CS5242 : Neural Networks and Deep Learning

Lecture 8b: Recurrent Neural Networks Applications

Semester 1 2022/23

Ai Xin

aixin@comp.nus.edu.sg

Department of Computer Science
National University of Singapore (NUS)

Slides Credit: Xavier Bresson

Outline

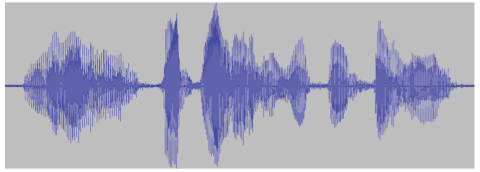
- Speech recognition
  - Introduction
  - Spectrogram
  - Inference
- Machine Translation
  - Introduction
  - Architecture
  - Inference
  - Training

Outline

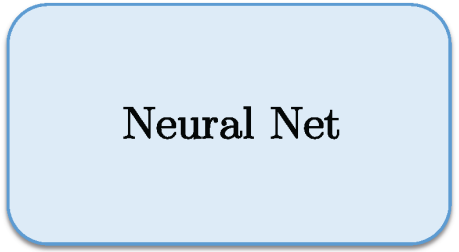
- Speech recognition
  - Introduction
  - Spectrogram
  - Inference
- Machine Translation
  - Introduction
  - Architecture
  - Inference
  - Training

Speech recognition

- Speech-to-text is a fundamental task :
  - Virtual assistants : Apple Siri, Amazon Alexa, Microsoft Cortana, Google Assistant require to change the speech signal into a sequence of words.
  - Beyond keyboard : Speech recognition in-place of keyboard (when error is less than 1%).
  - Current performances: 5.1% error by Microsoft Research, as good as human transcript.



Input :
Speech signal



"call a taxi"

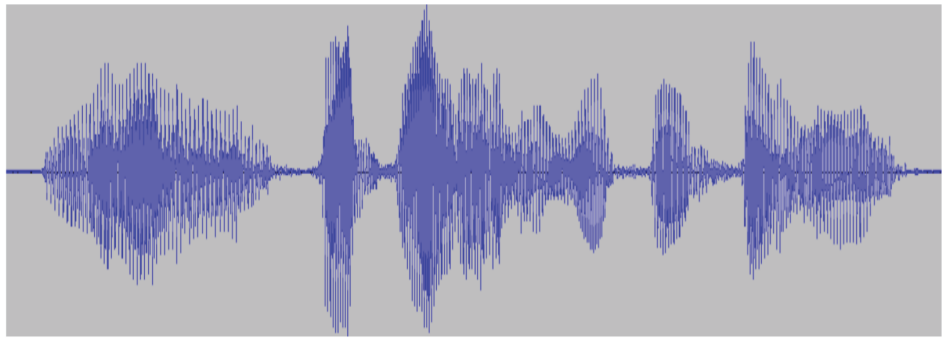
Output:
Sequence of words

Outline

- Speech recognition
  - Introduction
  - Spectrogram
  - Inference
- Machine Translation
  - Introduction
  - Architecture
  - Inference
  - Training

Spectrogram

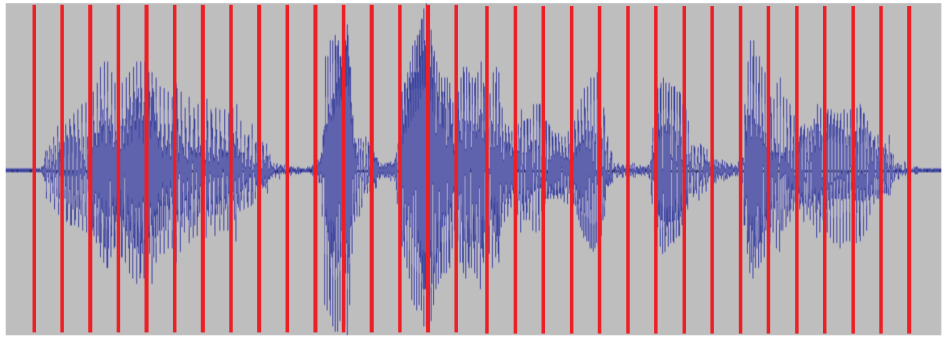
- We pre-process the raw audio signal to extract temporal and frequencial information.



Raw audio signal

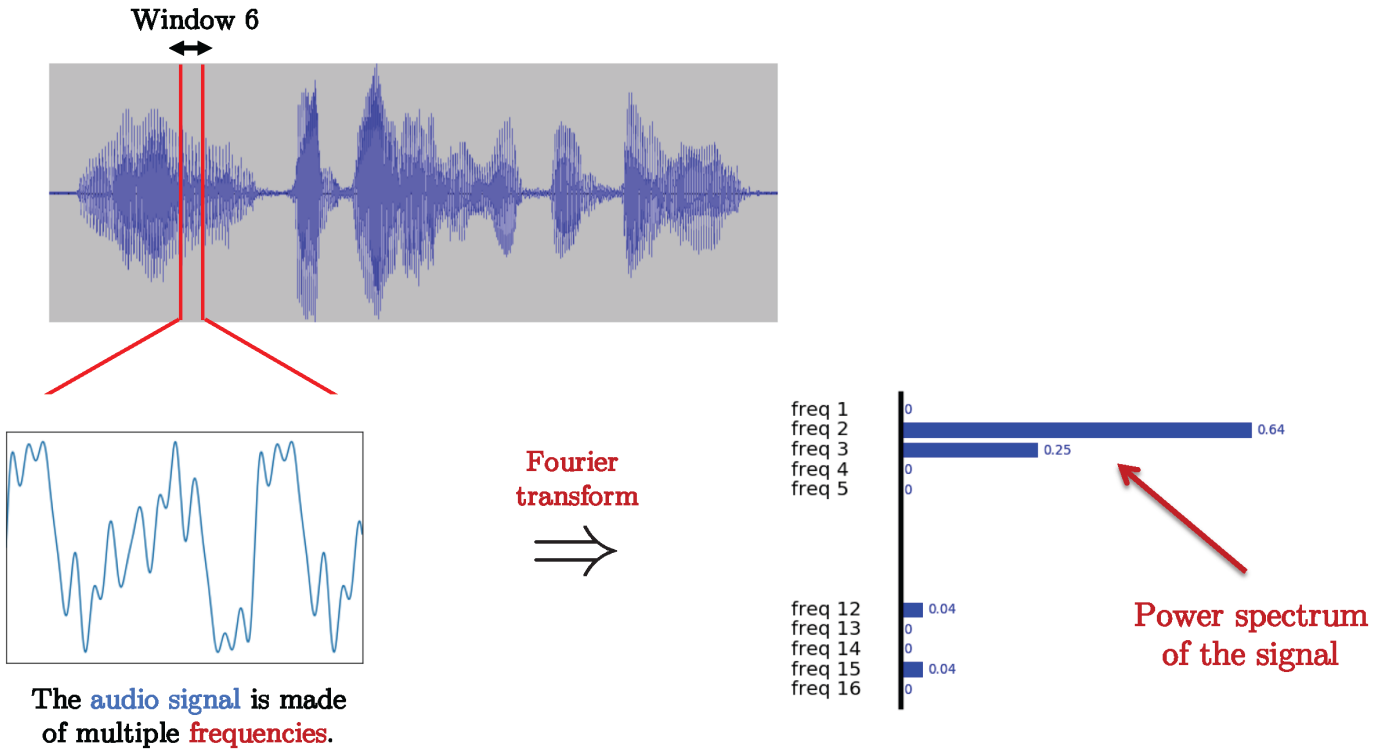
Spectrogram

- First, we cut the raw audio signal into small windows of 20 ms :



Spectrogram

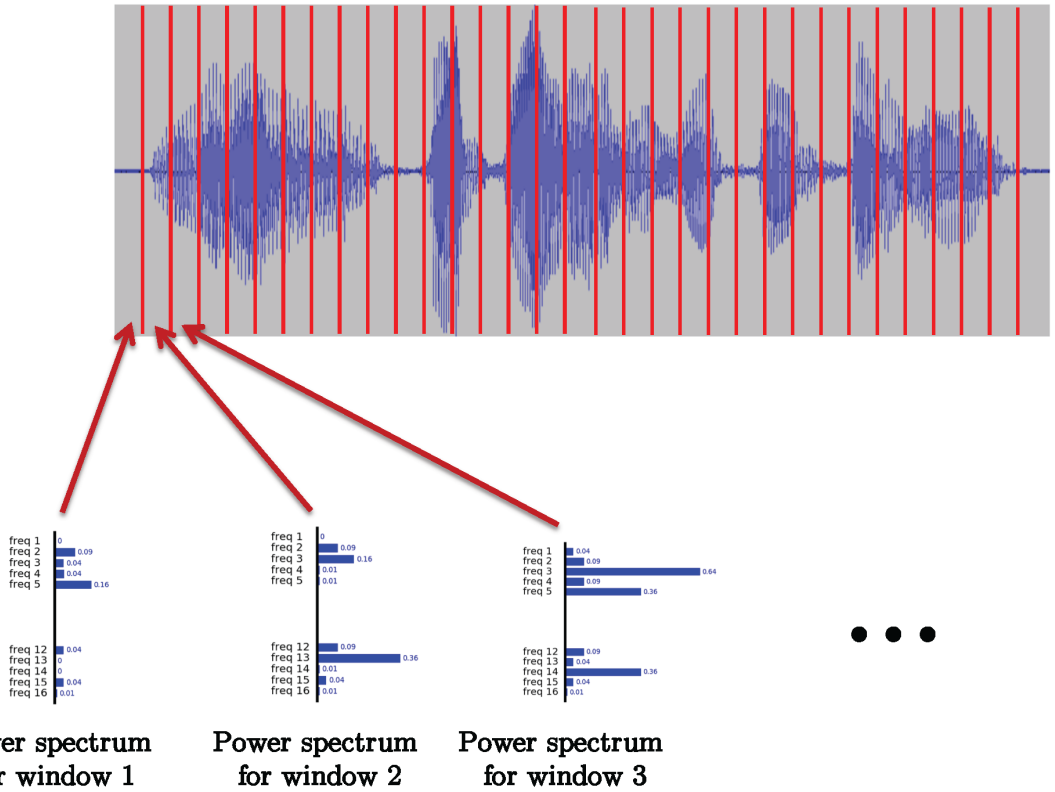
- We compute the power spectrum of the signal (Fourier transform) for each window :



The audio signal is made of multiple frequencies.

Spectrogram

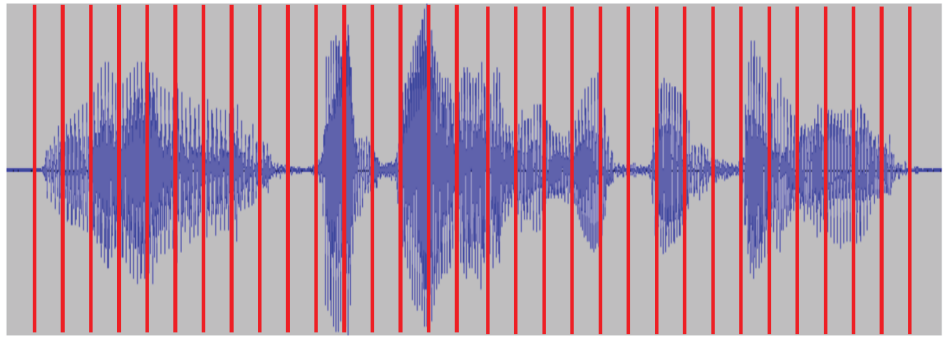
- We compute the power spectrum for all 20ms windows :

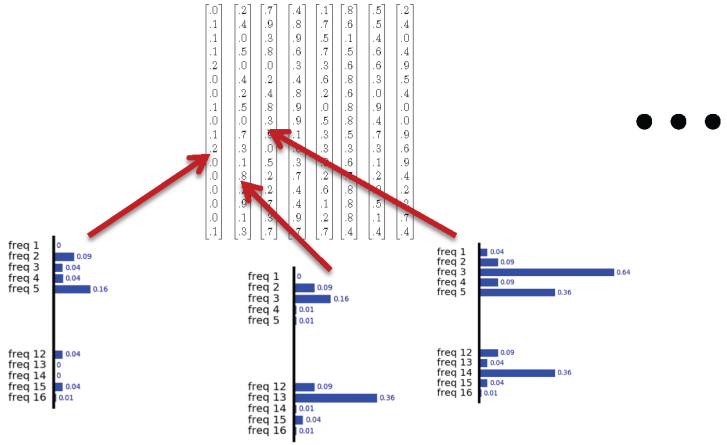


Power spectrum for window 1 | Power spectrum for window 2 | Power spectrum for window 3

Spectrogram

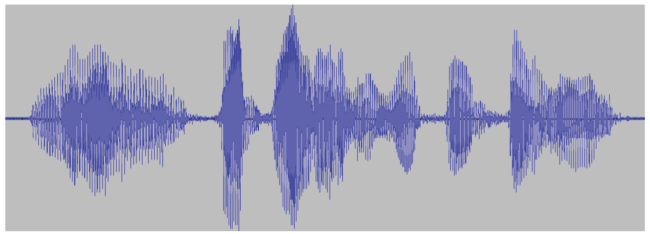
- We compute the power spectrum of all 20ms windows :



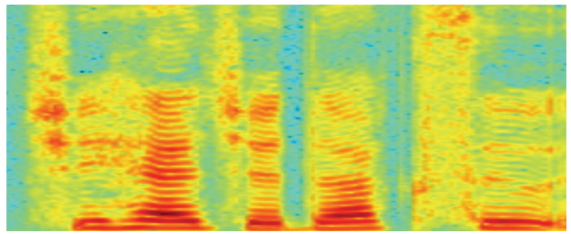


Spectrogram

- The raw audio signal has been converted into a sequence of vectors (size is between 100-200).
- This sequence of vectors is called the spectrogram.
- It can be computed very quickly with FFT.



