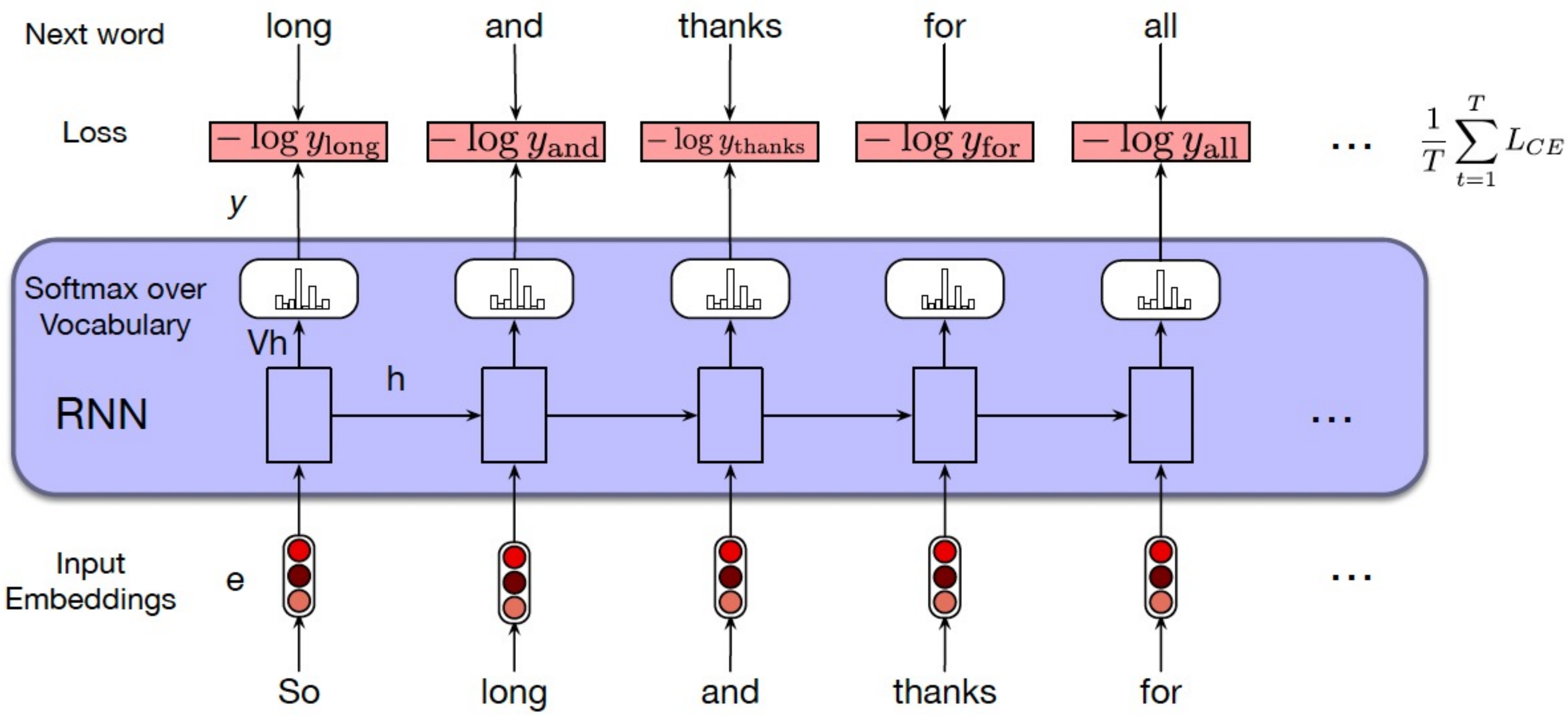
NLP Part 2

Lecture 16

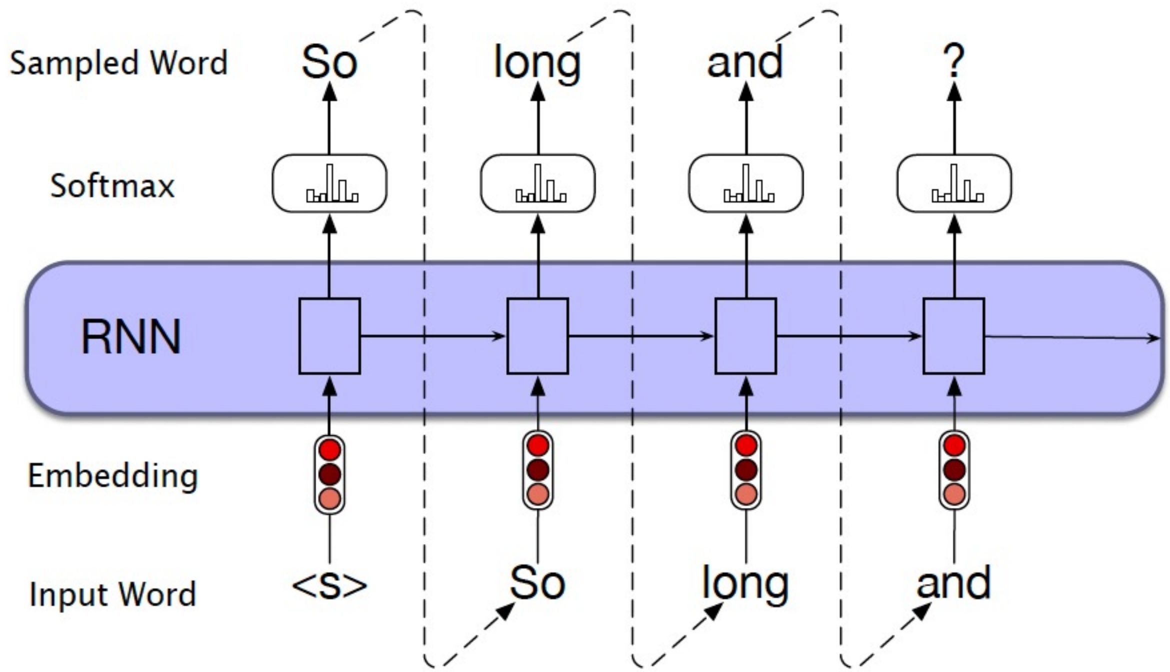
Subir Varma

Training the Language Model



Training the RNN
by trying to predict next word

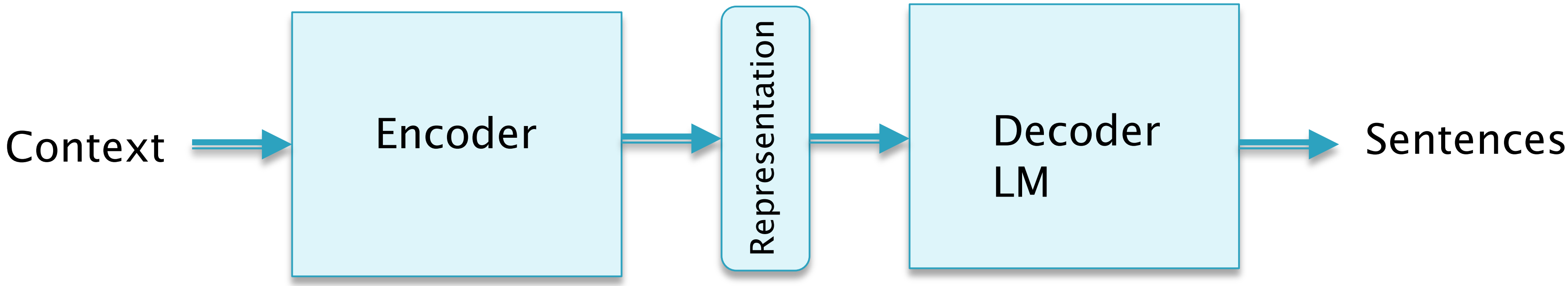
Language Generation



Auto-Regressive Network!

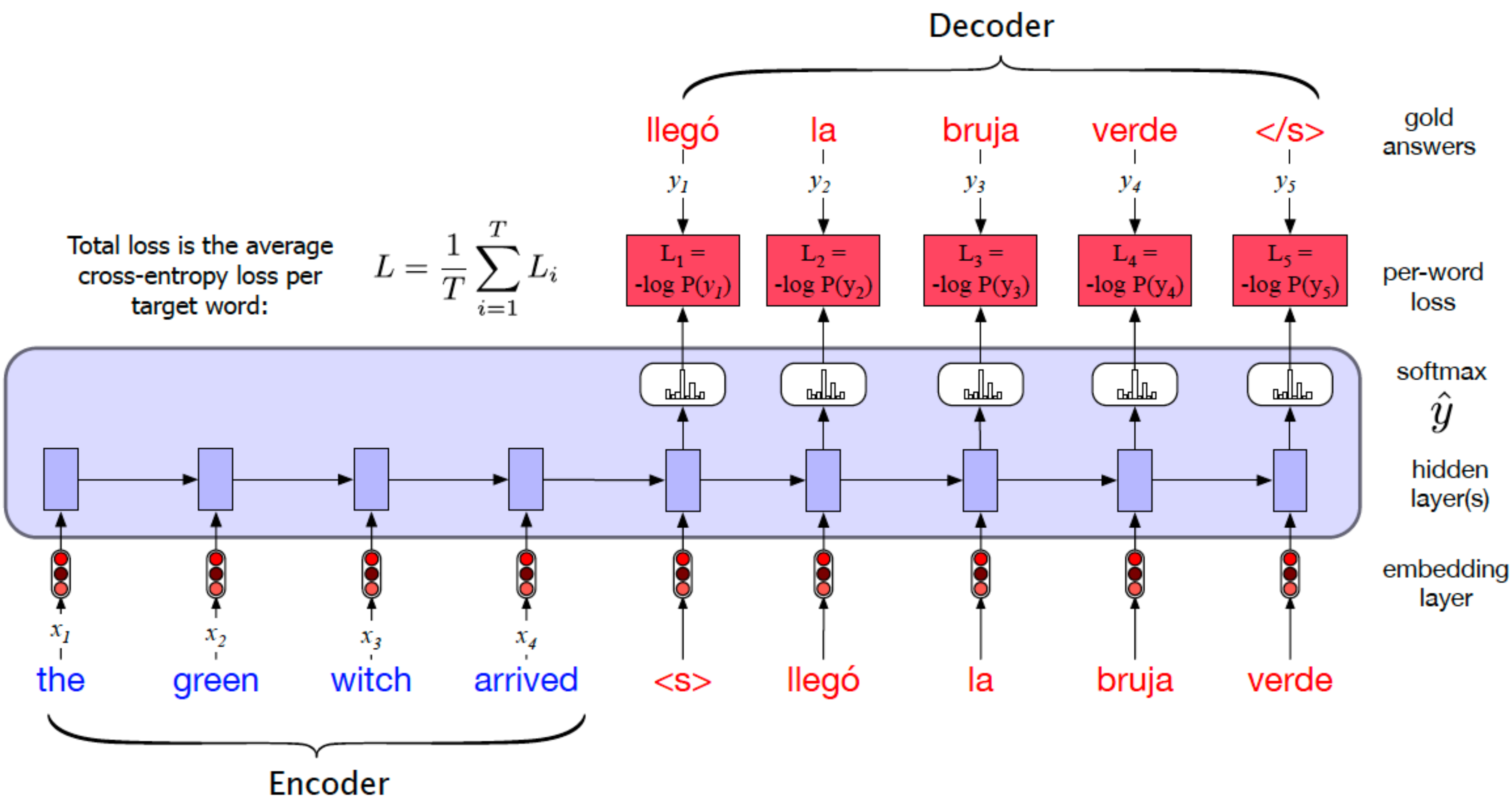
The output of the network serves as its next input

