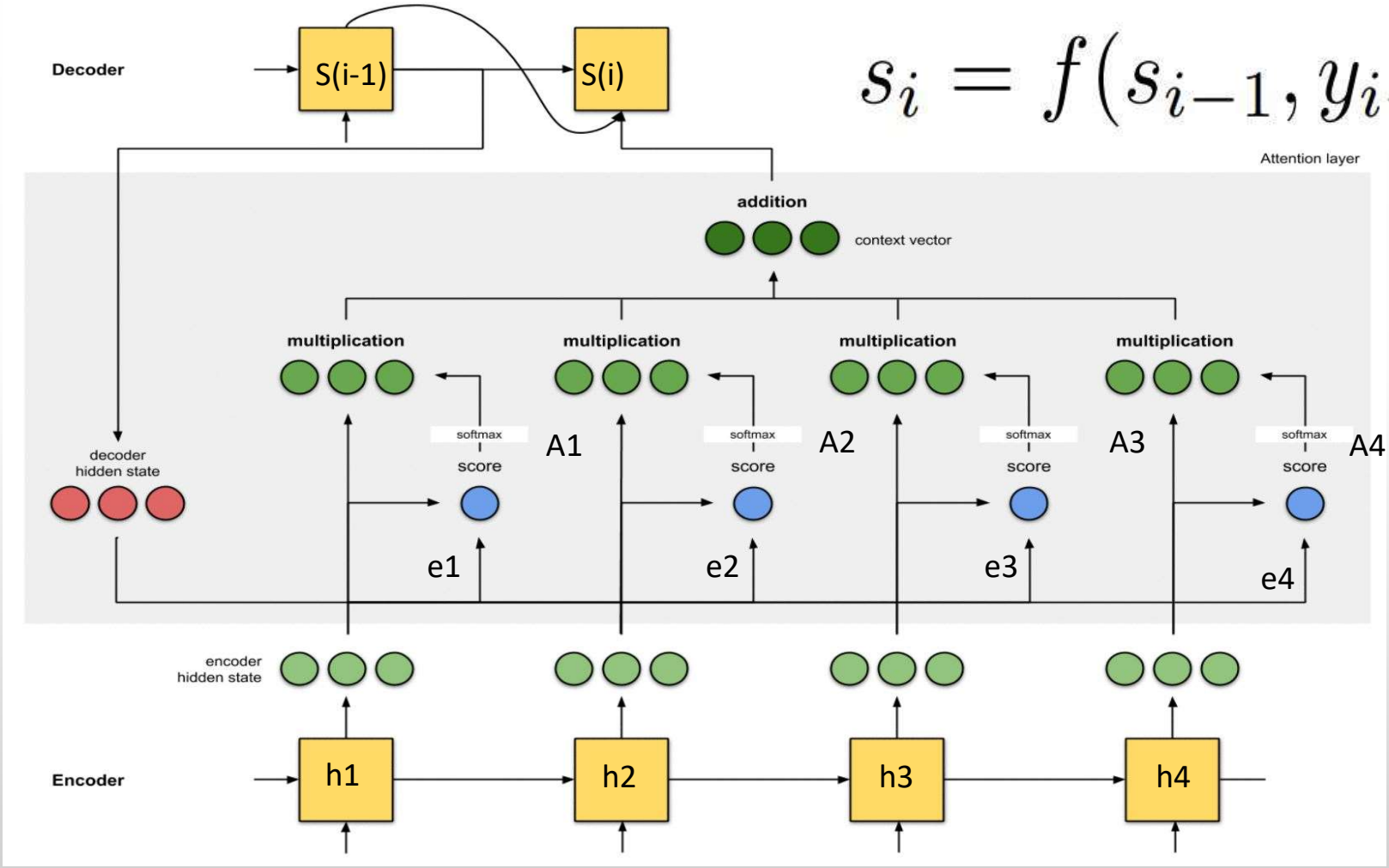


Step 4: Sum up Alignment Vectors

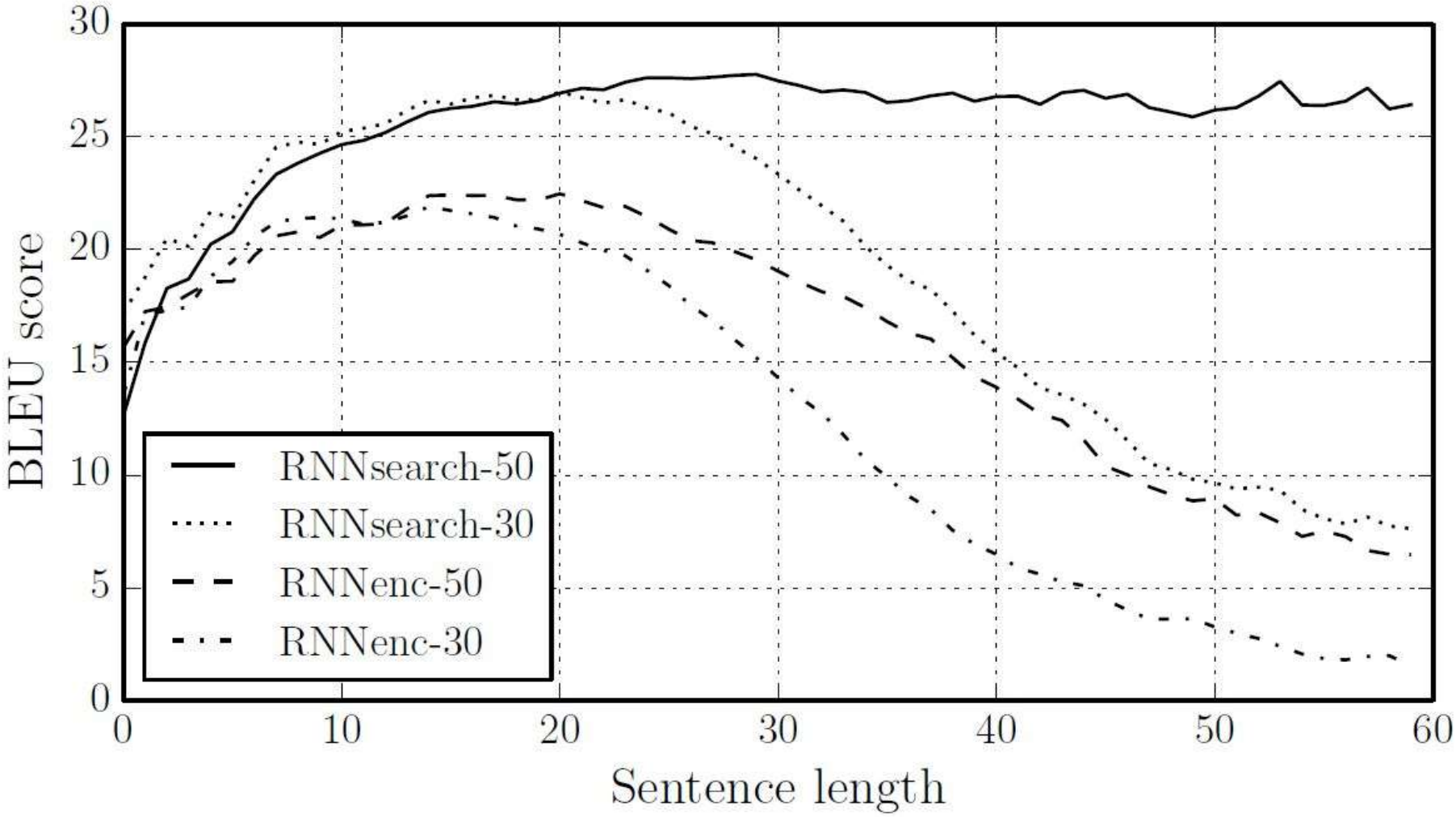
$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$$



Step 5: Feed Context Vector into Decoder



Attention vs non-attention



References:

Research Papers:

- **Neural Machine Translation by Jointly Learning to Align and Translate** https://arxiv.org/abs/1409.0473
- **Attention Is All You Need** https://arxiv.org/abs/1706.03762

Online Links

- **Sequence to sequence model: Introduction and concepts** https://towardsdatascience.com/sequence-to-sequence-model-introduction-and-concepts-44d9b41cd42d
- **The Illustrated Transformer** http://jalammar.github.io/illustrated-transformer/
- **seq2seq (Sequence to Sequence) Model for Deep Learning with PyTorch**
- https://www.guru99.com/seq2seq-model.html
- **Attention? Attention!** https://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html