Raw audio signal



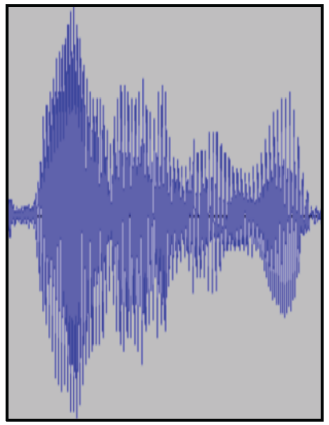
Spectogram

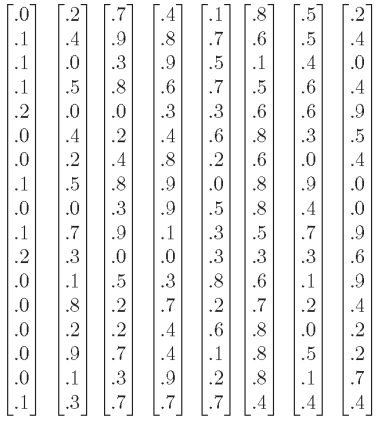
Outline

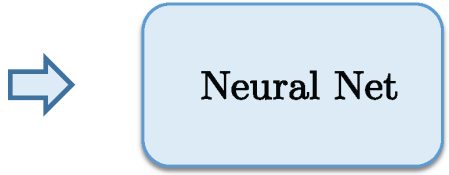
- Speech recognition
  - Introduction
  - Spectrogram
  - Inference
- Machine Translation
  - Introduction
  - Architecture
  - Inference
  - Training

Inference

- Inference process :
  - Feed the sequence of vectors (spectogram) to a trained RNN :







"Hello"

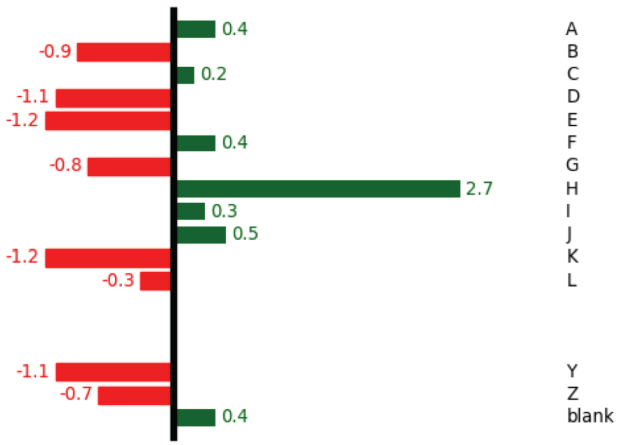
Input :
Audio signal

Spectogram

Output :
Sequence of words

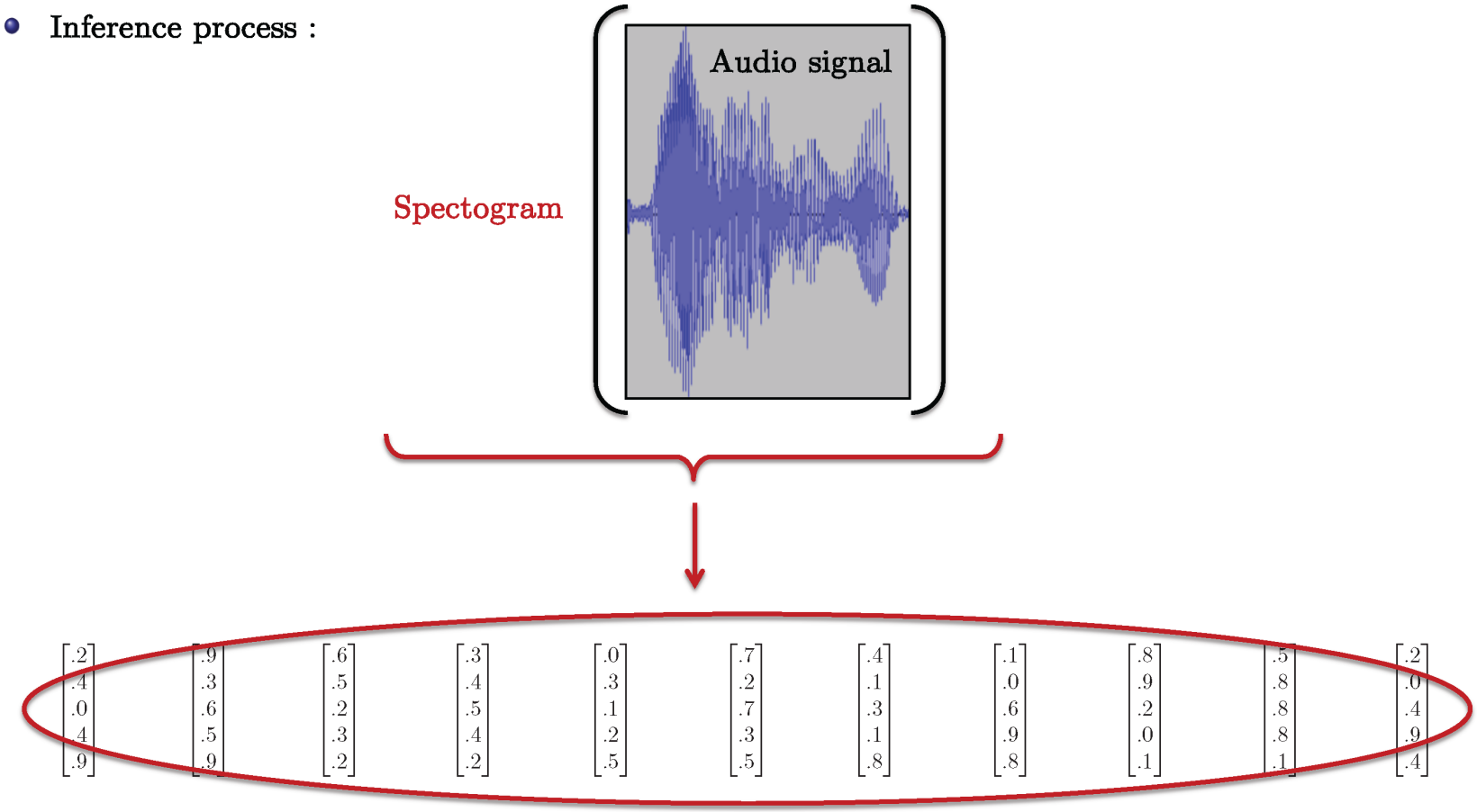
Inference

- Dictionary/vocabulary is defined as follows :
  - 27 classes :
    - The 26 letters of the alphabet
    - Blank symbol
    - Space Symbol



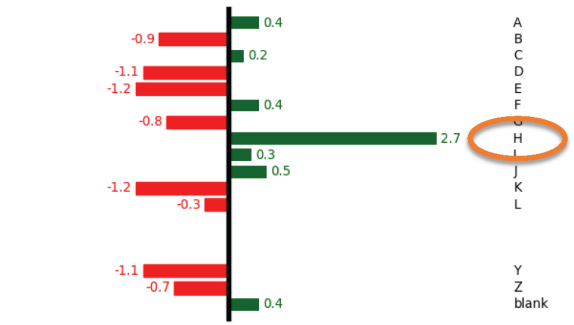
Inference

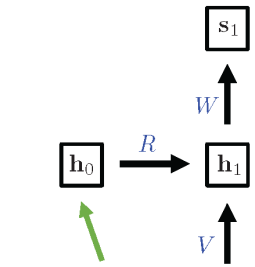
- Inference process :



Inference

- Inference process : Softmax the letter corresponding to the input spectrogram vector.



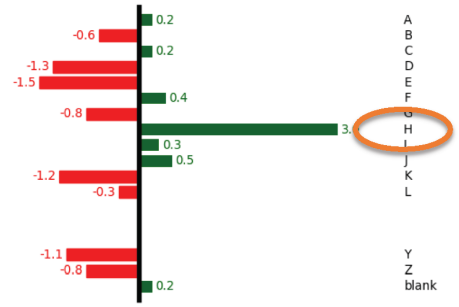


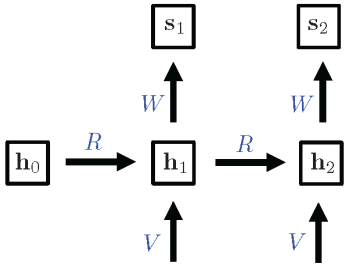
Hidden states initialized to zero vectors.

$$\begin{bmatrix}.2\\.4\\.0\\.4\\.9\end{bmatrix}\quad\begin{bmatrix}.9\\.3\\.6\\.5\\.9\end{bmatrix}\quad\begin{bmatrix}.6\\.5\\.2\\.3\\.2\end{bmatrix}\quad\begin{bmatrix}.3\\.4\\.5\\.4\\.2\end{bmatrix}\quad\begin{bmatrix}.0\\.3\\.1\\.2\\.5\end{bmatrix}\quad\begin{bmatrix}.7\\.2\\.7\\.3\\.5\end{bmatrix}\quad\begin{bmatrix}.4\\.1\\.3\\.1\\.8\end{bmatrix}\quad\begin{bmatrix}.1\\.0\\.6\\.9\\.8\end{bmatrix}\quad\begin{bmatrix}.8\\.9\\.2\\.0\\.1\end{bmatrix}\quad\begin{bmatrix}.5\\.8\\.8\\.8\\.1\end{bmatrix}\quad\begin{bmatrix}.2\\.0\\.4\\.9\\.4\end{bmatrix}$$

Inference

- Inference process : Softmax the letter corresponding to the input spectrogram vector.