Encoder–Decoder Systems

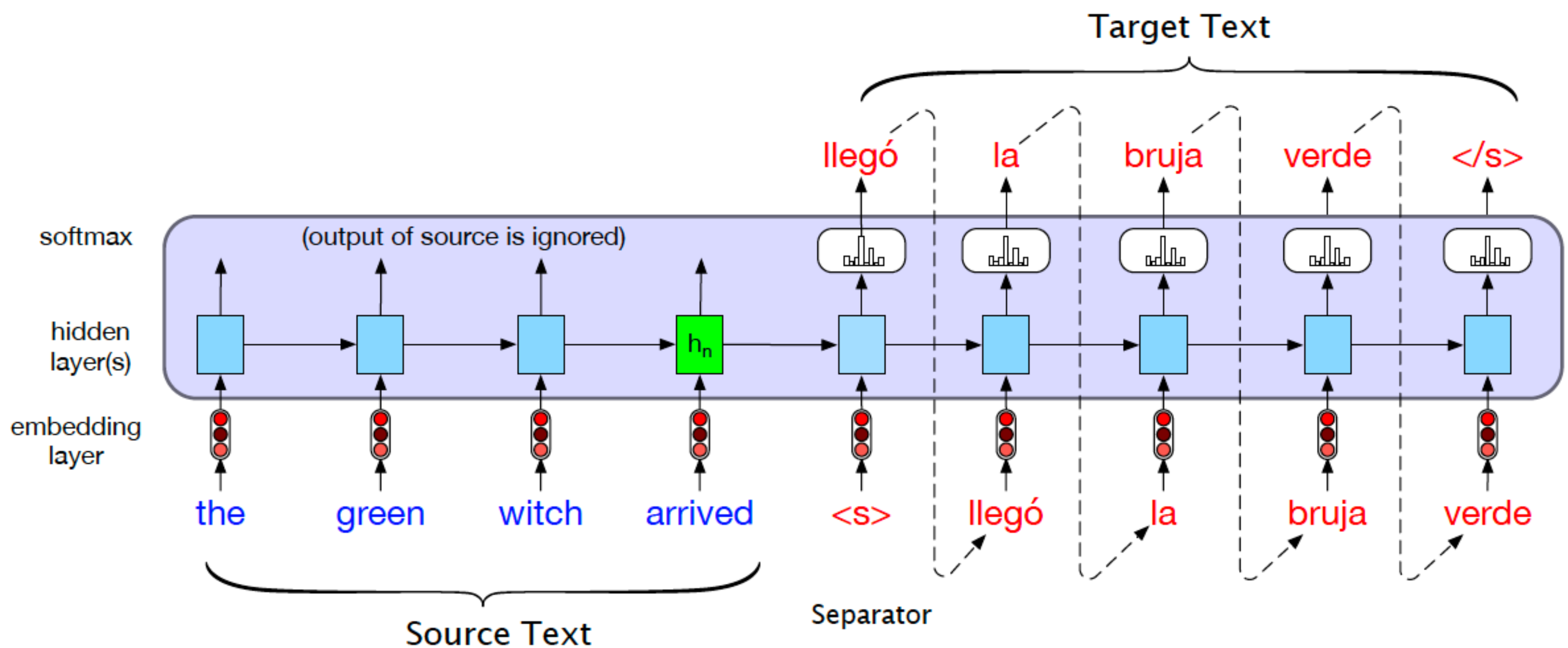


Machine Translation with Encoder Decoder Systems

Translation System: Training

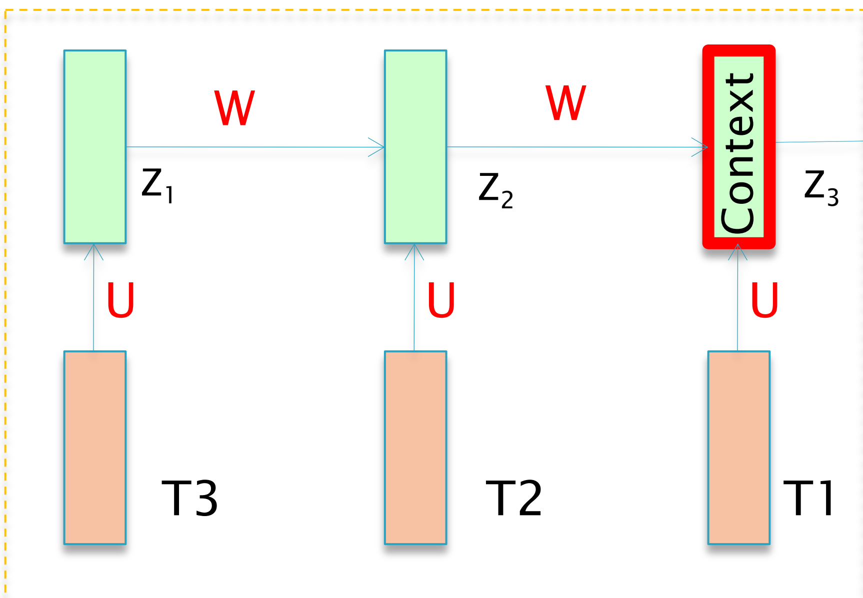


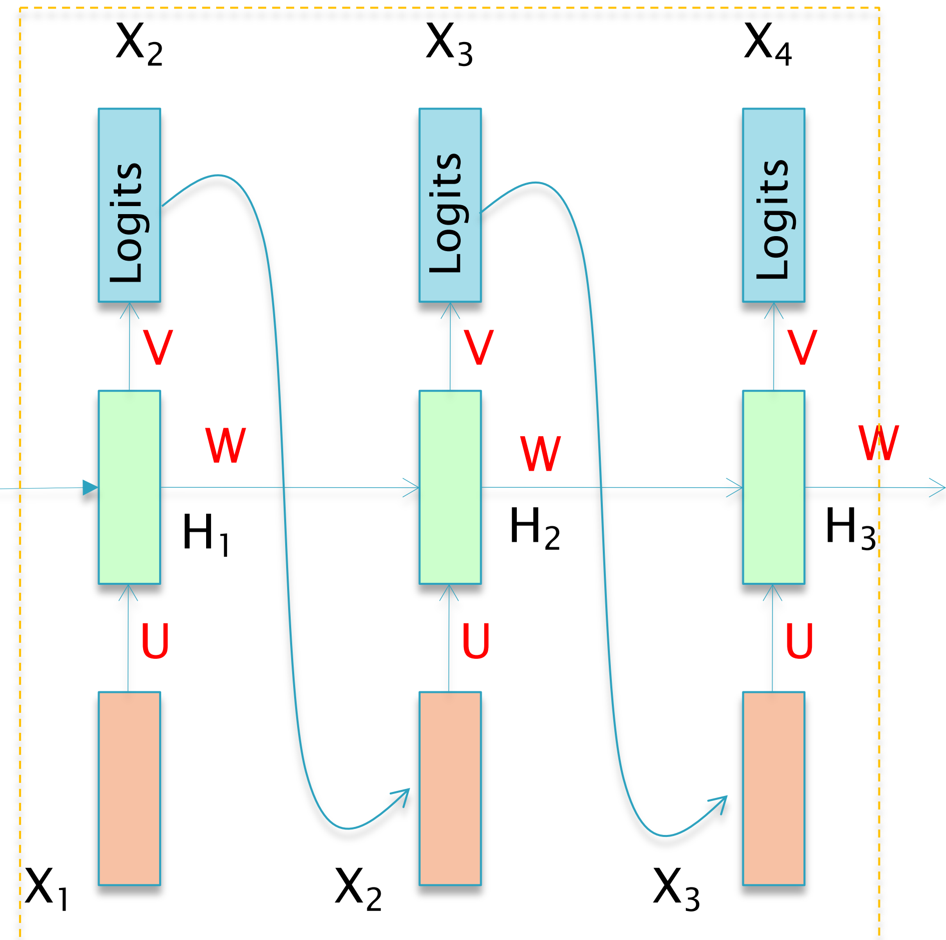
Translation System: Inference



Translation System

- Trained on 12M sentences with 348M French and 304M English words
- Vocab Size: 160K Eng, 80K French
- 348M parameters, 10 days training on 8 GPUs

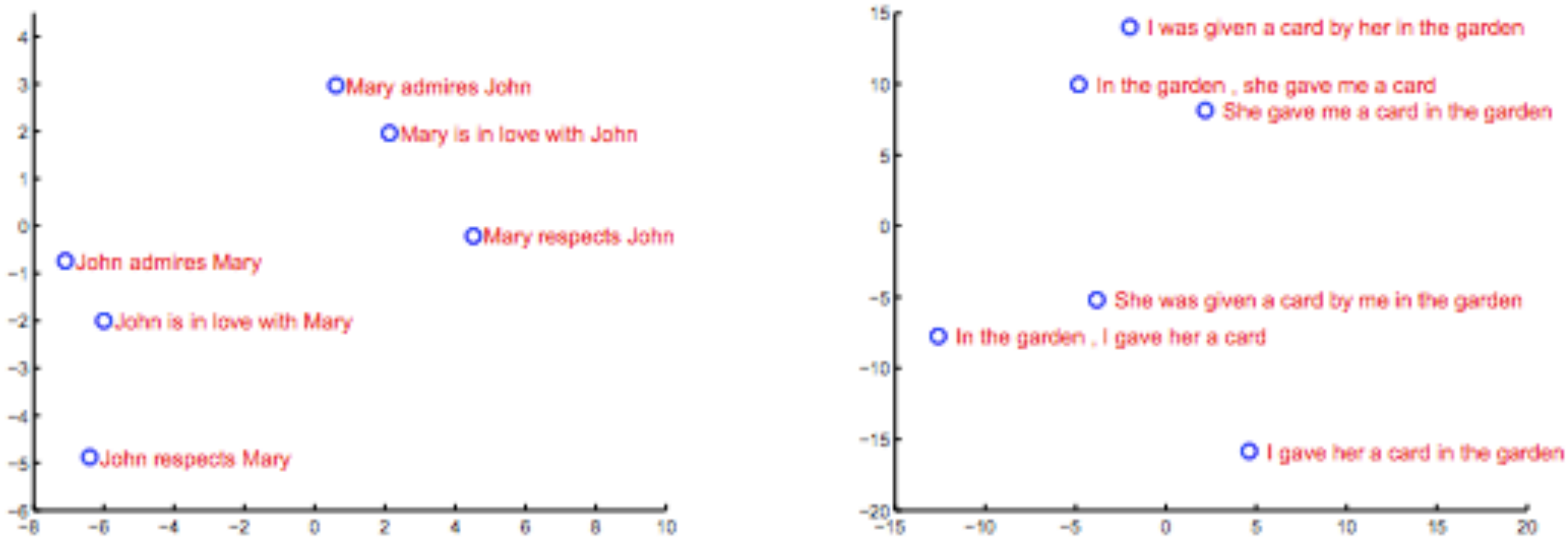




- Uses 5 layers of LSTMs, 1000 cells per LSTM
- 1000 dimensional word embedding
- Reverses the order of the input sequence T_i (Why does this help?)

Best BLEU Score of 34.8

Translation System (Sutskever et.al.): Sentence Representation



Visualization of the Final Hidden State in Encoder

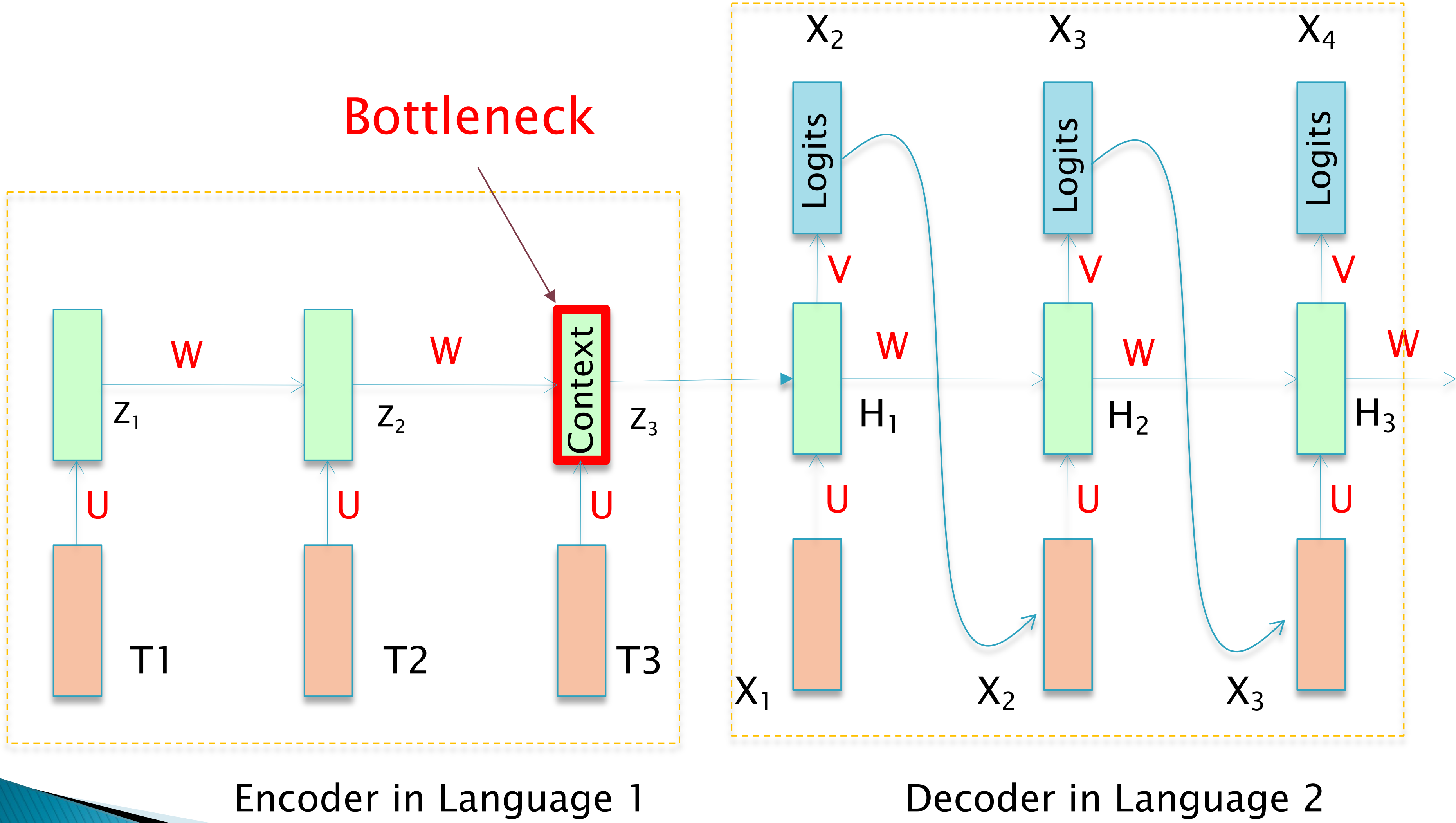
Shows that the representation is sensitive to the order of words