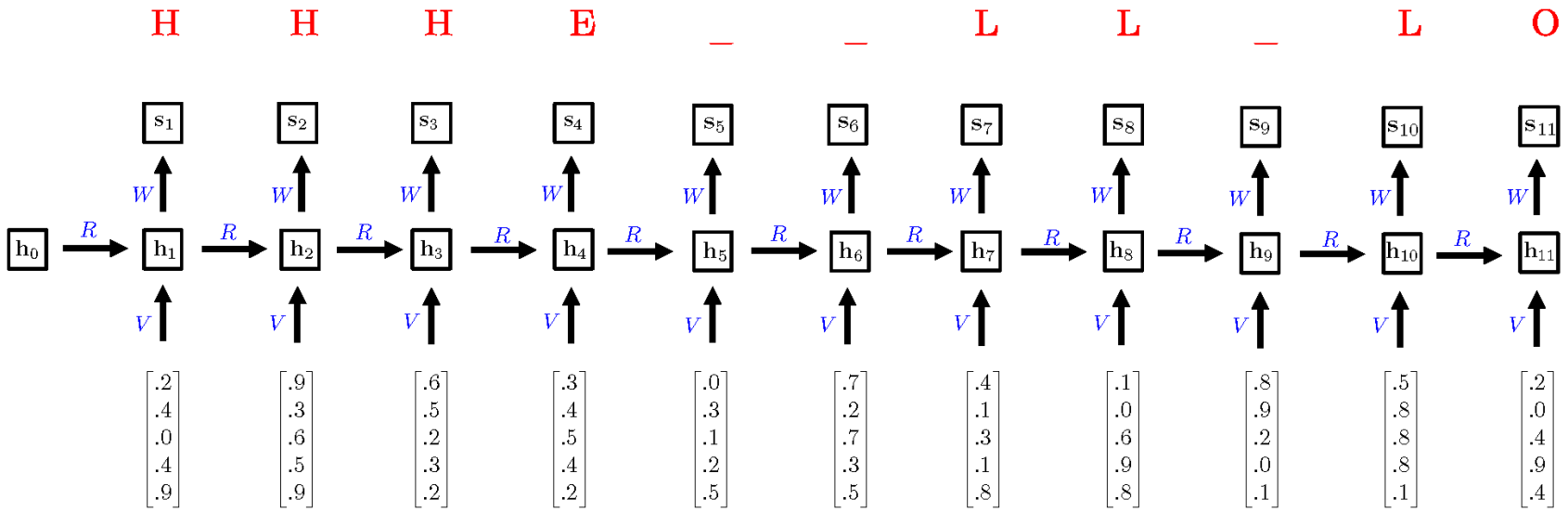




$$\begin{bmatrix}.2\\.4\\.0\\.4\\.9\end{bmatrix}\begin{bmatrix}.9\\.3\\.6\\.5\\.9\end{bmatrix}\begin{bmatrix}.6\\.5\\.2\\.3\\.2\end{bmatrix}\begin{bmatrix}.3\\.4\\.5\\.4\\.2\end{bmatrix}\begin{bmatrix}.0\\.3\\.1\\.2\\.5\end{bmatrix}\begin{bmatrix}.7\\.2\\.7\\.3\\.5\end{bmatrix}\begin{bmatrix}.4\\.1\\.3\\.1\\.8\end{bmatrix}\begin{bmatrix}.1\\.0\\.6\\.9\\.8\end{bmatrix}\begin{bmatrix}.8\\.9\\.2\\.0\\.1\end{bmatrix}\begin{bmatrix}.5\\.8\\.8\\.8\\.1\end{bmatrix}\begin{bmatrix}.2\\.0\\.4\\.9\\.4\end{bmatrix}$$

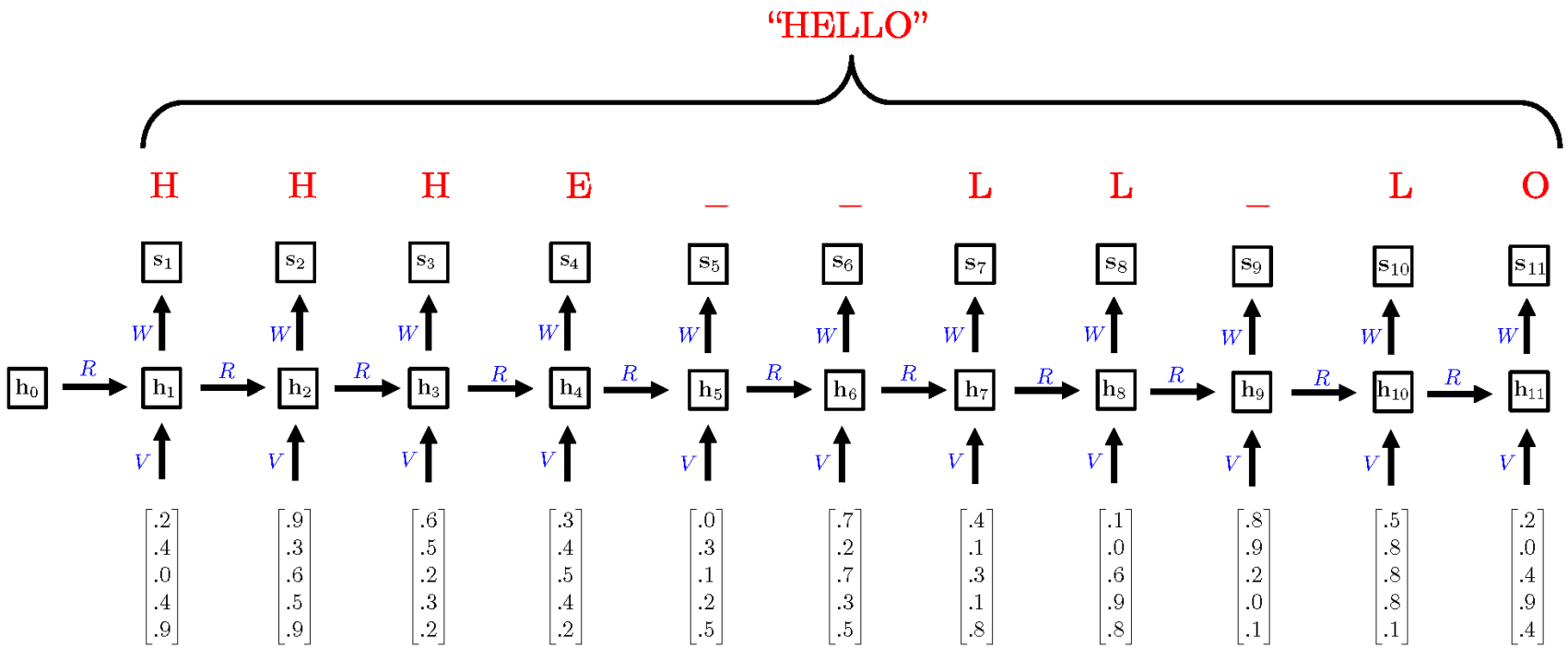
Inference

- Inference process :
  - Forward is done.



Inference

- Inference process :
  - The last layer at the very top is human hand-crafted :
    - Connectionist Temporal Classification (CTC)



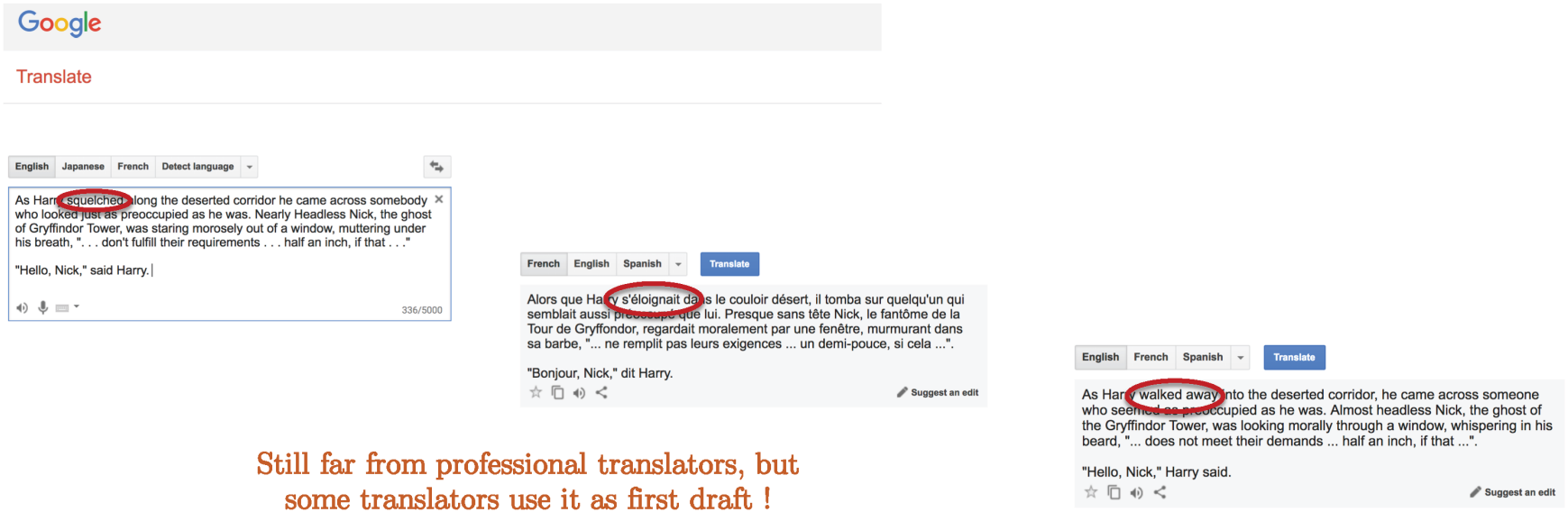
Outline

- Speech recognition
  - Introduction
  - Spectrogram
  - Inference
- **Machine Translation**
  - **Introduction**
  - Architecture
  - Inference
  - Training

Neural machine translation

- Google Translate switched to Neural Machine Translation in Nov. 2016.
  - This led to a big improvement in quality !

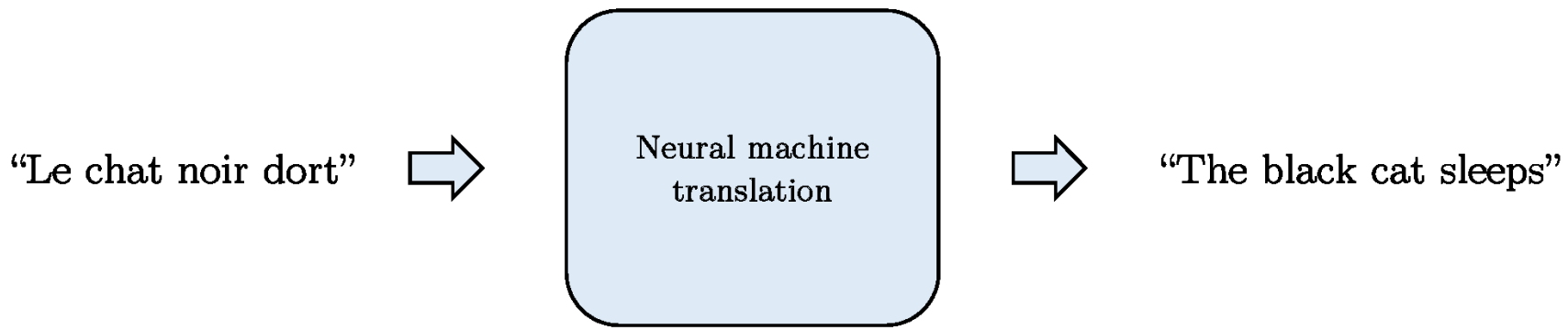
- Harry Potter and The Chamber of Secrets : English ⇒ French ⇒ English



Still far from professional translators, but
some translators use it as first draft !

Neural machine translation

- The challenges of translation :
  - The words can appear in different order: Alignment problem.
  - Input and output sentences do not necessarily have the same number of words: Length problem.
  - Input and output sentences do not have the same number of words in their dictionary: Vocabulary problem.



Outline

- Speech recognition
  - Introduction
  - Spectrogram
  - Inference
- **Machine Translation**
  - Introduction
  - **Architecture**
  - Inference
  - Training

Architecture

- A translation system is composed of two recurrent neural networks :
  - Encoder network :
    - The encoder network will summarize the input sequence in language A with a vector.
  - Decoder network :
    - The decoder network will take the encoding vector representing the input sequence in language A and will decode it to an output sequence in language B.
- Full end-to-end machine learning systems :
  - No conceptual prior on languages – opposite to Chomsky's paradigm.
  - Big change in the NLP community !

Architecture

Step 1: Take a sentence in French and encode it into a vector h of size 200.