A Problem with Encoder Decoder Architectures

If the input sequence is long, then the error rate increases

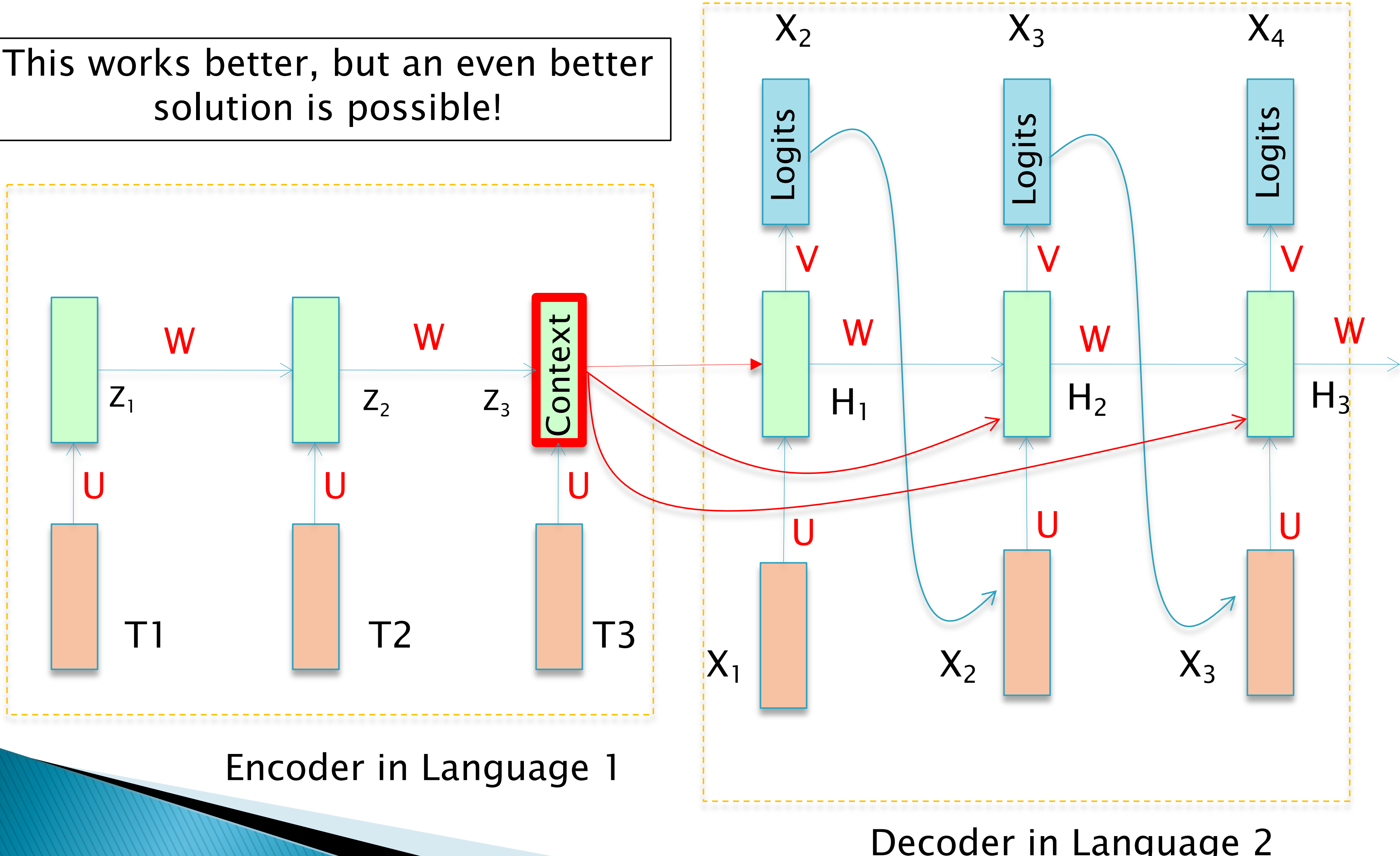
Machine Translation using the Attention Mechanism

Translation System

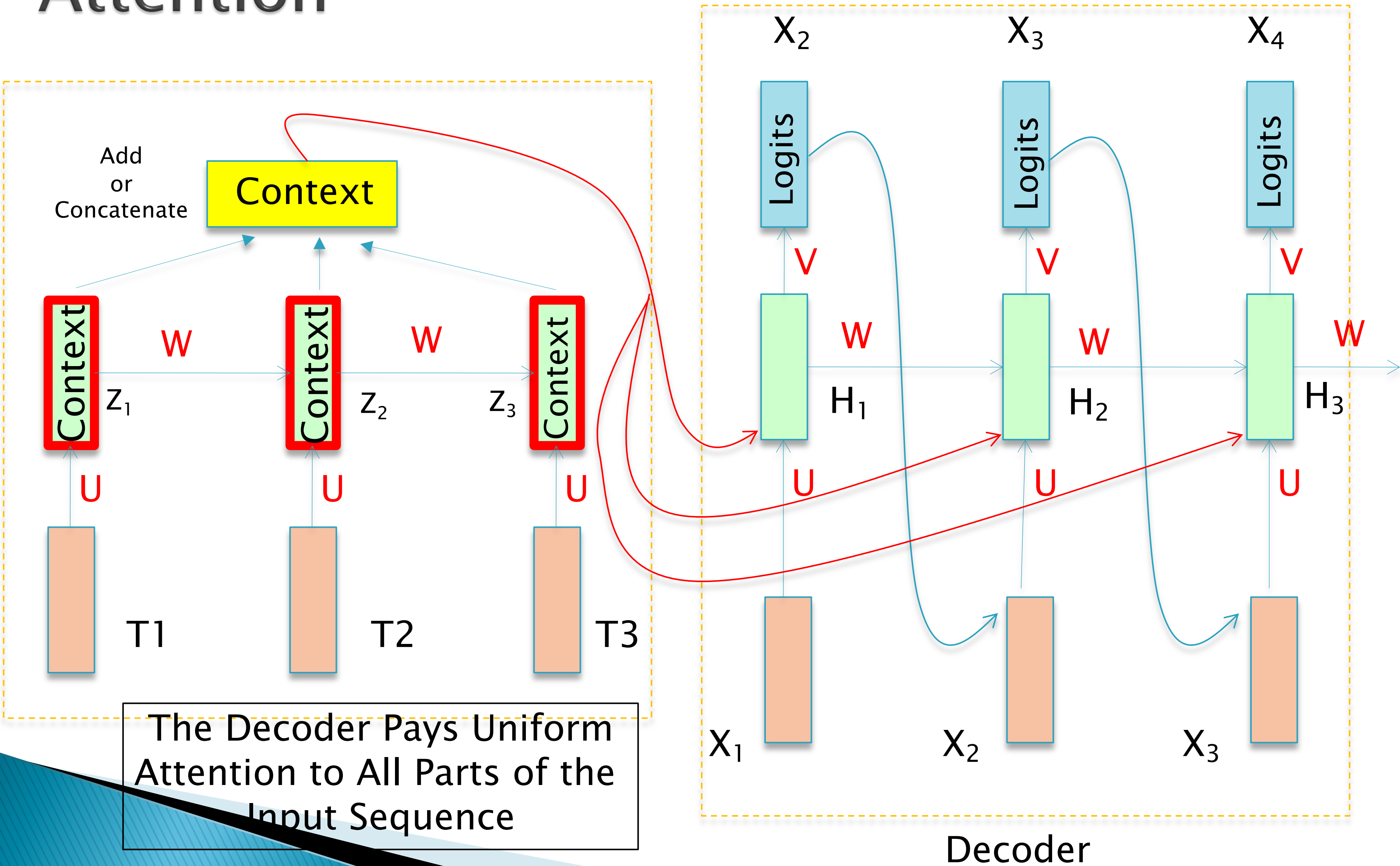


Translation System: Solution 1

This works better, but an even better solution is possible!