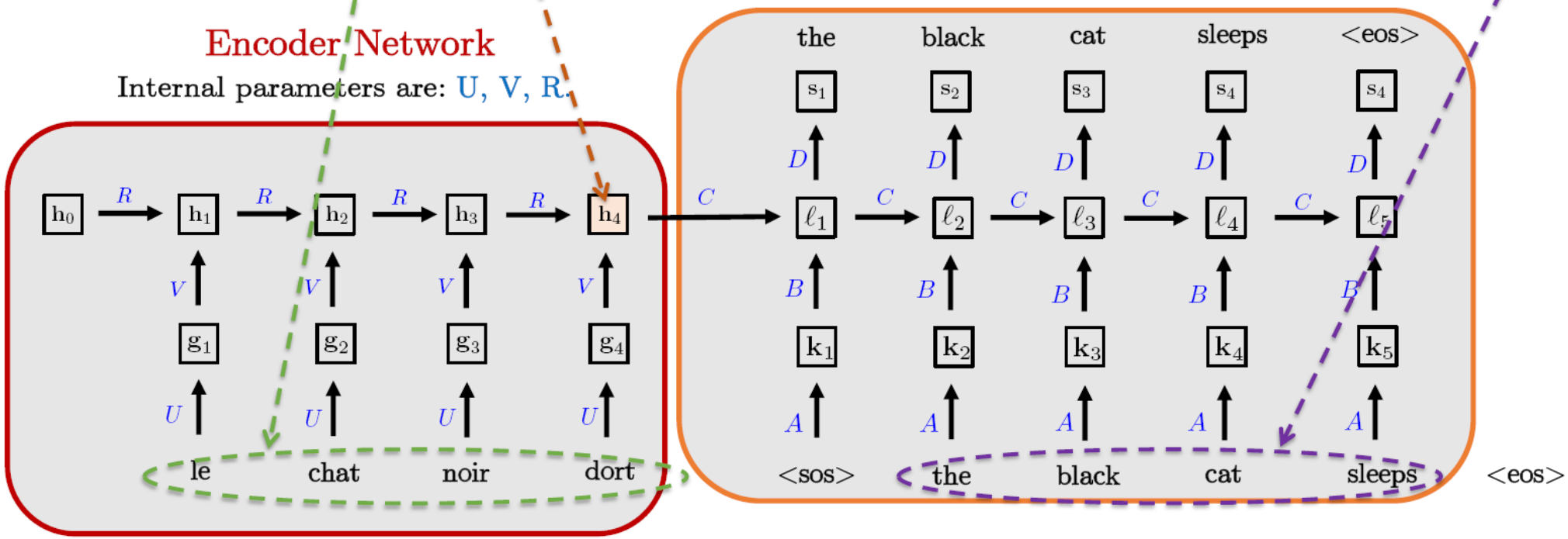
Step 2: Take the vector created by the encoder and use it to generate a sentence in English.

Encoder Network

Internal parameters are: U, V, R.

Decoder Network

Internal parameters are: A, B, C, D.



Architecture

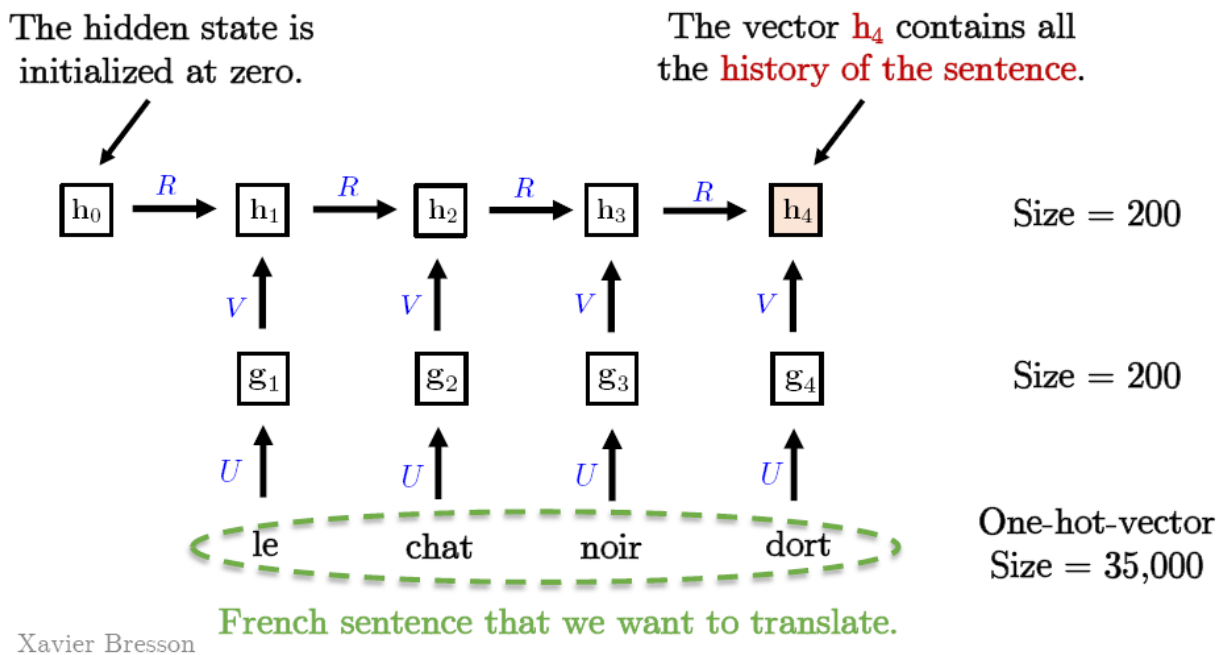
- Dictionaries : We have two vocabularies :
  - French vocabulary has 35,000 words.
    - Each French word is a one-hot-vector with 35,000 entries.
  - English vocabulary has 32,000 words.
    - Each English word is a one-hot-vector with 32,000 entries.

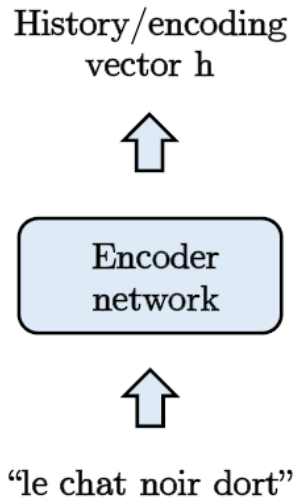
Outline

- Speech recognition
  - Introduction
  - Spectrogram
  - Inference
- **Machine Translation**
  - Introduction
  - Architecture
  - **Inference**
  - Training

Inference

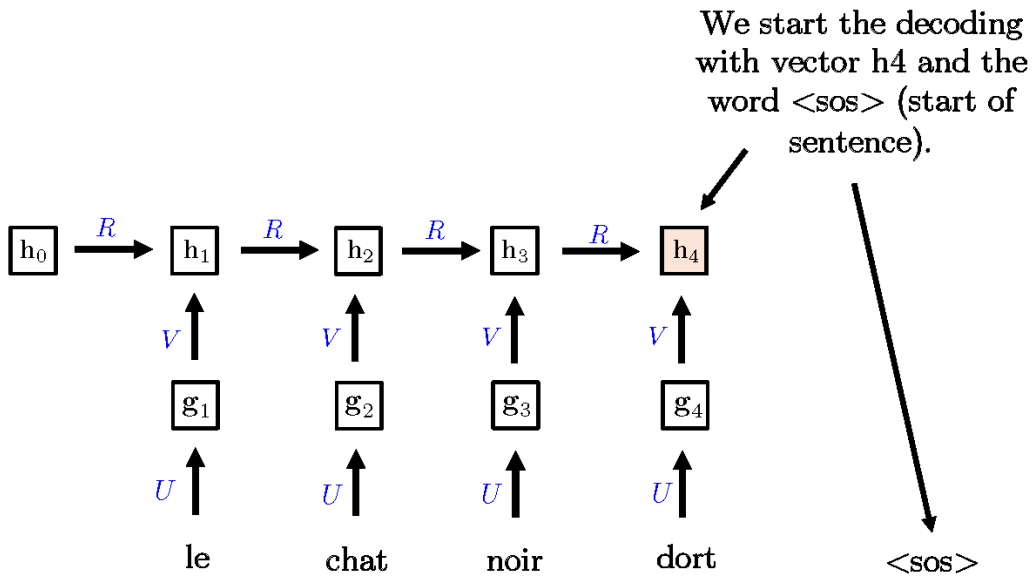
- The network has been trained, let us use it for inference.
- First step of inference:
  - Encode the input sequence with an history vector.

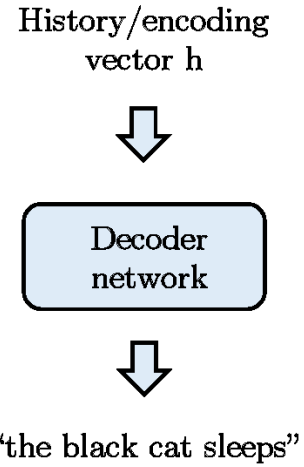




Inference

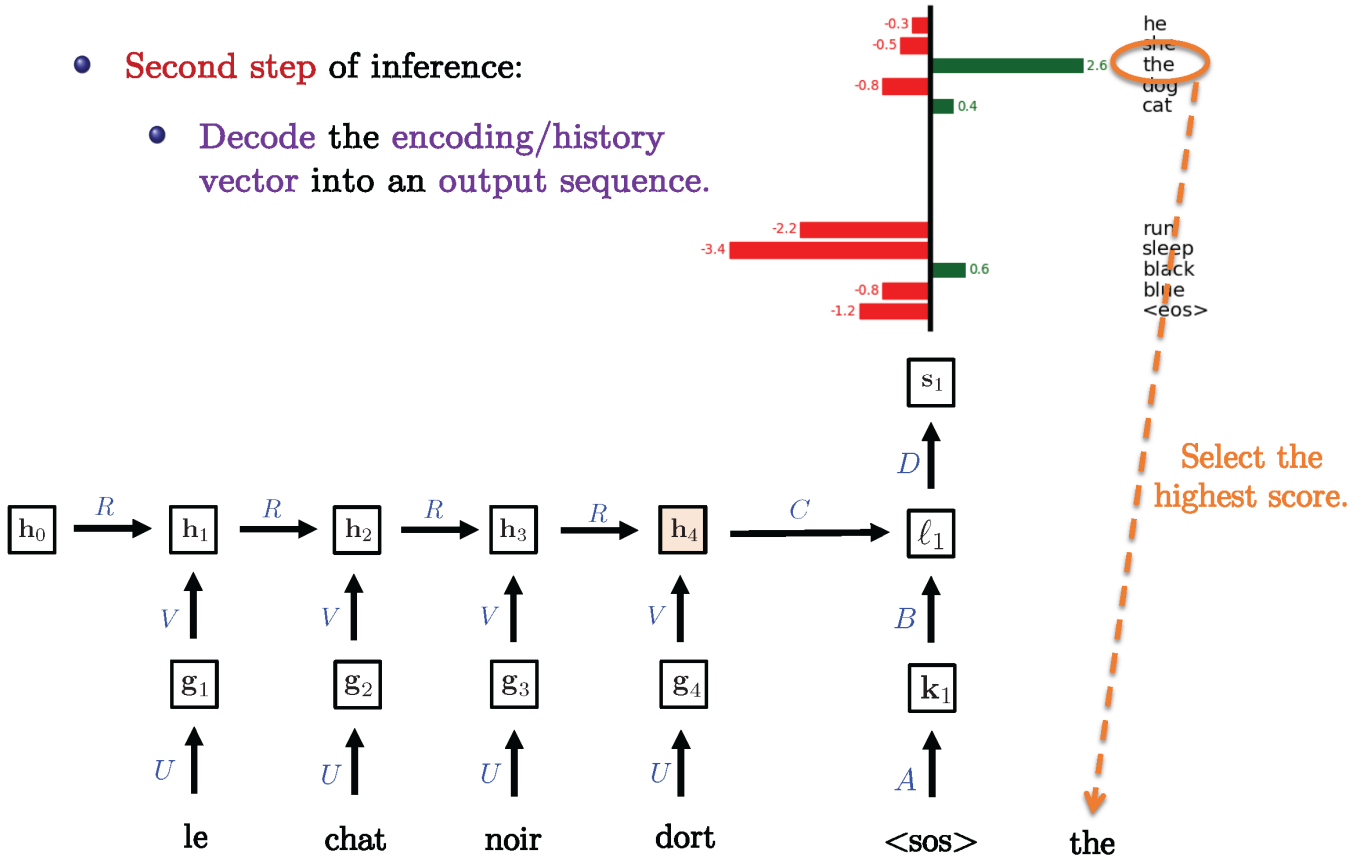
- Second step of inference:
  - Decode the encoding/history vector into an output sequence.