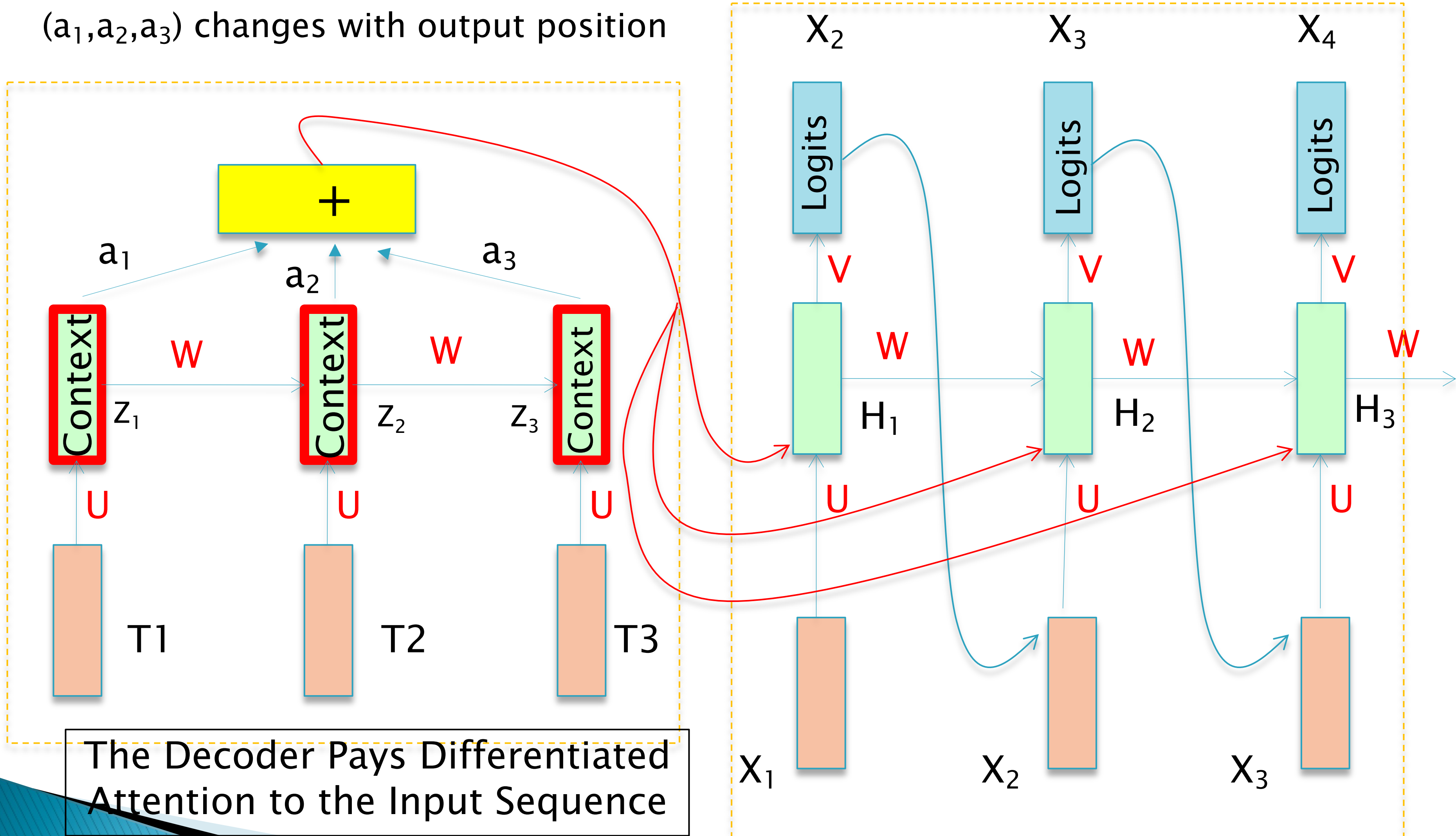


Expanding the Context States: Uniform Attention



The Decoder Pays Uniform Attention to All Parts of the Input Sequence

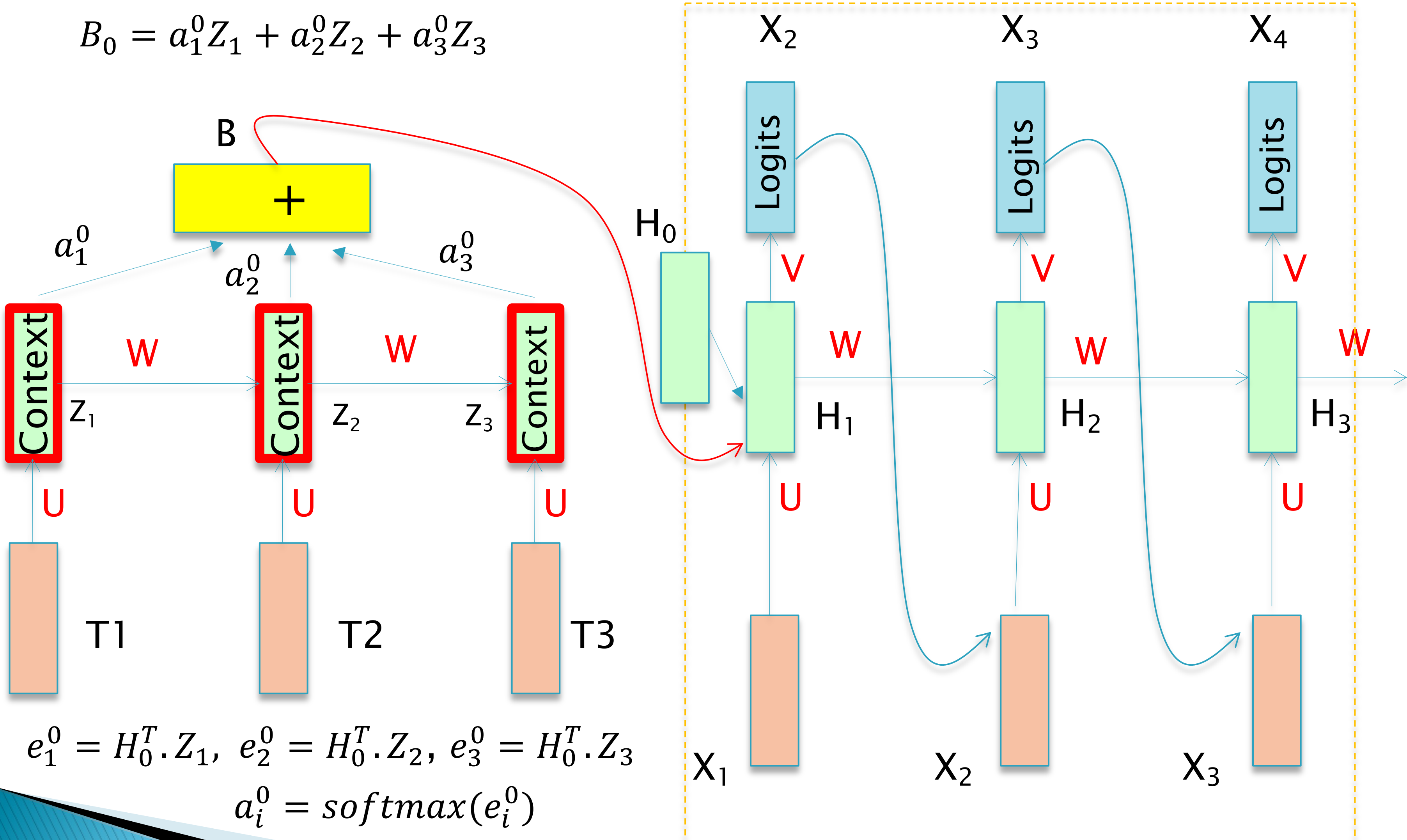
Differentiable (Weighted) Attention



The Decoder Pays Differentiated Attention to the Input Sequence

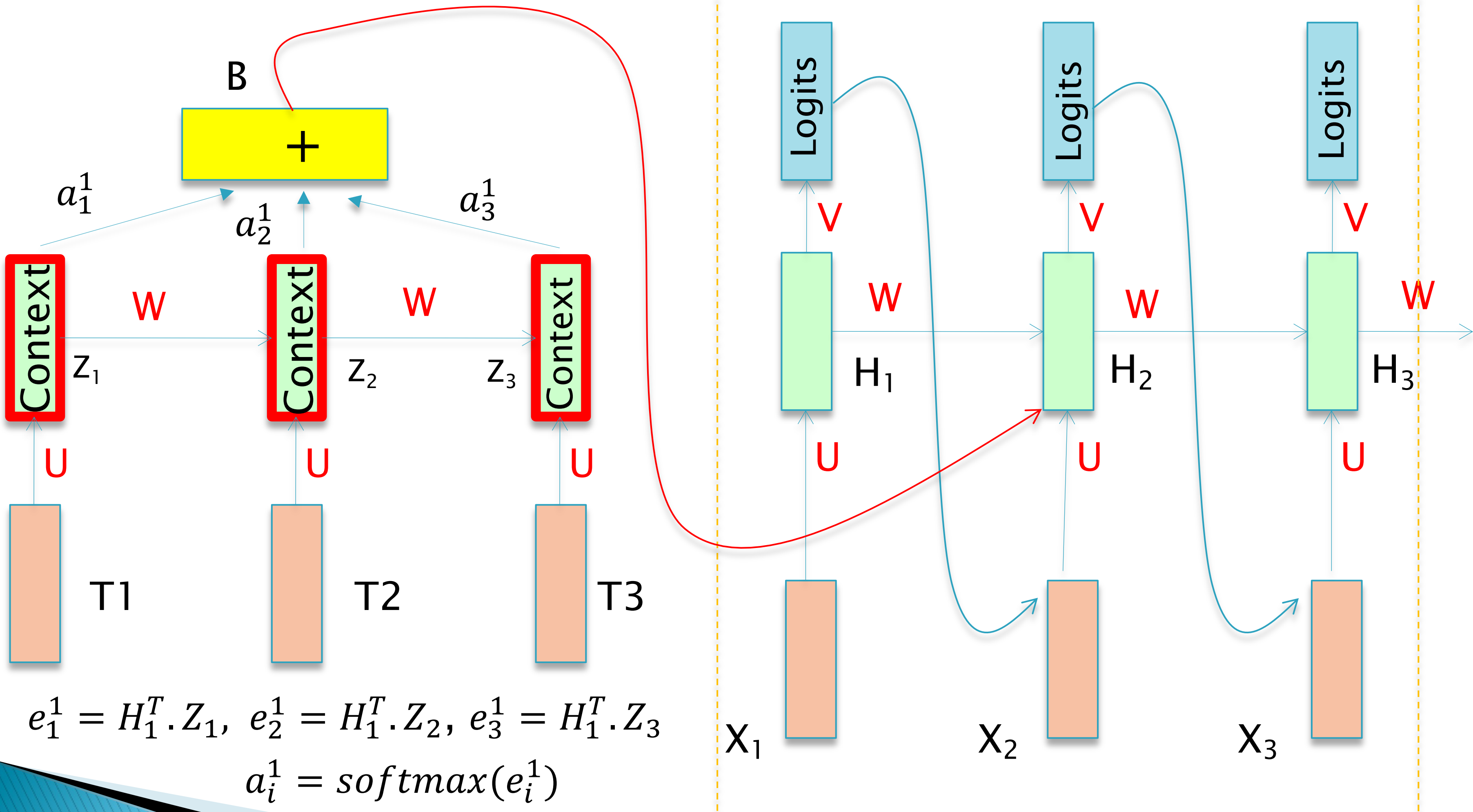
It should focus most on the part of the input that is most relevant to the next output word

Differentiable (Weighted) Attention



Differentiable (Weighted) Attention

$$B_1 = a_1^1 Z_1 + a_2^1 Z_2 + a_3^1 Z_3$$

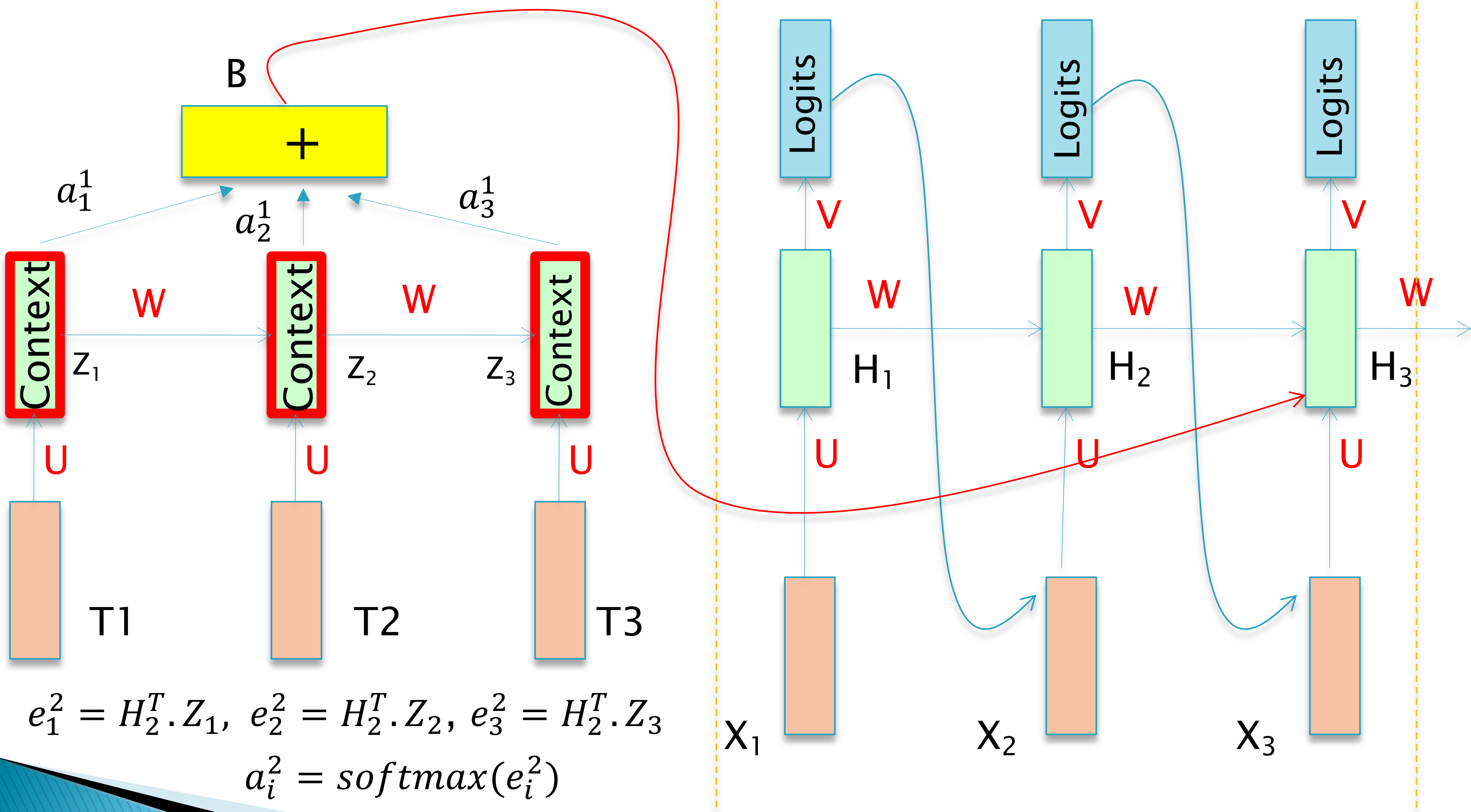


$$e_1^1 = H_1^T . Z_1,\ e_2^1 = H_1^T . Z_2,\ e_3^1 = H_1^T . Z_3$$

$$a_i^1 = softmax(e_i^1)$$

Differentiable (Weighted) Attention

$$B_2 = a_1^2 Z_1 + a_2^2 Z_2 + a_3^2 Z_3$$

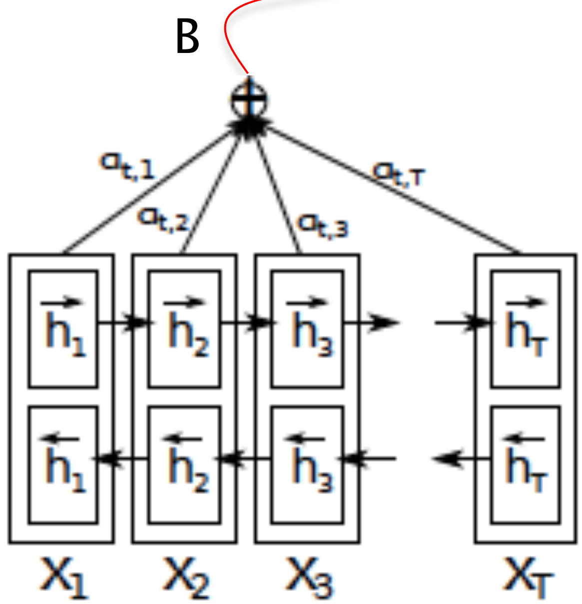


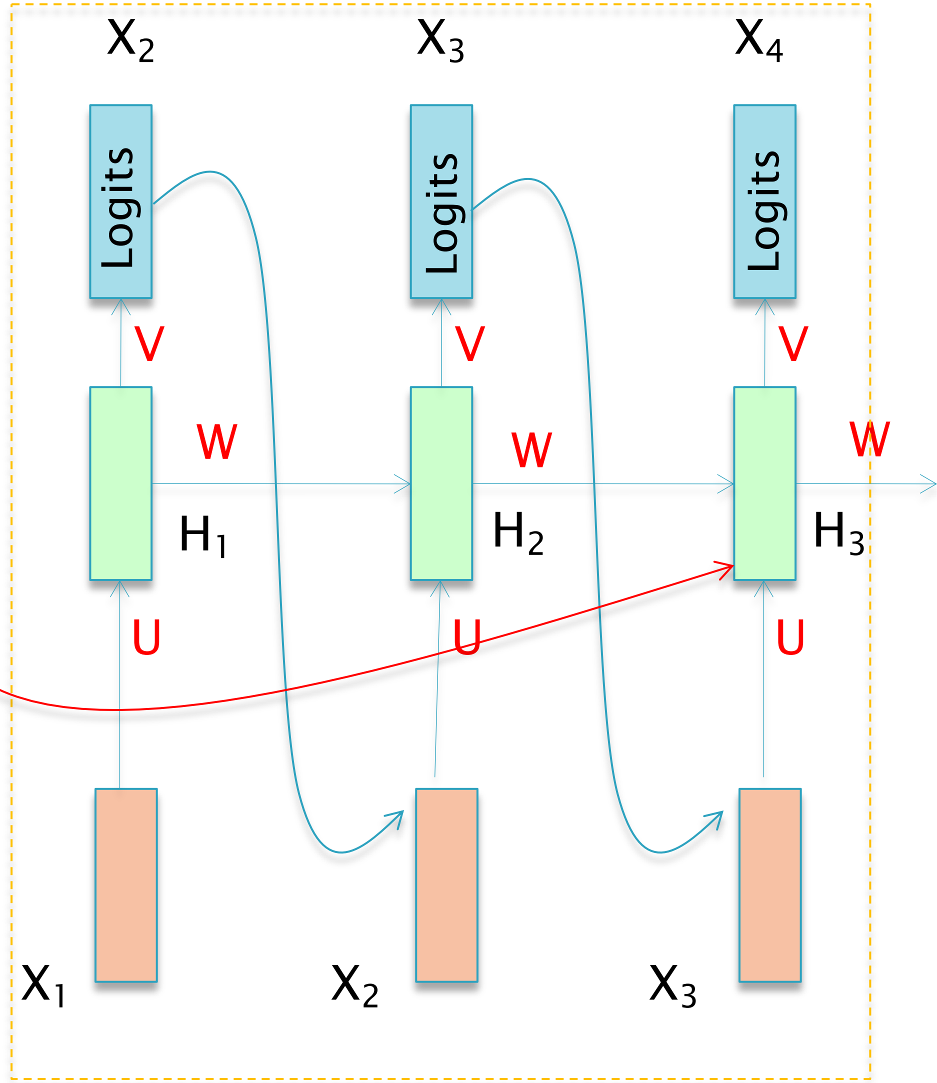
$$e_1^2 = H_2^T.Z_1,\ e_2^2 = H_2^T.Z_2,\ e_3^2 = H_2^T.Z_3$$

$$a_i^2 = softmax(e_i^2)$$

NMT (Bahdanau et.al) https://arxiv.org/pdf/1409.0473.pdf

$$B_2 = a_1^2 Z_1 + a_2^2 Z_2 + a_3^2 Z_3$$





- 348M total words
- Vocabulary: 30K words
- 1000 nodes per cell
- Embedding dim 620