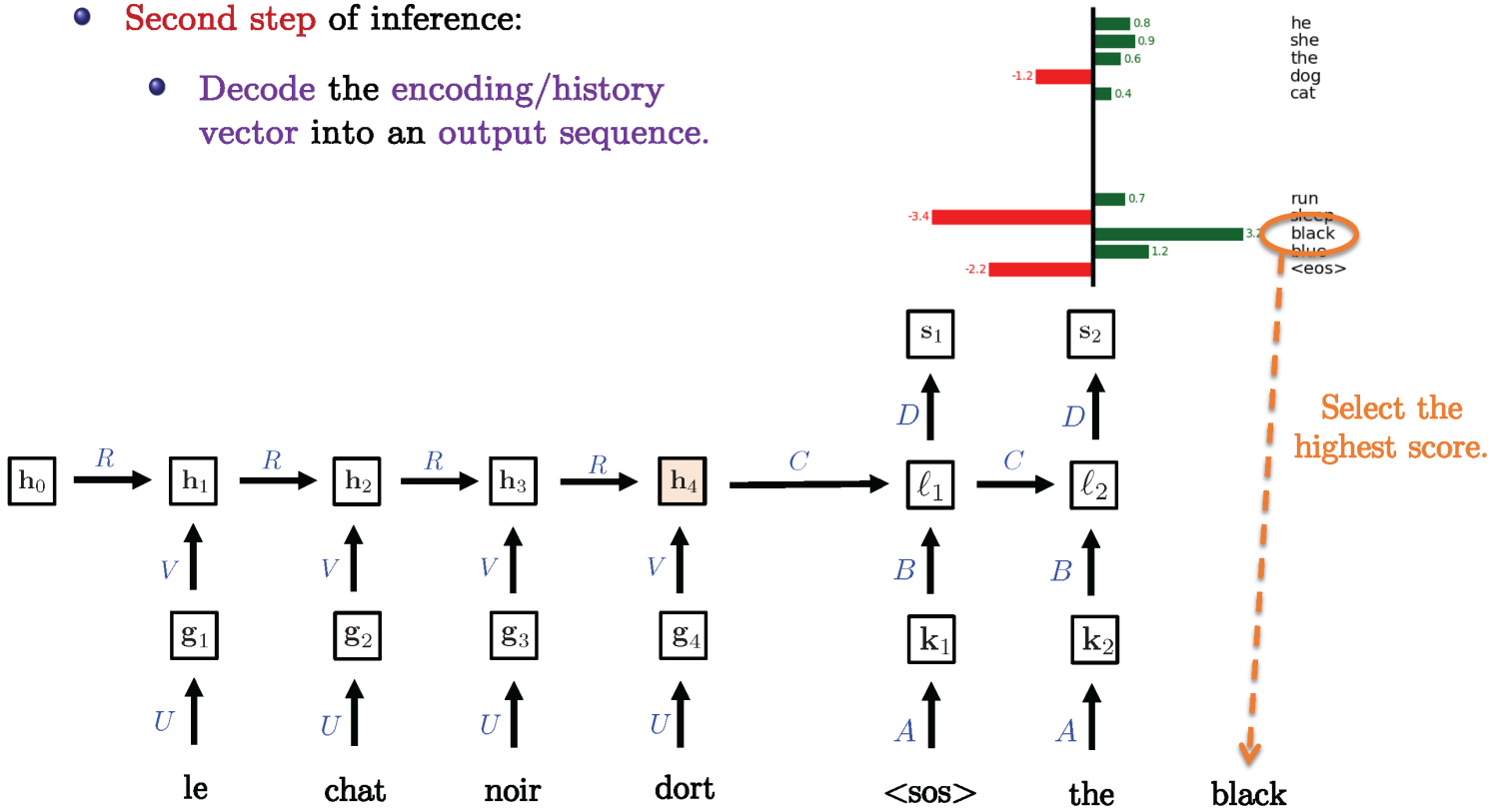
Inference

- Second step of inference:
  - Decode the encoding/history vector into an output sequence.



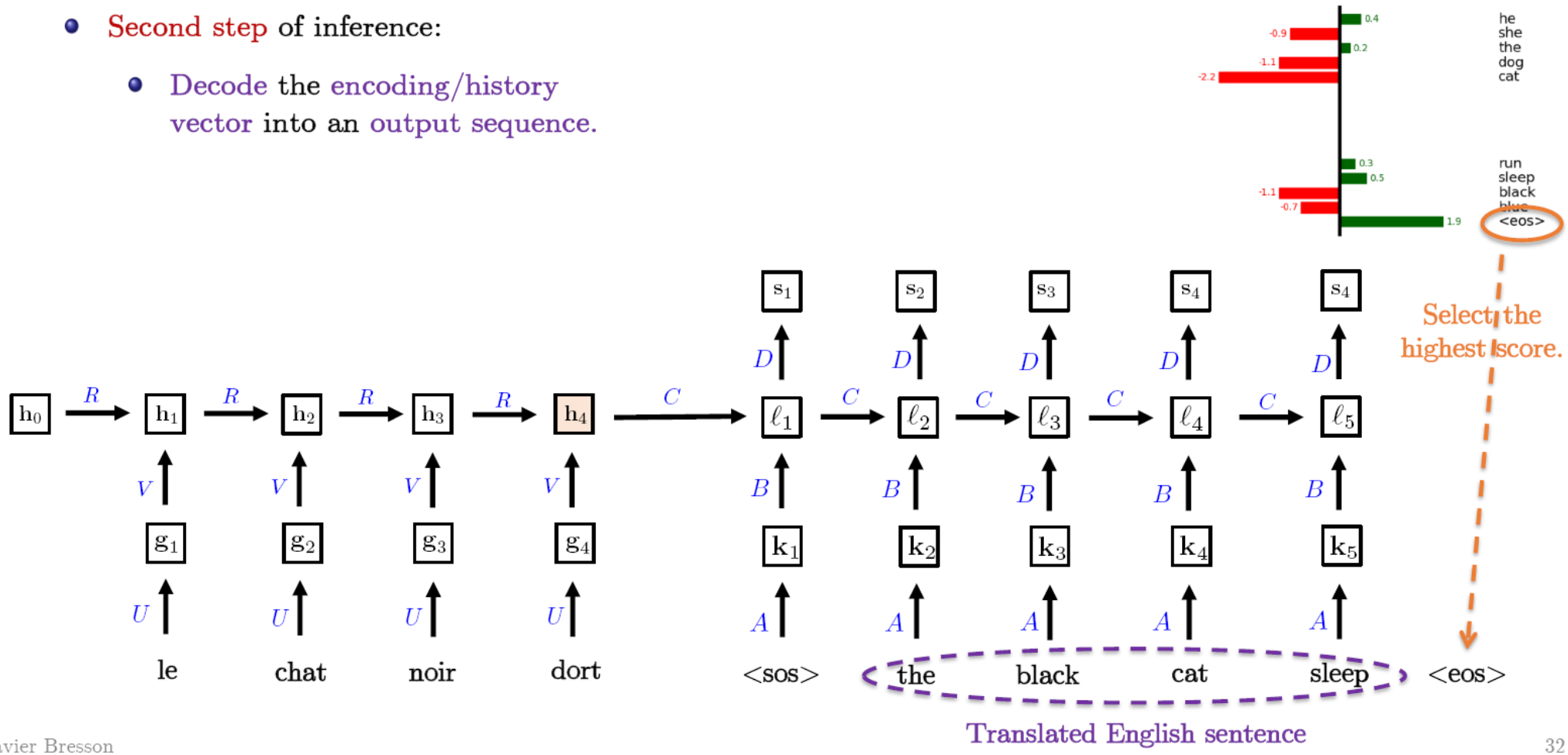
Inference

- Second step of inference:
  - Decode the encoding/history vector into an output sequence.



Inference

- Second step of inference:
  - Decode the encoding/history vector into an output sequence.



Outline

- Speech recognition
  - Introduction
  - Spectrogram
  - Inference
- **Machine Translation**
  - Introduction
  - Architecture
  - Inference
  - **Training**

Training

- Let us train the network with 10 million French ⇒ English sentences :

Sentence 1	le chat noir dort	<sos> the black cat sleep <eos>
Sentence 2	hier je suis aller a la plage	<sos> yesterday I went to the beach <eos>
Sentence 3	je m'appelle thomas	<sos> my name is thomas <eos>
⋮	⋮	⋮
Sentence 10,000,000	tu est un tres bon joueur de tennis	<sos> you are a very good tennis player <eos>

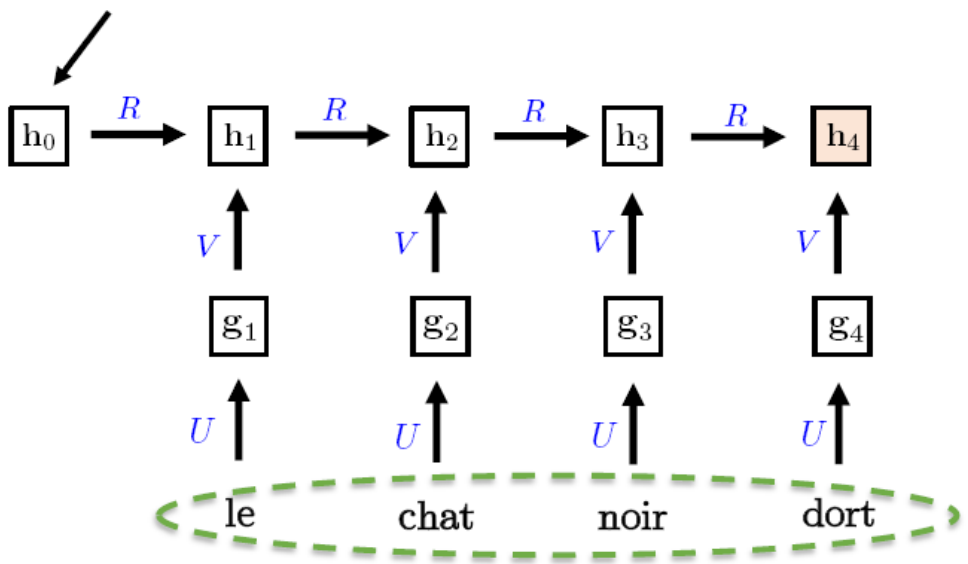
Training

- Let us feed the 1st French ⇒ English sentence :

Labels: the, black, cat, sleeps, <eos>

Translated English sentence 1

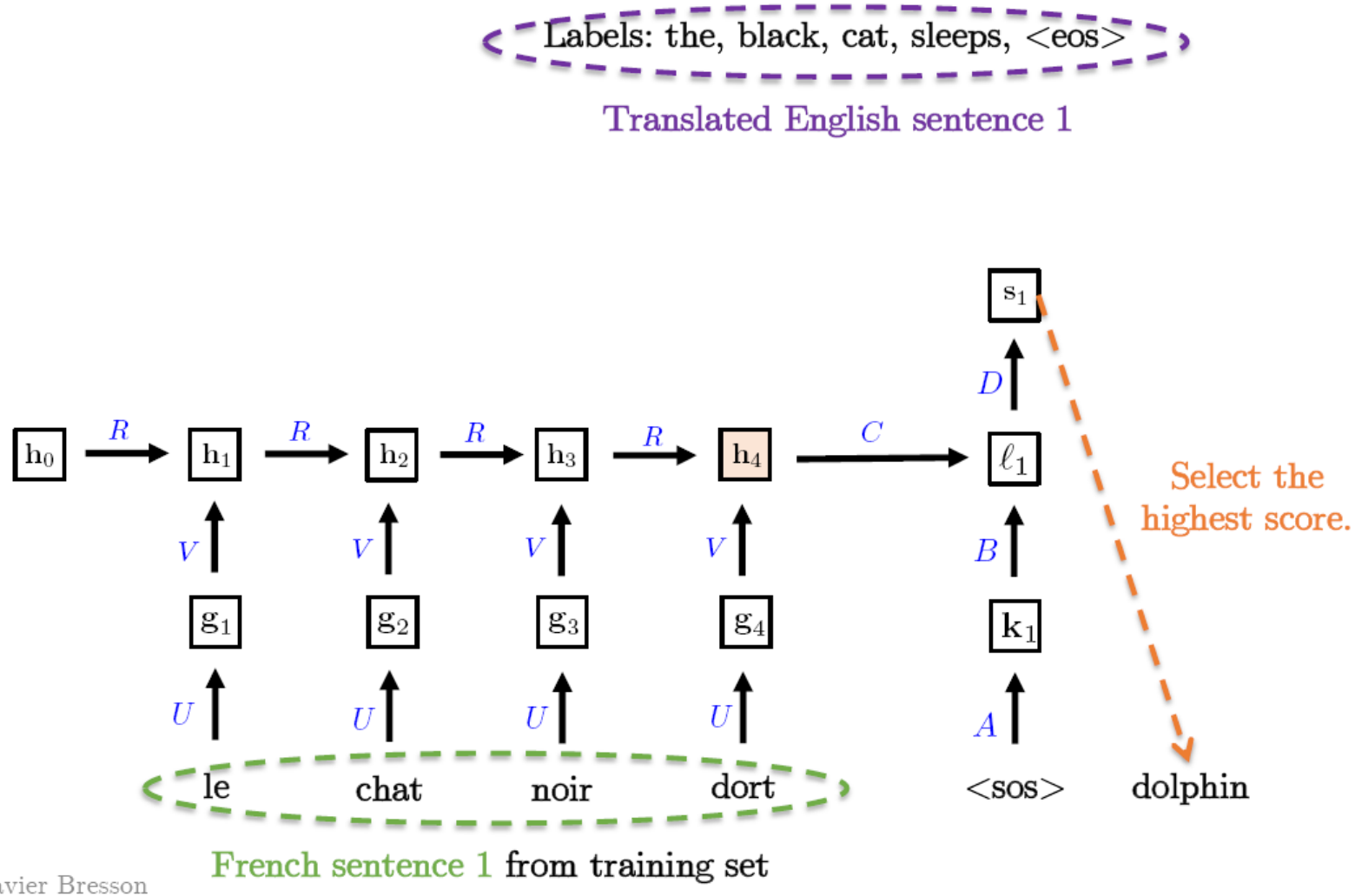
The hidden state is initialized at zero.



French sentence 1 from training set

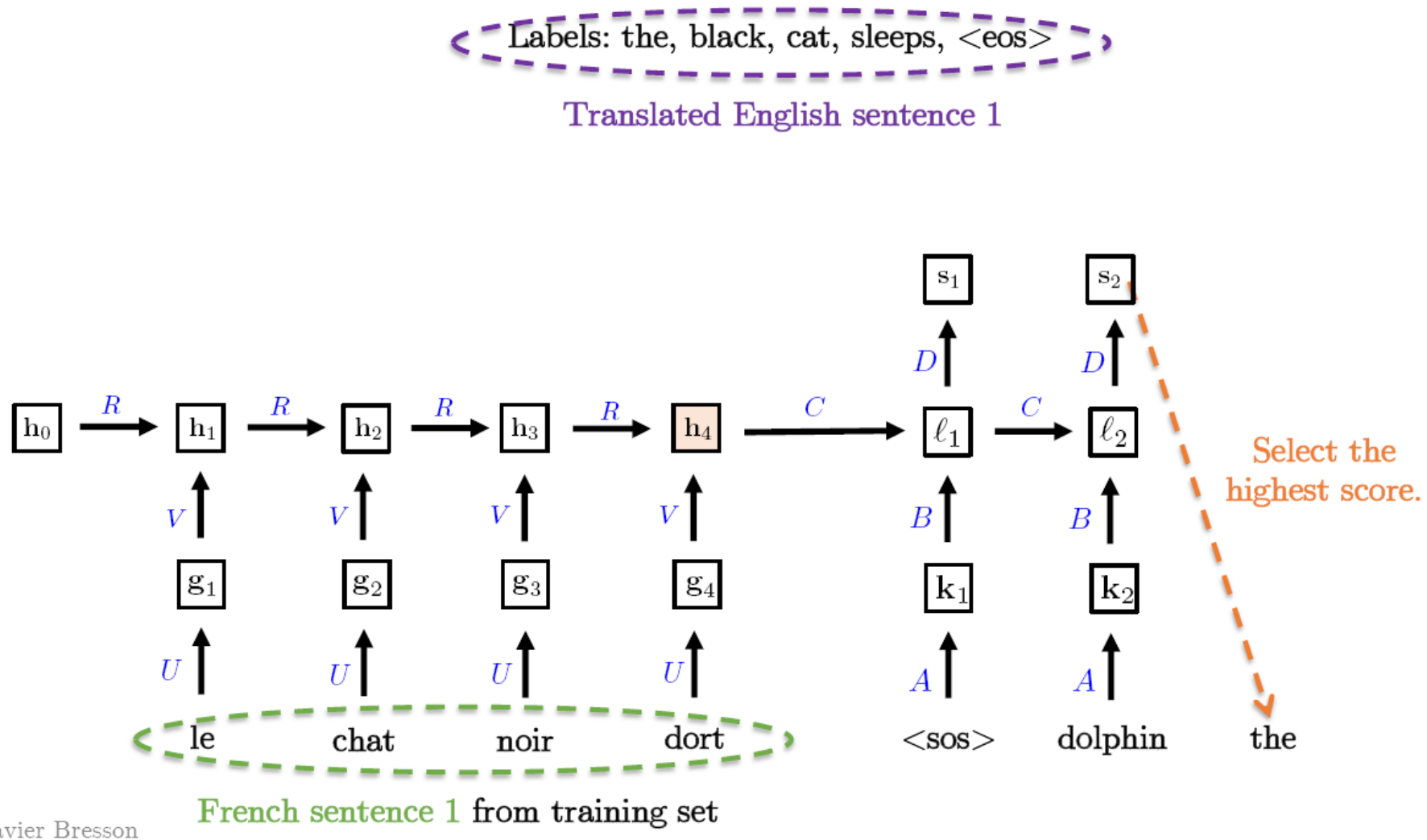
Training

- Let us feed the 1st French ⇒ English sentence :



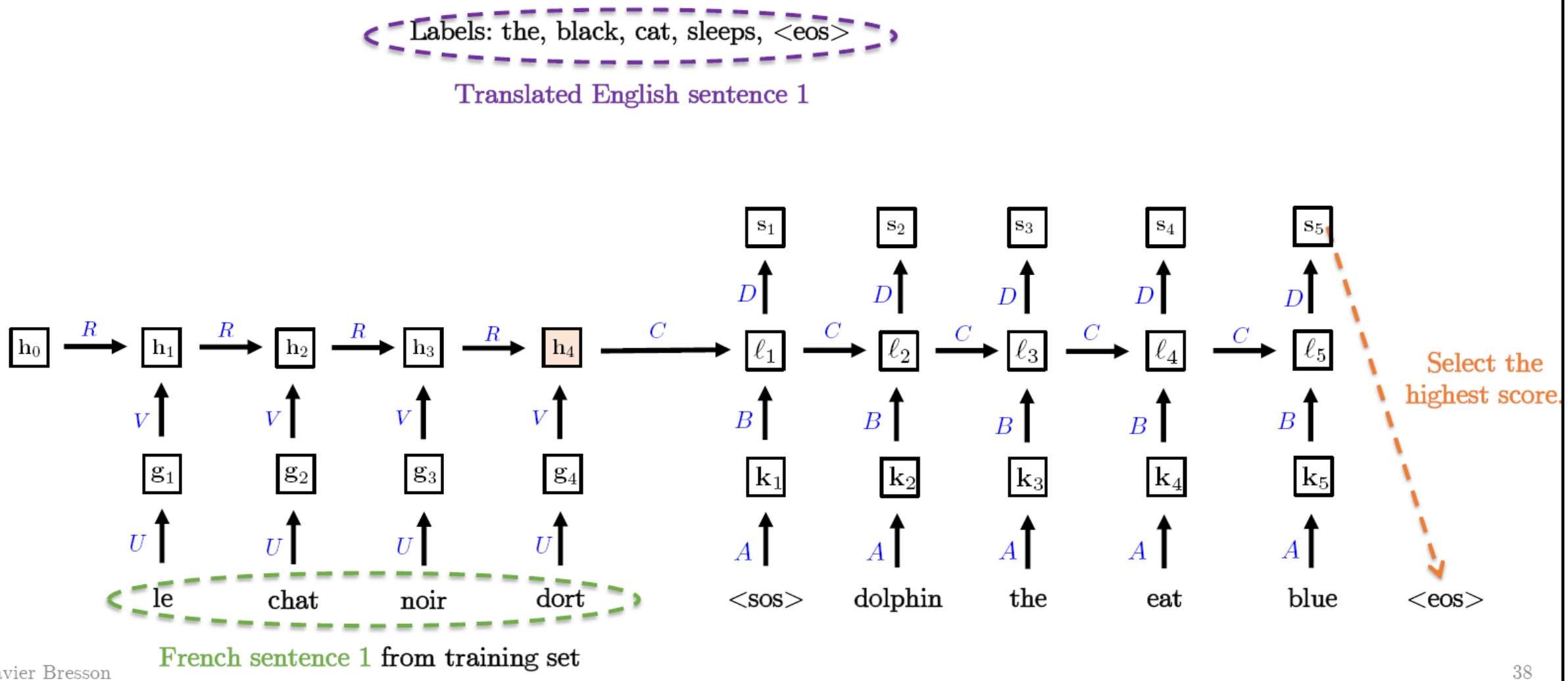
Training

- Let us feed the 1st French ⇒ English sentence :



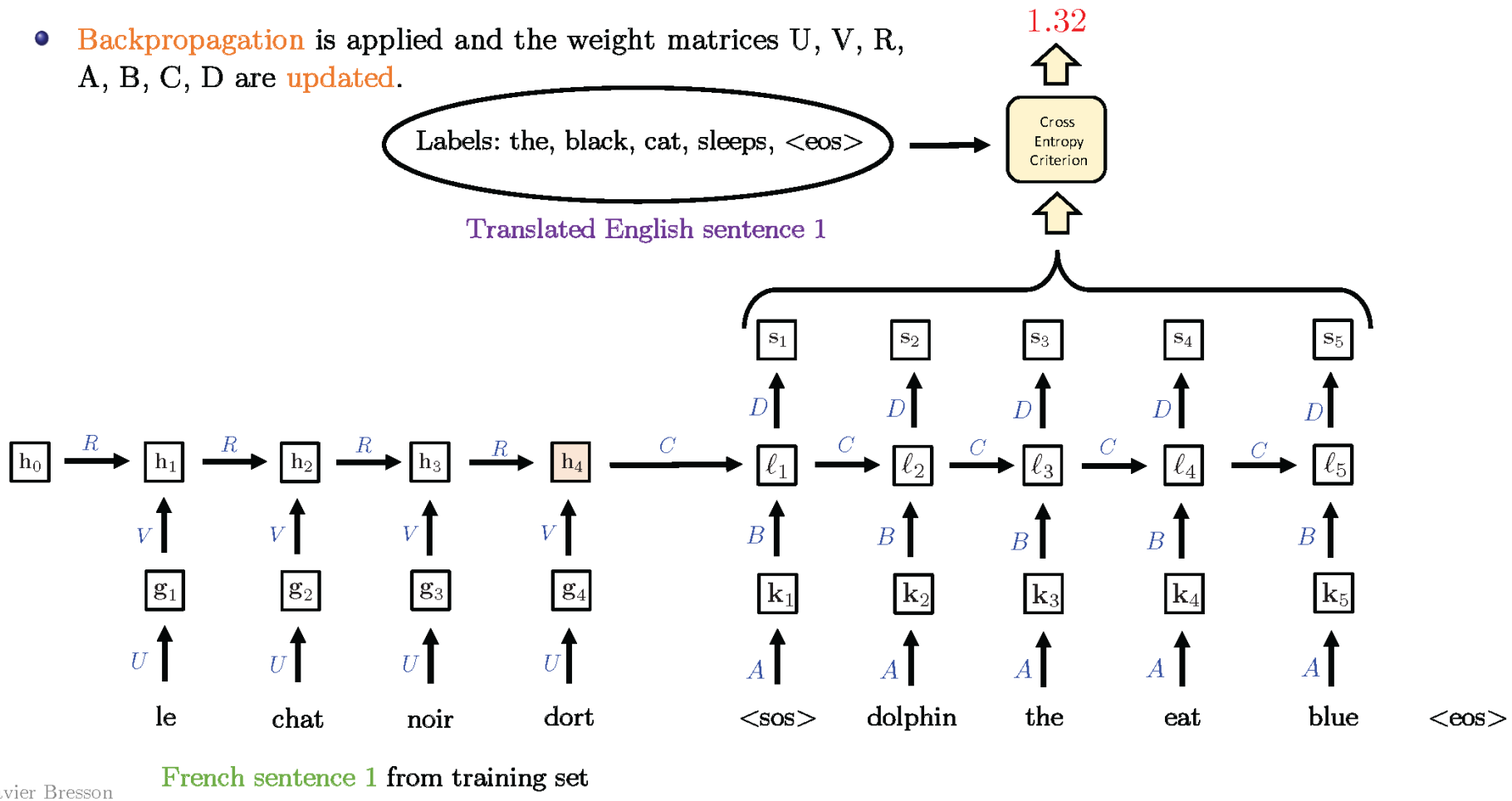
Training

- Let us feed the 1st French ⇒ English sentence :