Plot of Attention Values in English to French Translation

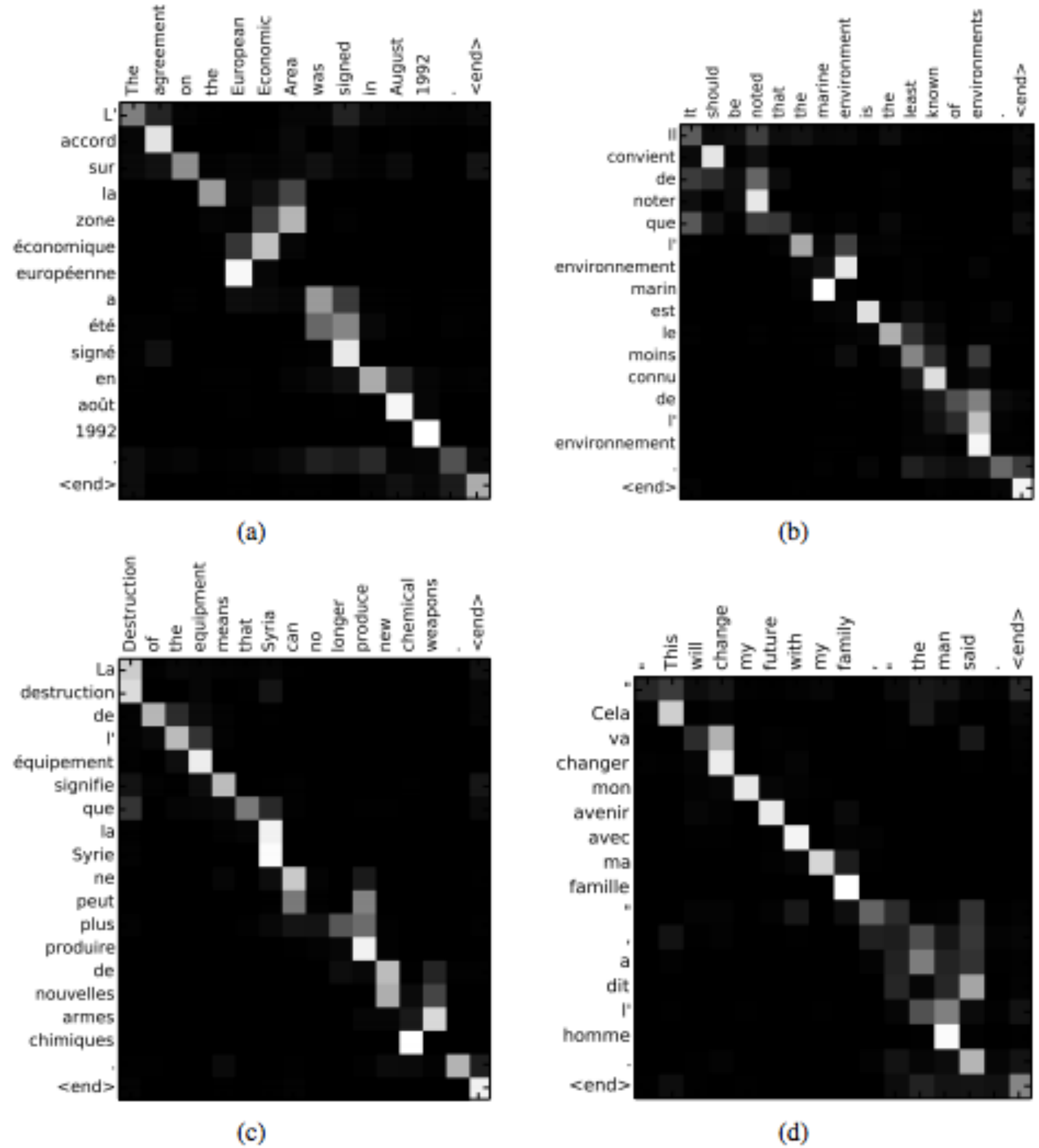
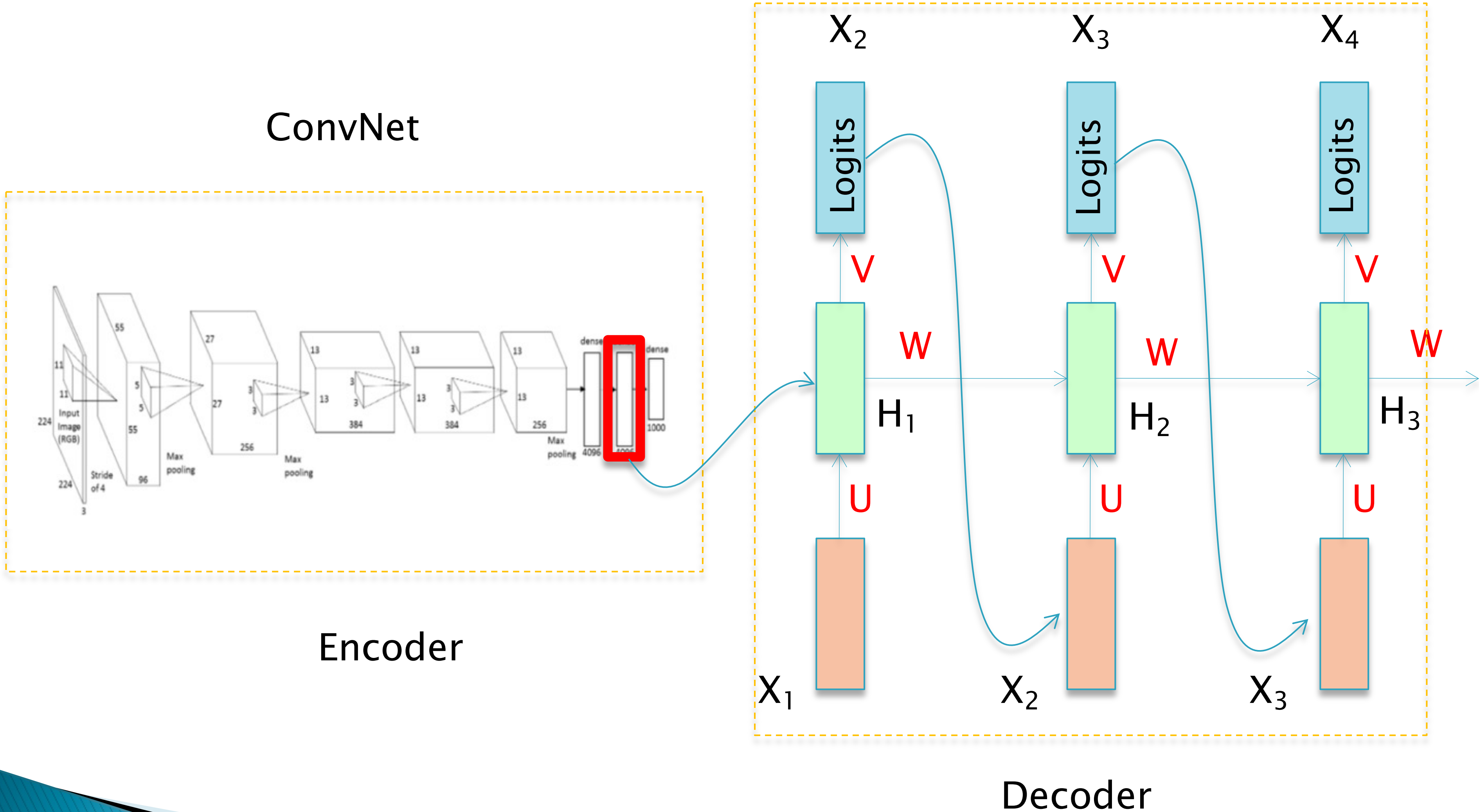


Figure 3: Four sample alignments found by RNNsearch-50. The x-axis and y-axis of each plot correspond to the words in the source sentence (English) and the generated translation (French), respectively. Each pixel shows the weight α_{ij} of the annotation of the j-th source word for the i-th target word (see Eq. (6)), in grayscale (0: black, 1: white). (a) an arbitrary sentence. (b–d) three randomly selected samples among the sentences without any unknown words and of length between 10 and 20 words from the test set.

Image Captioning

Explain Images with Multimodal Recurrent Neural Networks, Mao et al.
Deep Visual-Semantic Alignments for Generating Image Descriptions, Karpathy and Fei-Fei
Show and Tell: A Neural Image Caption Generator, Vinyals et al.
Long-term Recurrent Convolutional Networks for Visual Recognition and Description, Donahue et al.
Learning a Recurrent Visual Representation for Image Caption Generation, Chen and Zitnick

Generating a Context



Results

Image Captioning: Example Results

Captions generated using neuraltalk2
All images are CC0 Public domain: cat suitcase, cat tree, dog, bear, surfers, tennis, giraffe, motorcycle

A cat sitting on a suitcase on the floor

A cat is sitting on a tree branch

A dog is running in the grass with a frisbee

A white teddy bear sitting in the grass

Two people walking on the beach with surfboards

A tennis player in action on the court

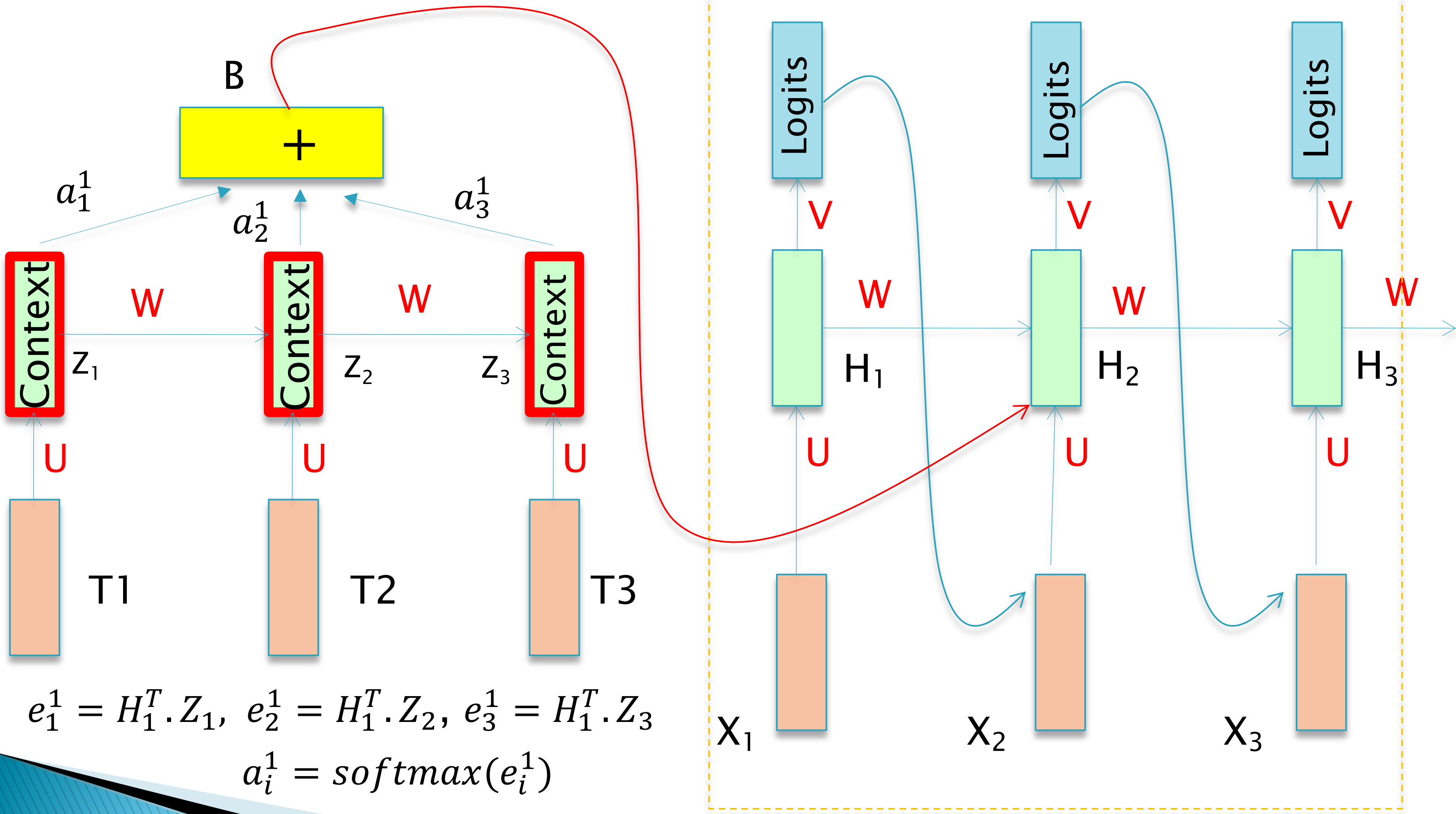
Two giraffes standing in a grassy field

A man riding a dirt bike on a dirt track

Trained using the Microsoft CoCo Dataset
- 330K Images
- 5 captions per image

How Can We Use Attention with Images?