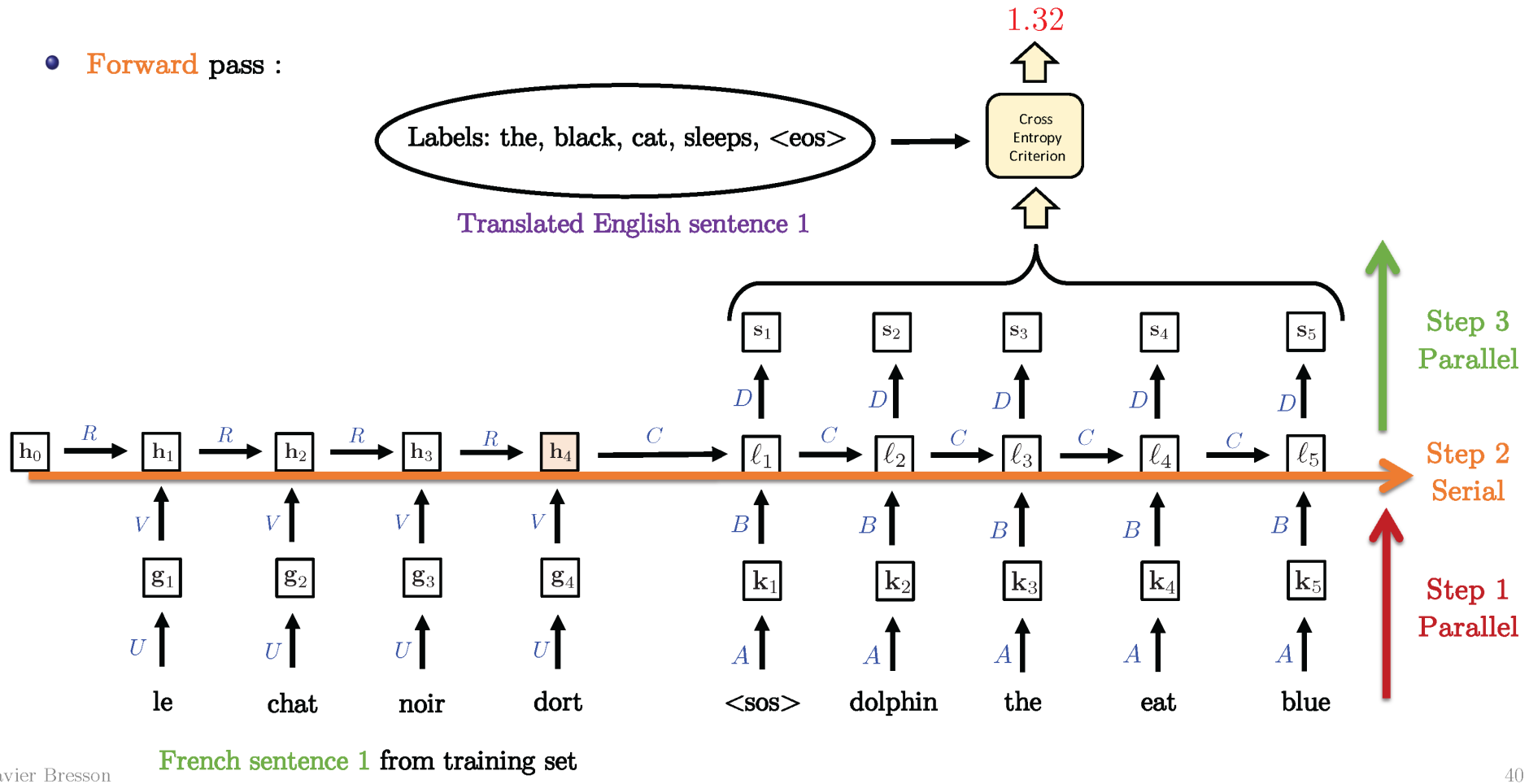
Training

- Backpropagation is applied and the weight matrices U, V, R, A, B, C, D are updated.



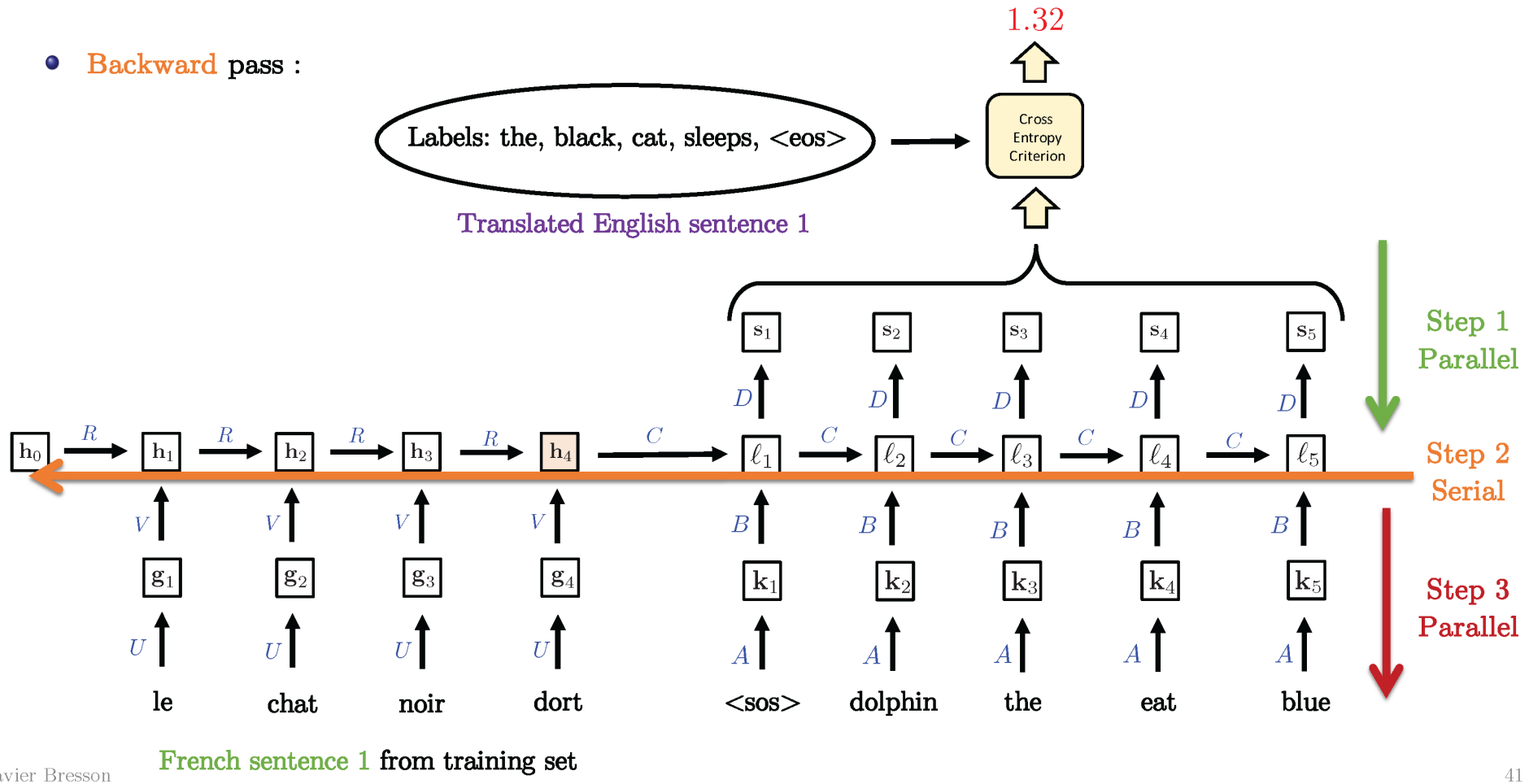
Training

- Forward pass :



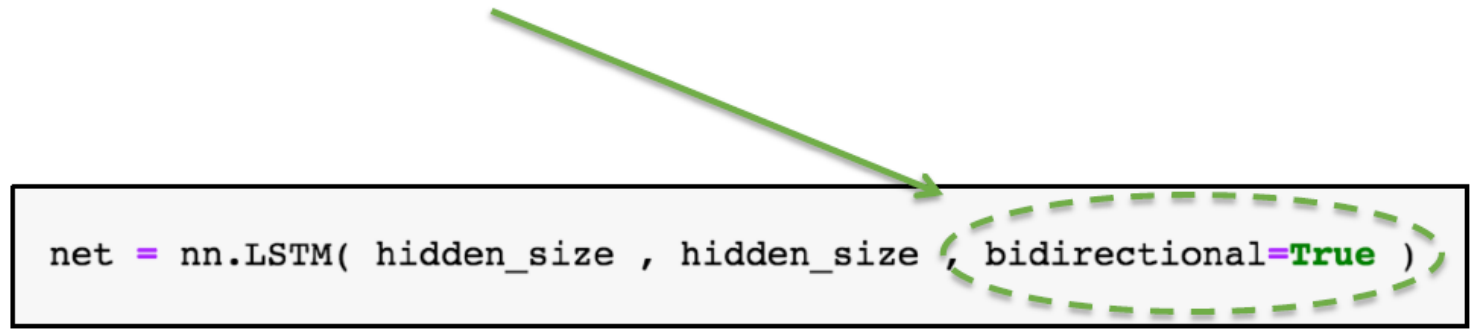
Training

- Backward pass :



Training

- State-of-the-art machine translation machine :
  - Stack 10 layers of bi-directional LSTM :

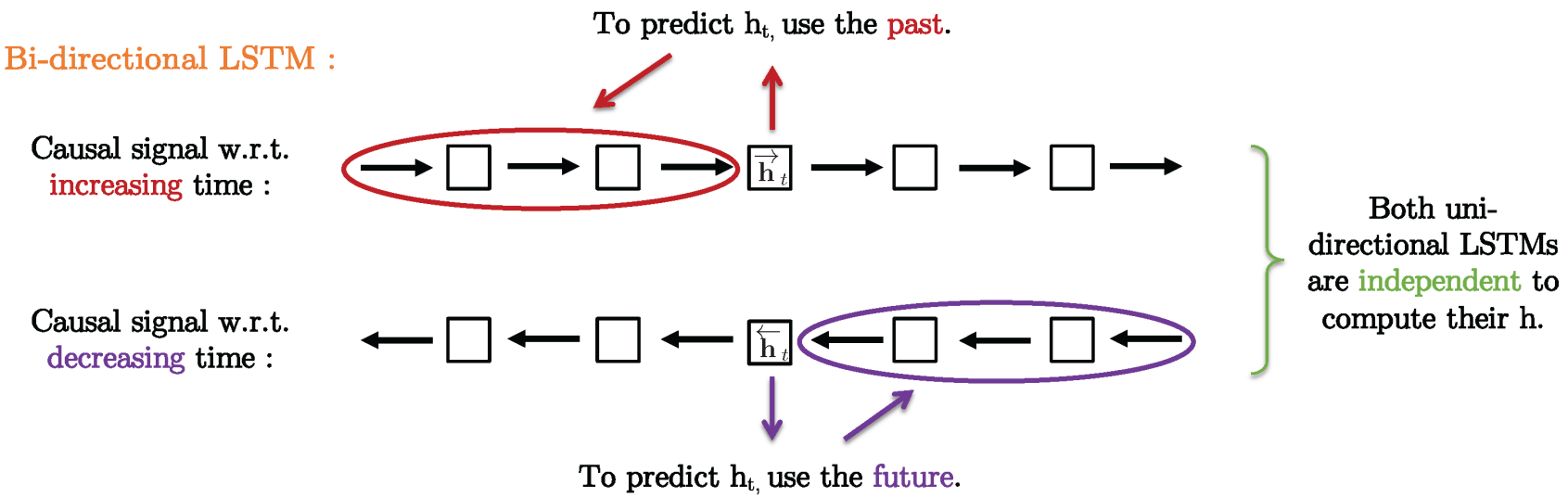


```
net = nn.LSTM( hidden_size , hidden_size , bidirectional=True )
```

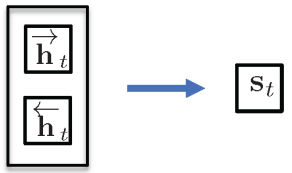
  - Use an attention mechanism.