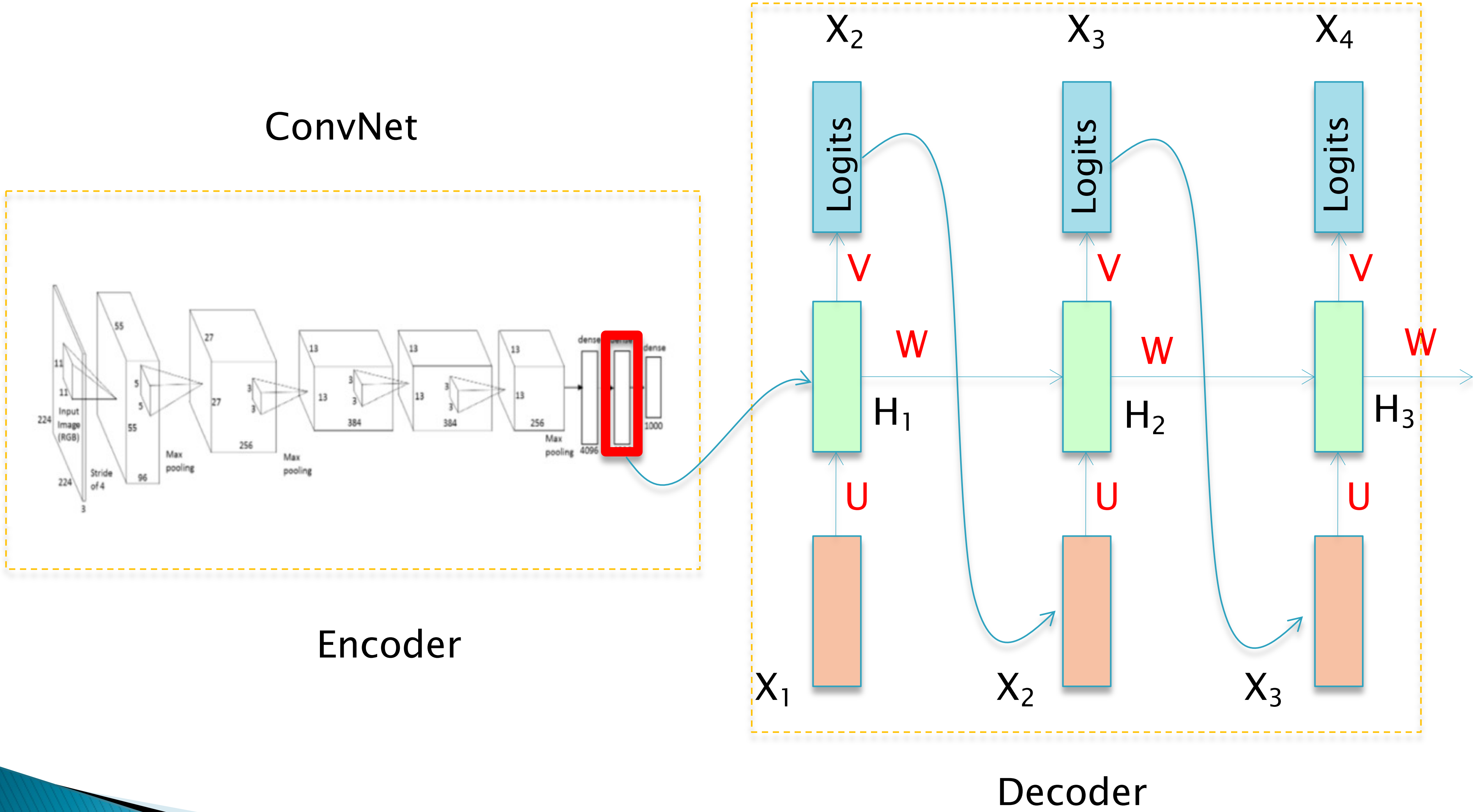
$$B_1 = a_1^1 Z_1 + a_2^1 Z_2 + a_3^1 Z_3$$



$$e_1^1 = H_1^T . Z_1,\ e_2^1 = H_1^T . Z_2,\ e_3^1 = H_1^T . Z_3$$

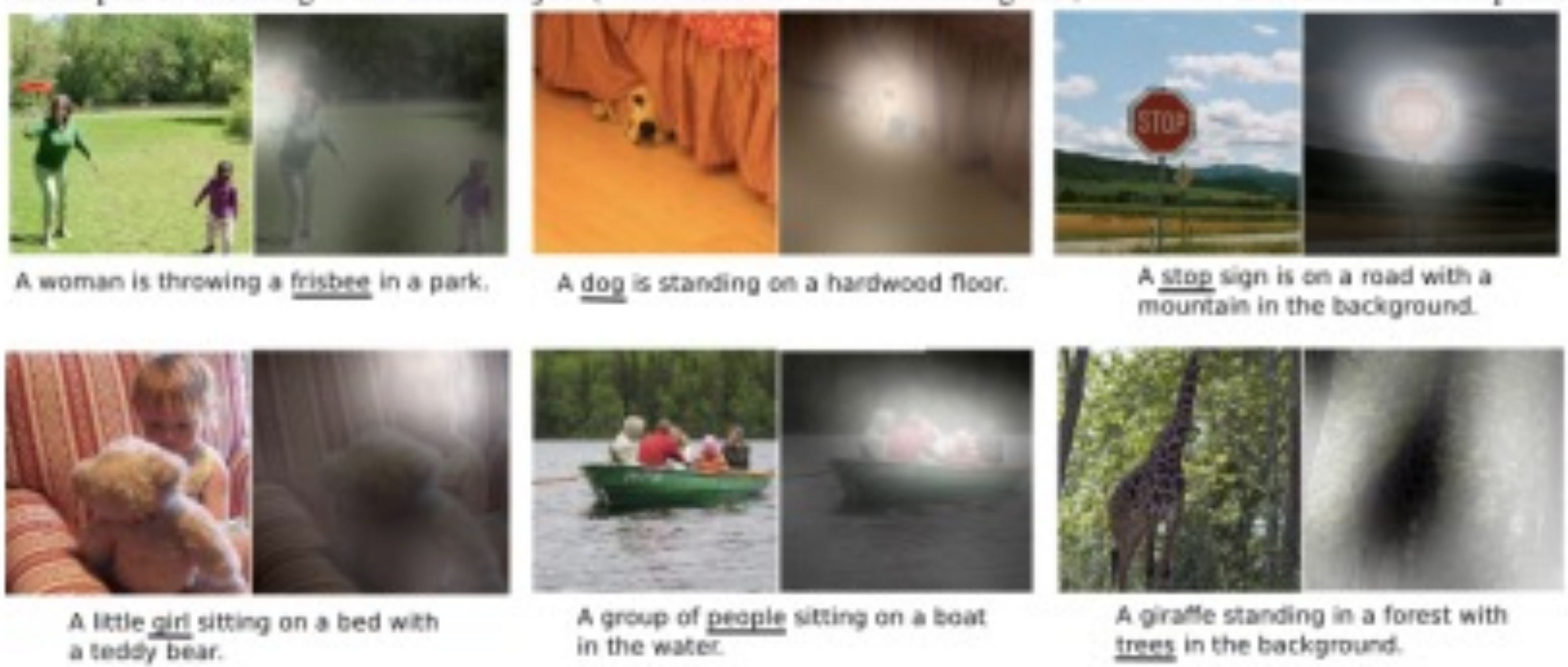
$$a_i^1 = softmax(e_i^1)$$

How Can We Use Attention with Images?

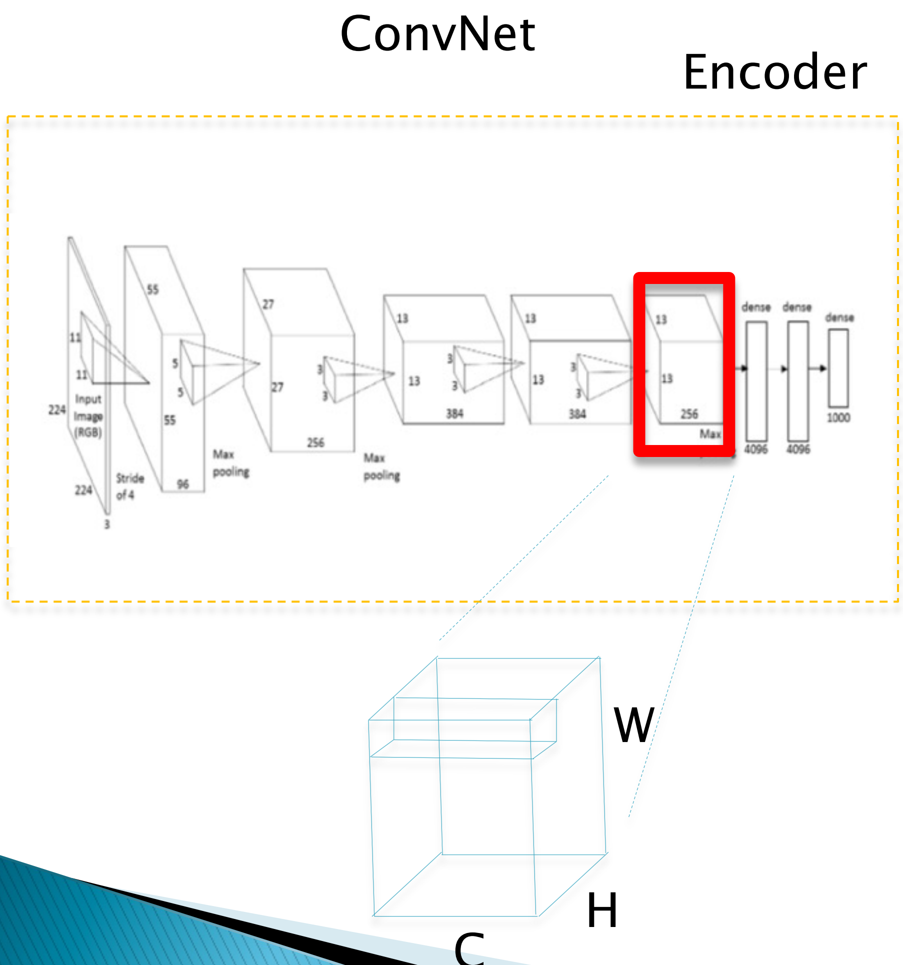


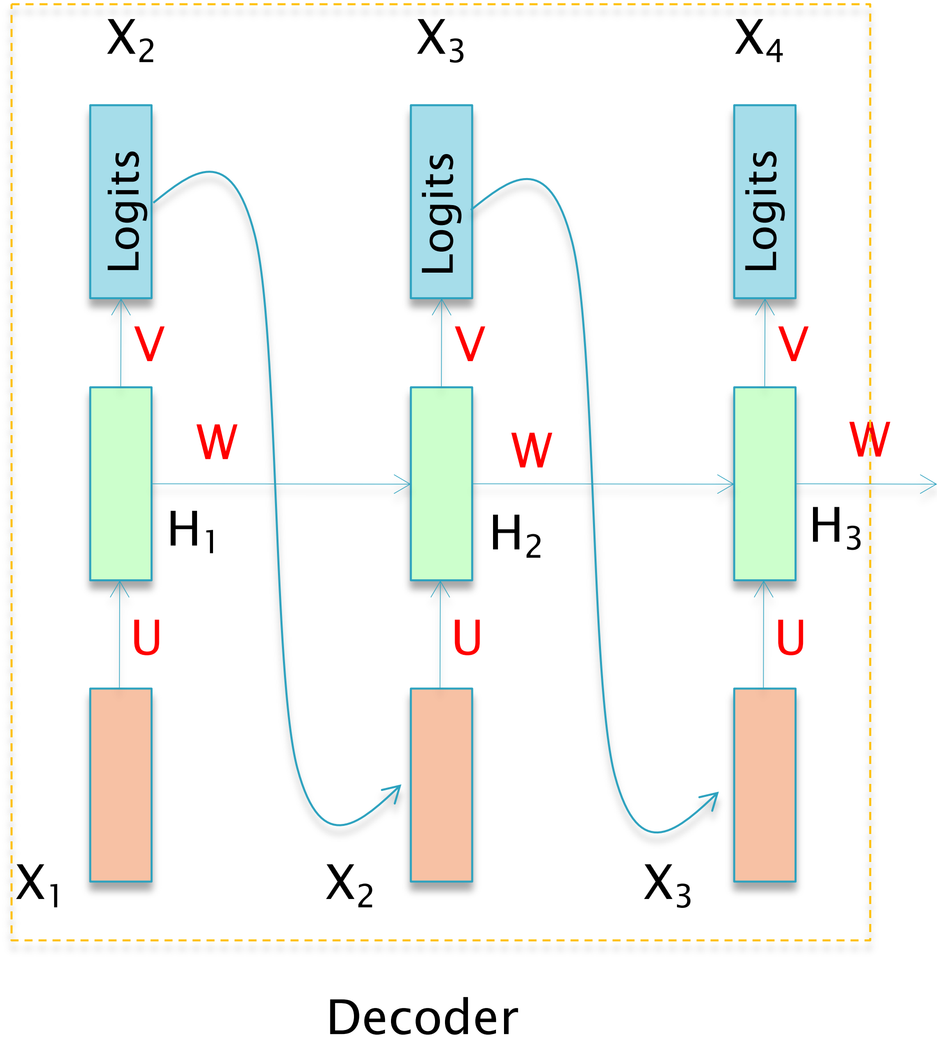
Using Attention with Images

Figure 3. Examples of attending to the correct object (*white* indicates the attended regions, *underlines* indicated the corresponding word)



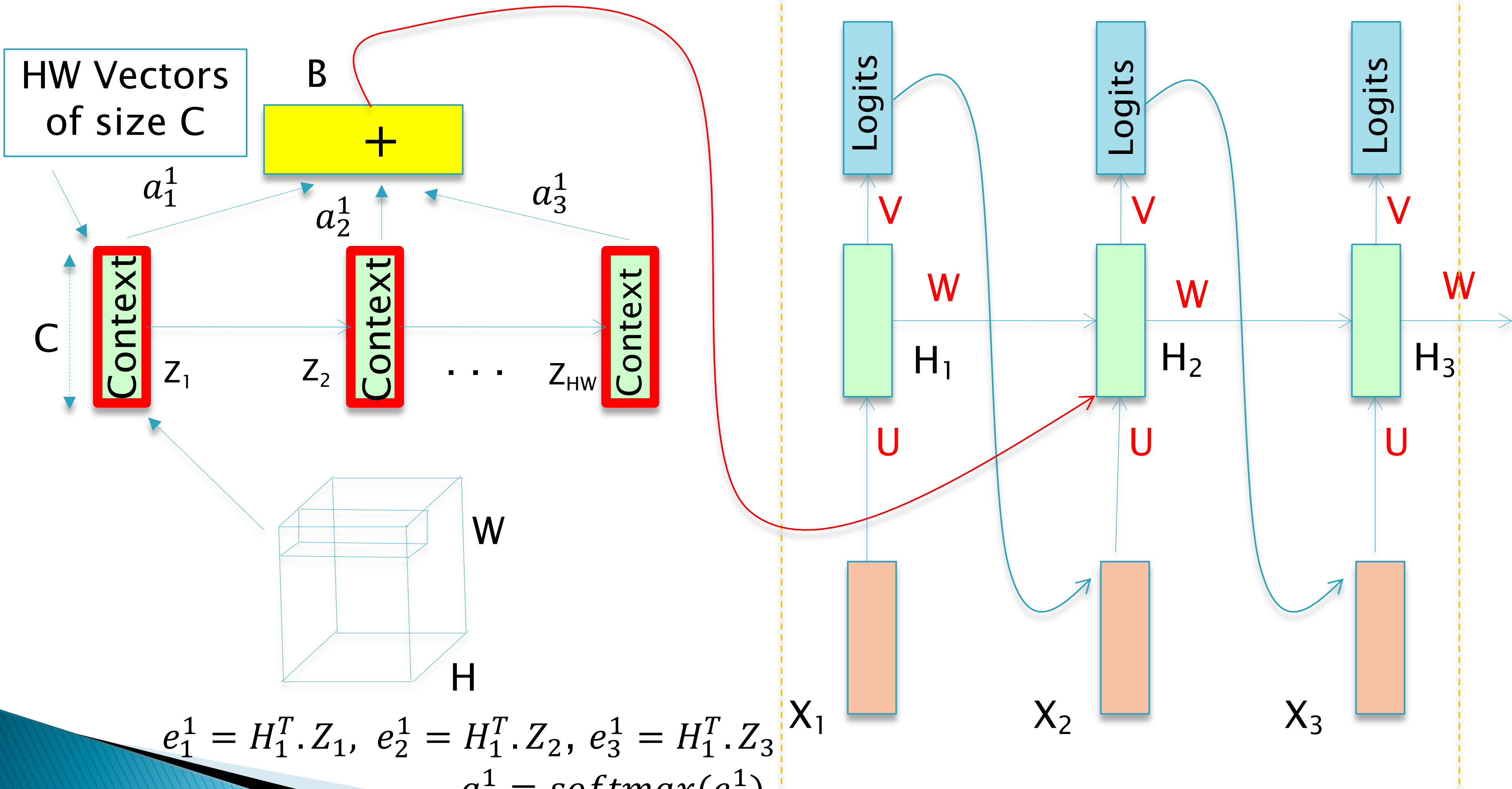
Generating Image Attention Contexts





Generating Image Attention Contexts

$$B_1 = a_1^1 Z_1 + a_2^1 Z_2 + a_3^1 Z_3$$



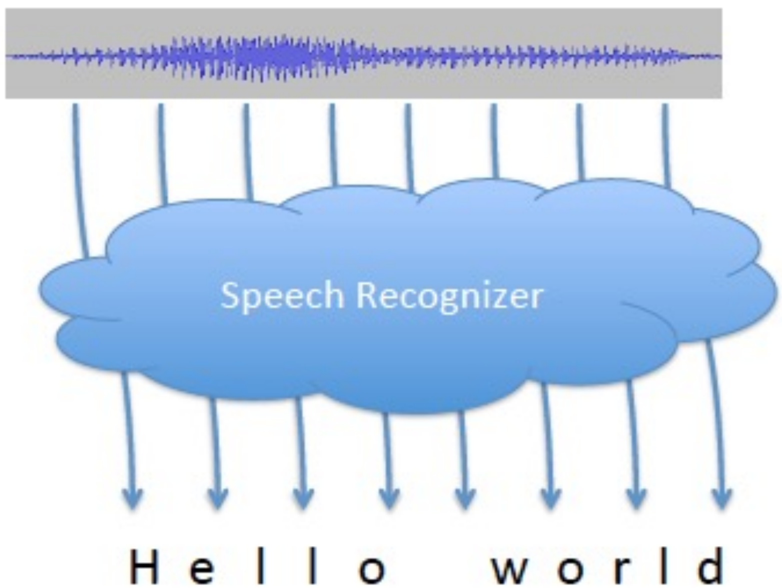
$$e_1^1 = H_1^T . Z_1,\ e_2^1 = H_1^T . Z_2,\ e_3^1 = H_1^T . Z_3$$

$$a_i^1 = softmax(e_i^1)$$

Speech Transcription

The Speech Transcription Problem

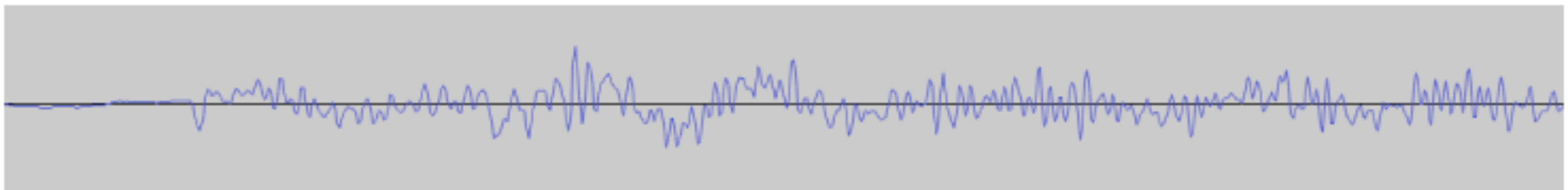
- Given speech audio, generate a transcript.



Important goal of AI: historically hard for machines, easy for people.

Raw Audio

- Simple 1D signal:



Typical sample rates for speech: 8KHz, 16KHz.
Each sample typically 8-bit or 16-bit.

- 1D vector: $X = [x_1 x_2 \ldots]$

Pre–Processing

Two ways to start:

– Minimally pre-process (e.g., simple spectrogram).

– Train model from raw audio wave.

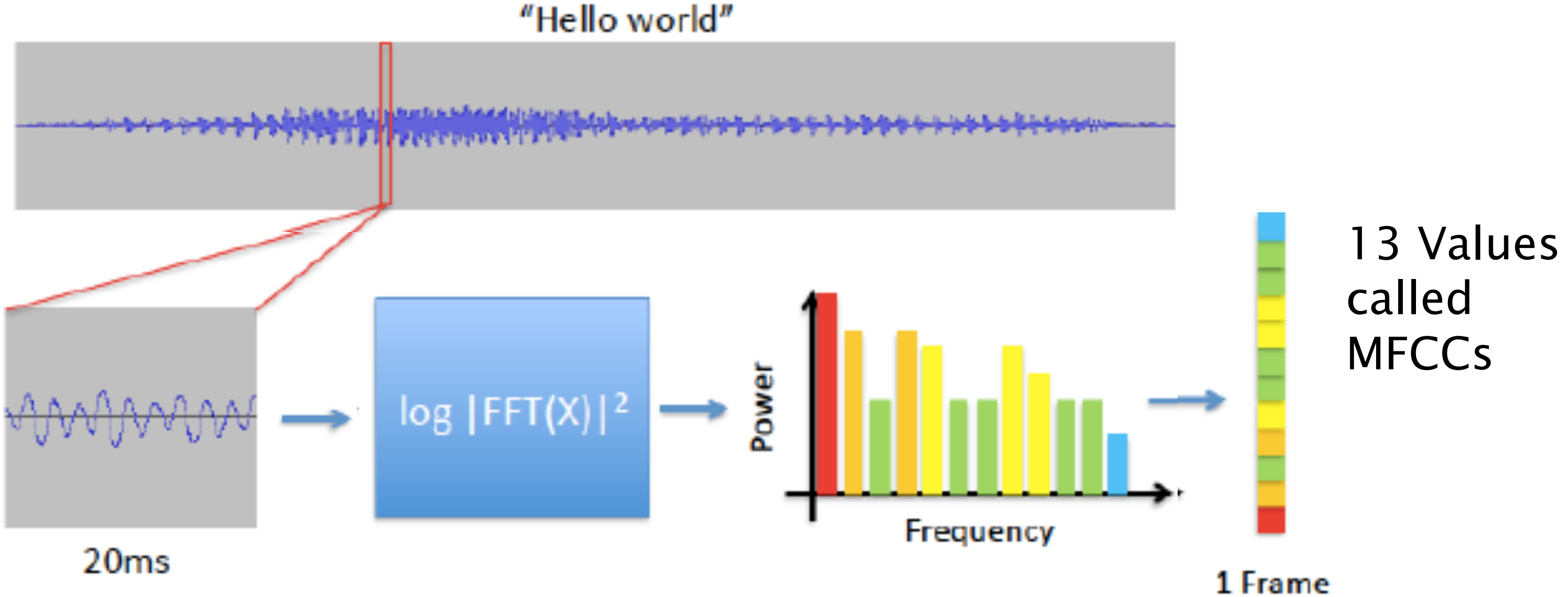
- It works!

See, e.g., Sainath et al., Interspeech 2015

Speech Spectrogram

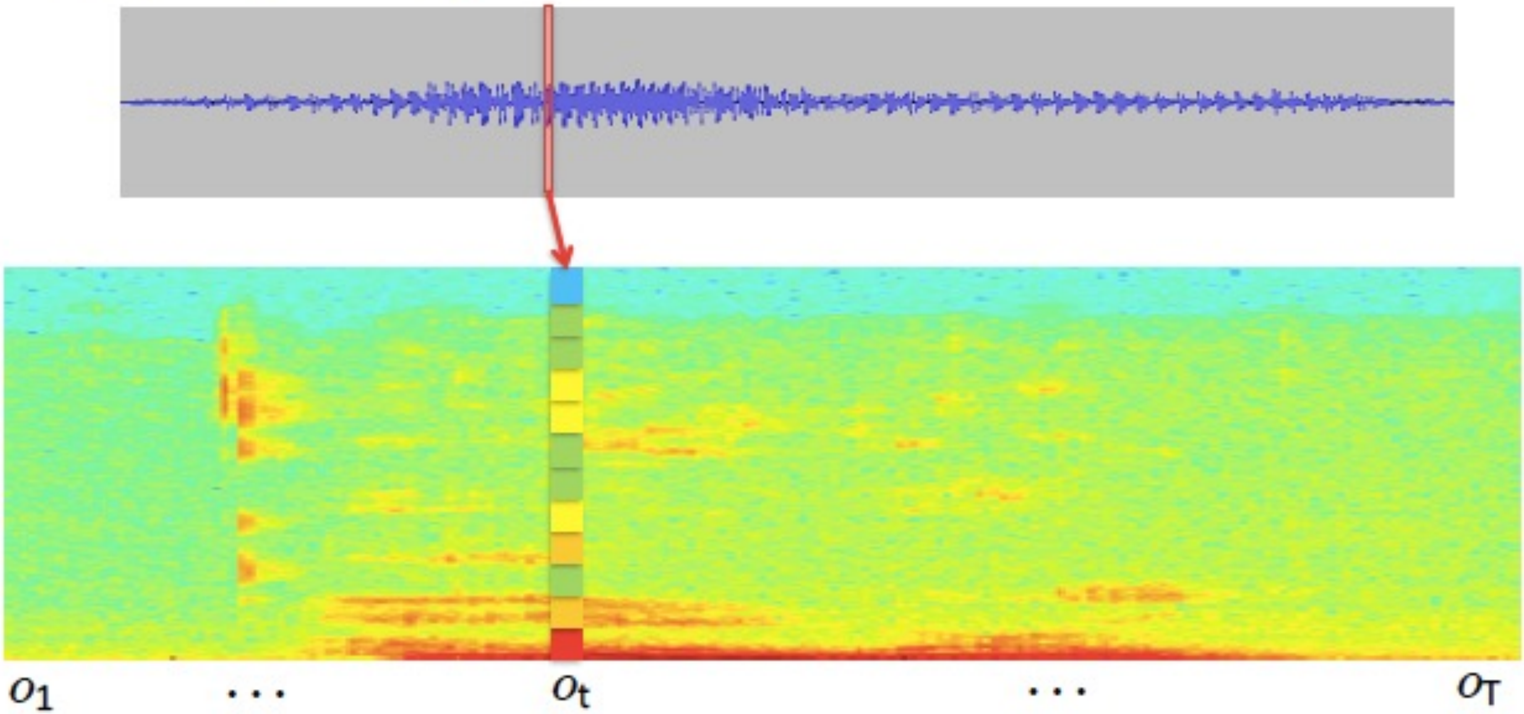
Take a small window (e.g., 20ms) of waveform.

– Compute FFT and take magnitude. (i.e., power)
– Describes frequency content in local window.