Training

- Bi-directional LSTM :



Combine both hidden vectors h to better solve NLP tasks (e.g. Q&A)



Seq2Seq Model with Attention

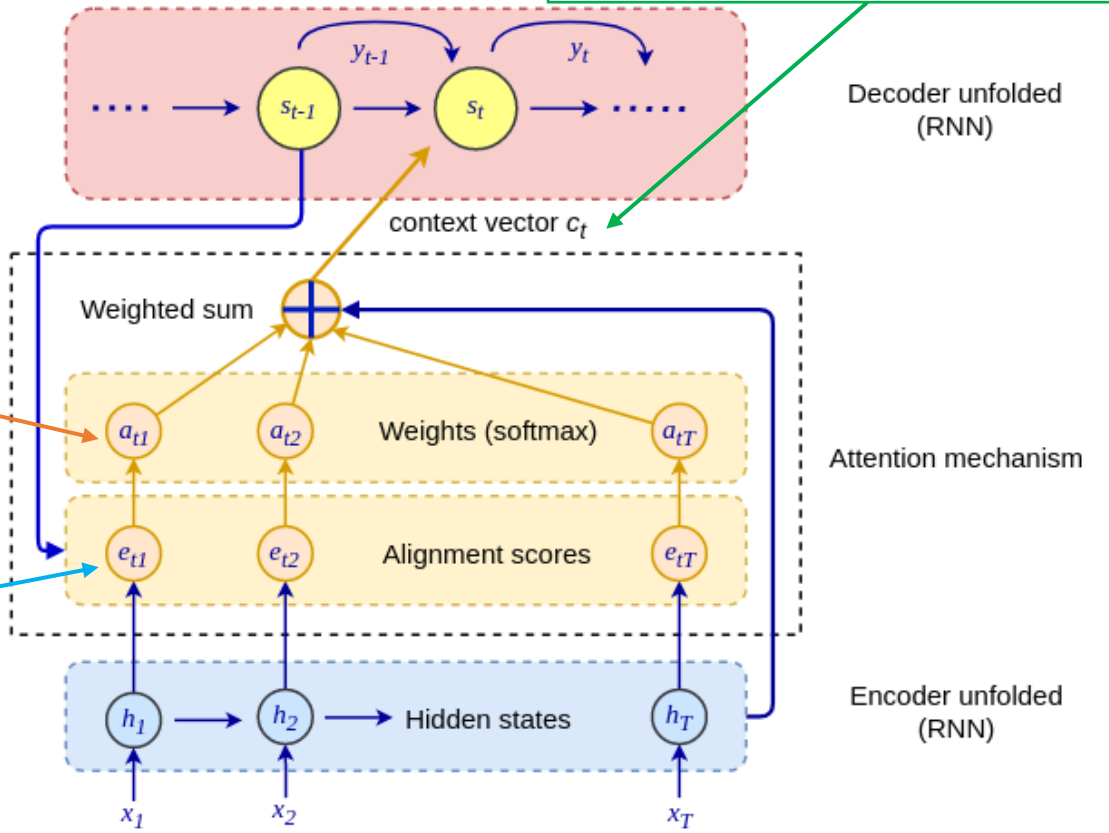
Bahdanau Attention

$$\mathbf{c}_t = \sum_{i=1}^{T} \alpha_{t,i}\mathbf{h}_i = \sum_{i=1}^{T} \frac{exp(e_{t,i})}{\sum_{j=1}^{T} exp(e_{t,j})}\mathbf{h}_i$$

Decoder Hidden States $S = \mathbf{s}_t, t = 1, 2, \ldots$

$$\mathbf{s}_t = f(\mathbf{s}_{t-1}, y_{t-1}, \mathbf{c}_t)$$

Context vector $\mathbf{c}_t = \sum_{i=1}^{T} \alpha_{t,i}\mathbf{h}_i$



Attention Weights

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{j=1}^{T} \exp(e_{t,j})}$$

Alignment Scores

$$e_{t,i} = a(\mathbf{s}_{t-1}, \mathbf{h}_i)$$
$$= \mathbf{v}^T \tanh(\mathbf{W}[\mathbf{h}_i; \mathbf{s}_{t-1}])$$

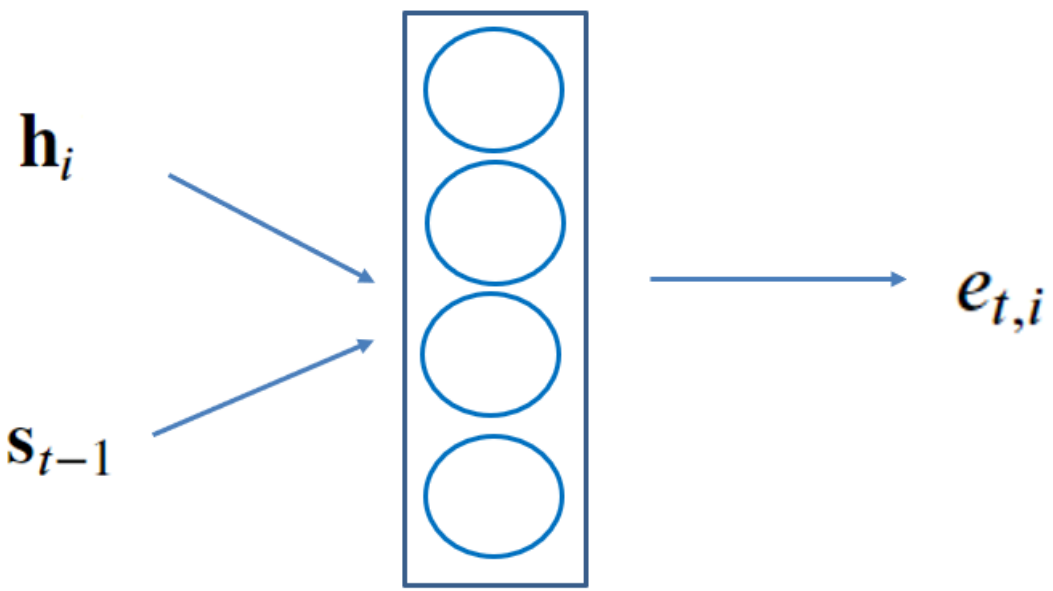
$H = \mathbf{h}_i, i = 1, 2, \ldots . T$

Encoder Hidden States

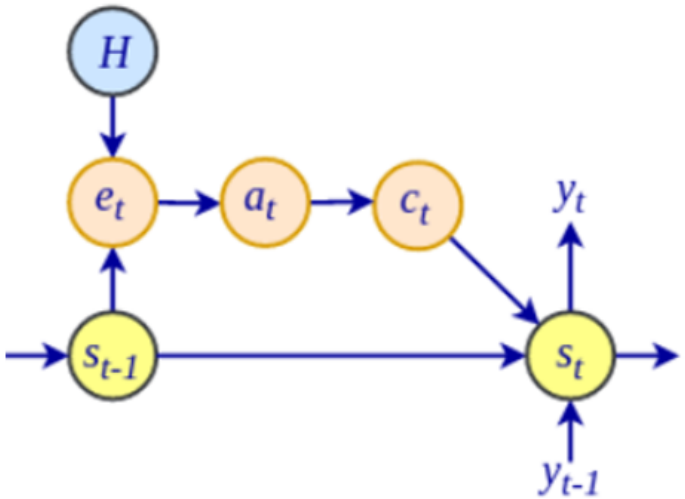
Alignment scores

$$e_{t,i} = a(\mathbf{s}_{t-1}, \mathbf{h}_i) = \begin{cases} \mathbf{s}_{t-1}^T \mathbf{h}_i & \text{dot} \rightarrow \text{mutiply and sum} \\ \mathbf{s}_{t-1}^T \mathbf{W} \mathbf{h}_i & \text{general} \rightarrow \text{a linear model} \\ \mathbf{v}^T \tanh(\mathbf{W}[\mathbf{h}_i; \mathbf{s}_{t-1}]) & \text{concat} \rightarrow \text{a MLP} \end{cases}$$

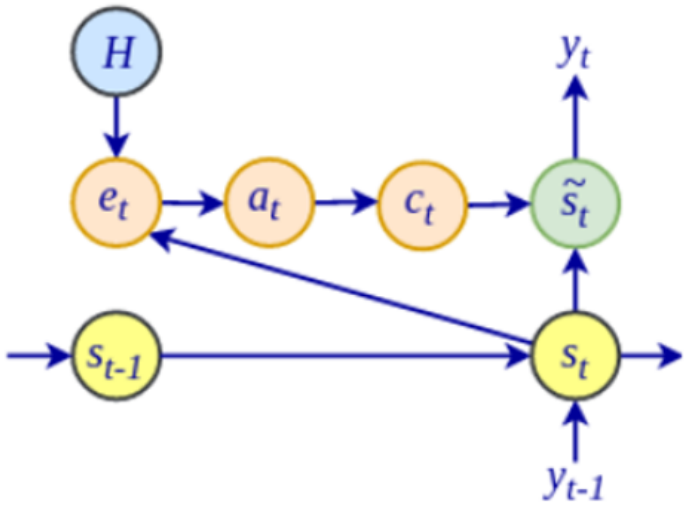
Bahdanau Attention



Bahdanau Attention



Luong Attention



	Bahdanau	Luong
Decoder Hidden State		$\mathbf{s}_t = RNN_{decoder}(\mathbf{s}_{t-1}, y_{t-1})$
Alignment Scores	$e_{t,i} = a(\mathbf{s}_{t-1}, \mathbf{h}_i) = \mathbf{s}_{t-1}^T \mathbf{h}_i$	$e_{t,i} = a(\mathbf{s}_t, \mathbf{h}_i) = \mathbf{s}_t^T \mathbf{h}_i$
Attention Weights	$\alpha_{t,i} = \text{softmax}(e_{t,i}/\mathbf{e}_t)$	$\alpha_{t,i} = \text{softmax}(e_{t,i}/\mathbf{e}_t)$
Context Vector	$\mathbf{c}_t = \sum_{i=1}^{T} \alpha_{t,i} \mathbf{h}_i$	$\mathbf{c}_t = \sum_{i=1}^{T} \alpha_{t,i} \mathbf{h}_i$
Decoder Hidden State (Attentional)	$\mathbf{s}_t = RNN_{decoder}([\mathbf{s}_{t-1}; \mathbf{c}_t], y_{t-1})$	$\tilde{\mathbf{s}}_t = \tanh\left(W_c \left[\mathbf{c}_t; \mathbf{s}_t\right]\right)$
Final Output	$y_t = \text{softmax}(\mathbf{W}_y \mathbf{s}_t)$	$y_t = \text{softmax}(\mathbf{W}_y \tilde{\mathbf{s}}_t)$

Reference

- Bahdanau Attention Paper https://arxiv.org/abs/1409.0473
- Luong Attention Paper https://arxiv.org/abs/1508.04025
- Ivan Vasilev “Advanced Deep Learning with Python”