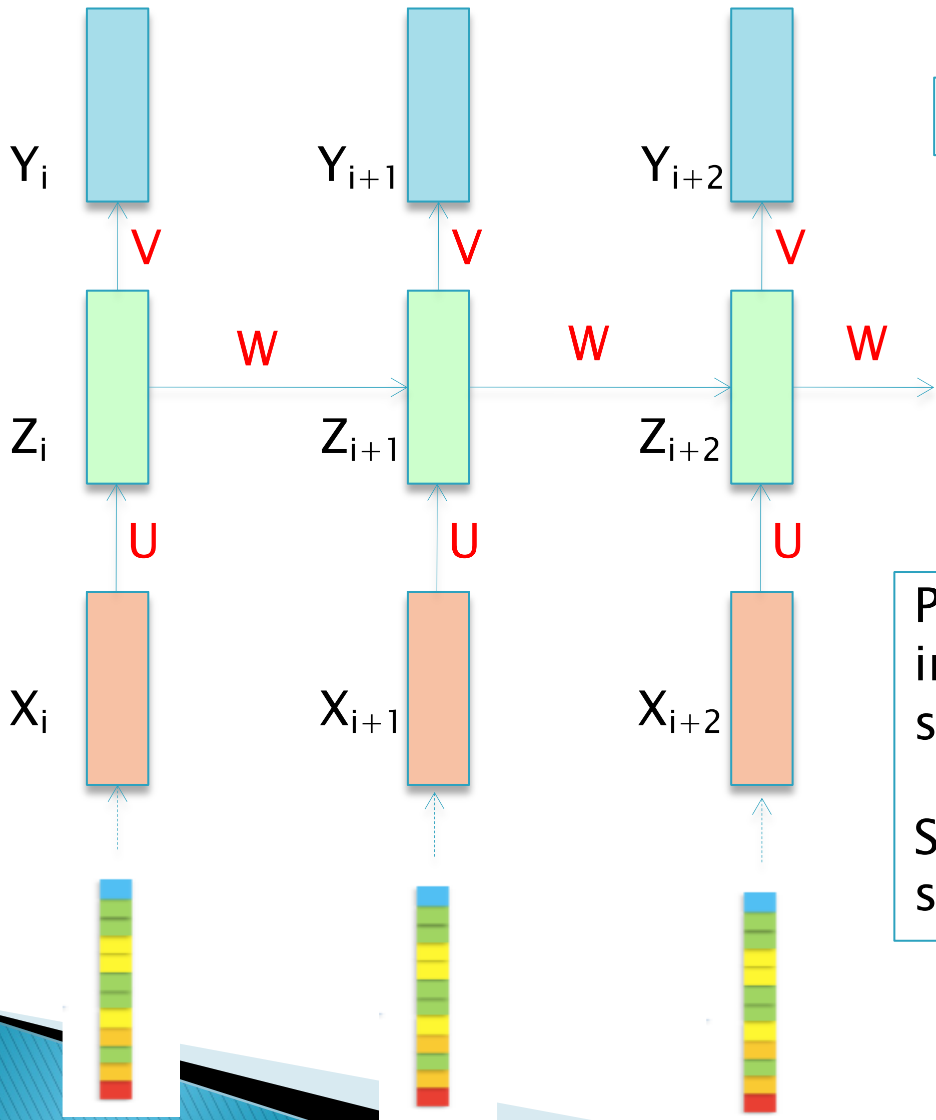


Speech Spectrogram

- Concatenate frames from adjacent windows to form "spectrogram".



The Speech Transcription Problem

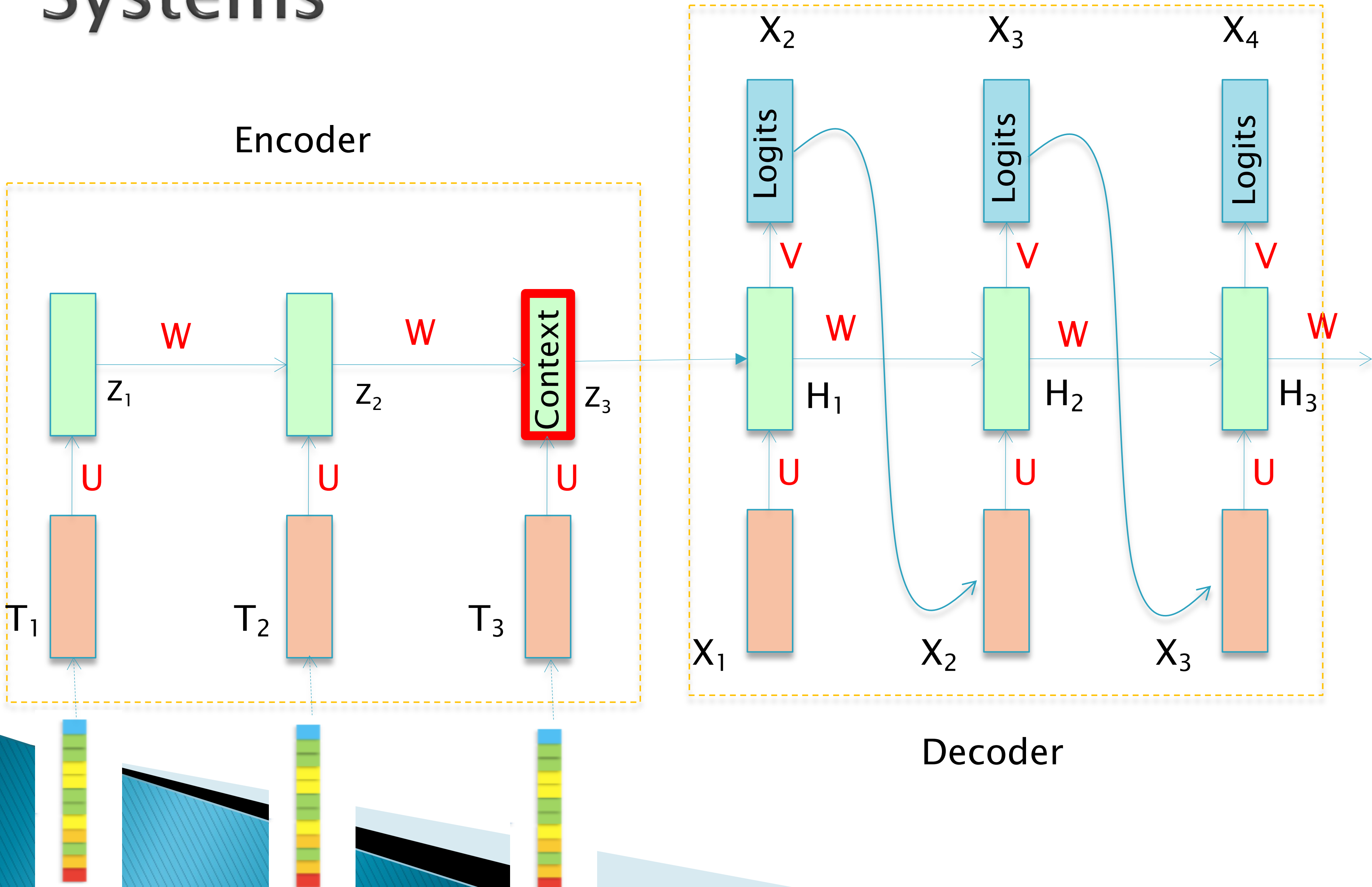


Output = Words

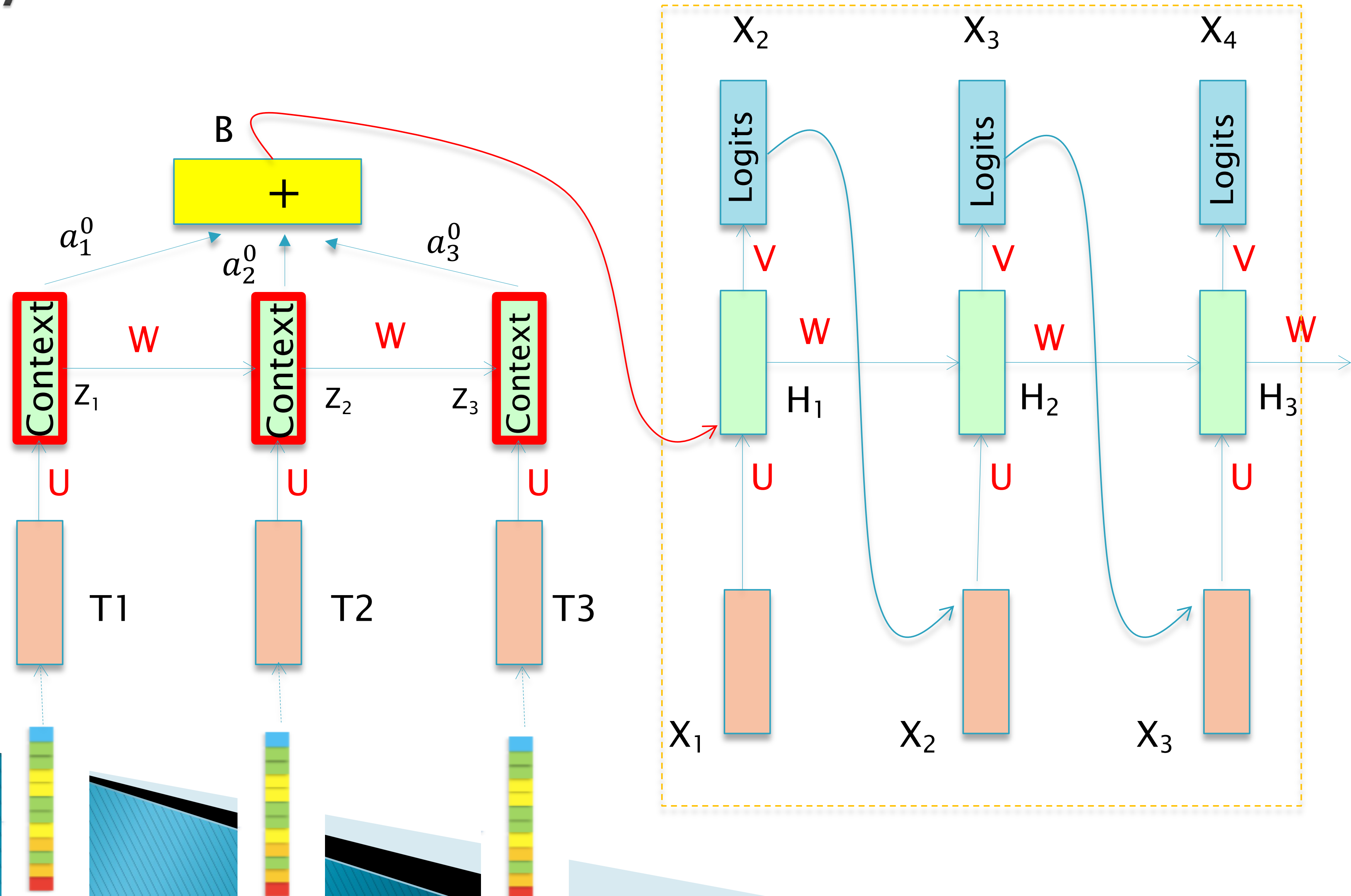
Problem: Difference in the input and output sequence sizes

Solution: Use Encoder Decoder systems

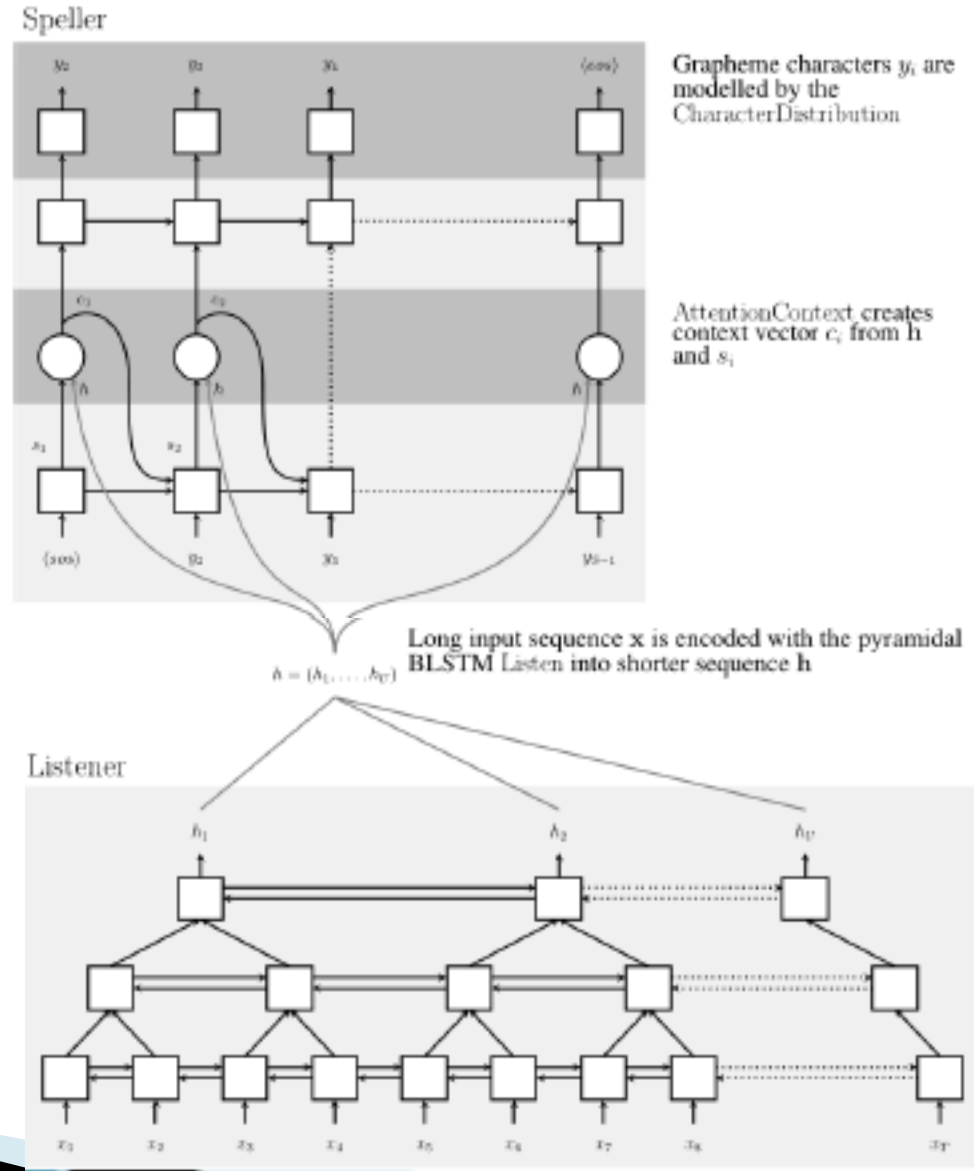
Speech Transcription with Enc-Dec Systems



Speech Transcription with Enc-Dec Systems with Attention



The "Listen, Attend and Spell" System



Pyramidal
Bi-Directional LSTM

- 4 Layers, reduced time resolution by 8 times
- 512 LSTM nodes/layer

https://arxiv.org/abs/1508.01211

Further Reading

- Das and Varma: ChapterNLP
- Chollet: Chapter 11, Section 11.5
  Chapter 12, Section 12.1