NLP Part 1

Lecture 15

Subir Varma

Tasks in NLP

- Building Word Embeddings
- Language Modeling
- Text Categorization
- Generating Text about a Topic
- Language Translation
- Question Answering
- Image Captioning
- Speech Transcription
- Generating Text Summaries

Problem being Solved

How to Find Representations for Words

How to Find Representations for Sentences/Paragraphs

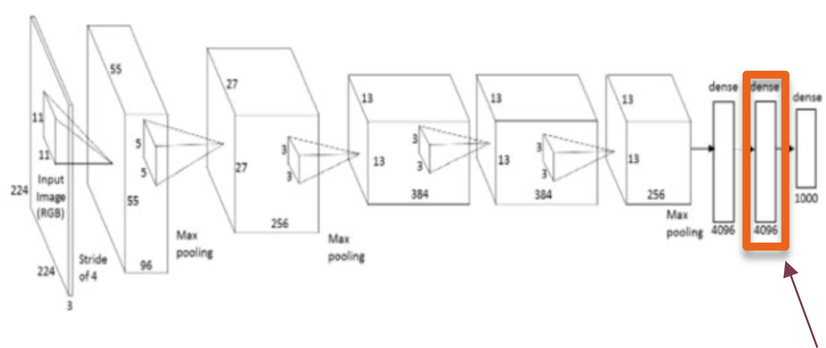


Image Representation

Word Embeddings

1-Hot Encoding

$$
\begin{array}{l c}
 & \begin{array}{ccccc} 1 & 2 & 3 & \cdots & 10000 \end{array} \\
\begin{array}{l} \text{word 1} \\ 2 \\ 3 \\ \vdots \\ 100 \end{array} & \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ \vdots & & & & \\ & & & & \\ & & & & \end{pmatrix}
\end{array}
$$

<u>Review 1</u>

$$
\begin{array}{l c}
 & \begin{array}{ccccc} 1 & 2 & 3 & \cdots & 10000 \end{array} \\
\begin{array}{l} \text{word 1} \\ 2 \\ 3 \\ \vdots \\ 100 \end{array} & \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 1 \\ \vdots & & & & \\ & & & & \\ & & & & \end{pmatrix}
\end{array}
$$

<u>Review 2</u>

- Results in very high dimensional representations
- Does not capture relationship between words

Richer Representations

We want **richer representations** expressing **semantic similarity**.

Distributional semantics:
"You shall know a word by the company it keeps." — J.R. Firth (1957)

Idea: produce **dense** vector representations based on the **context/use** of words.

Word Embeddings

	bite	cute	furry	loud
kitten	0	1	0	0
cat	0	1	1	0
dog	1	0	1	1

Use inner product or cosine as **similarity kernel**. E.g.:

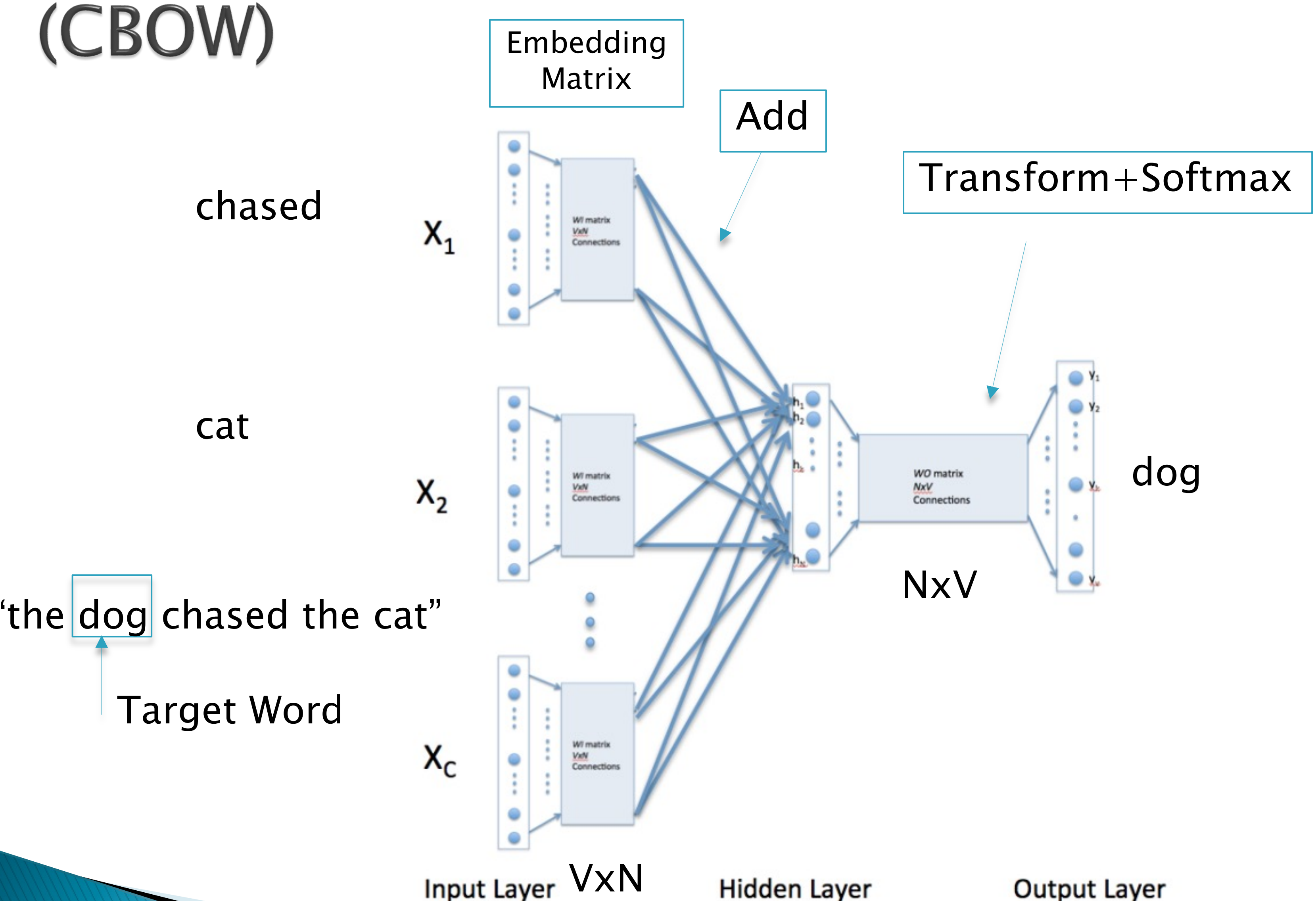
$$sim(\text{kitten}, \text{cat}) = cosine(\textbf{kitten}, \textbf{cat}) \approx 0.58$$

$$sim(\text{kitten}, \text{dog}) = cosine(\textbf{kitten}, \textbf{dog}) = 0.00$$

$$sim(\text{cat}, \text{dog}) = cosine(\textbf{cat}, \textbf{dog}) \approx 0.29$$

Reminder: $cosine(\mathbf{u}, \mathbf{v}) = \dfrac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \times \|\mathbf{v}\|}$

Word2Vec: Continuous Bag of Words (CBOW)



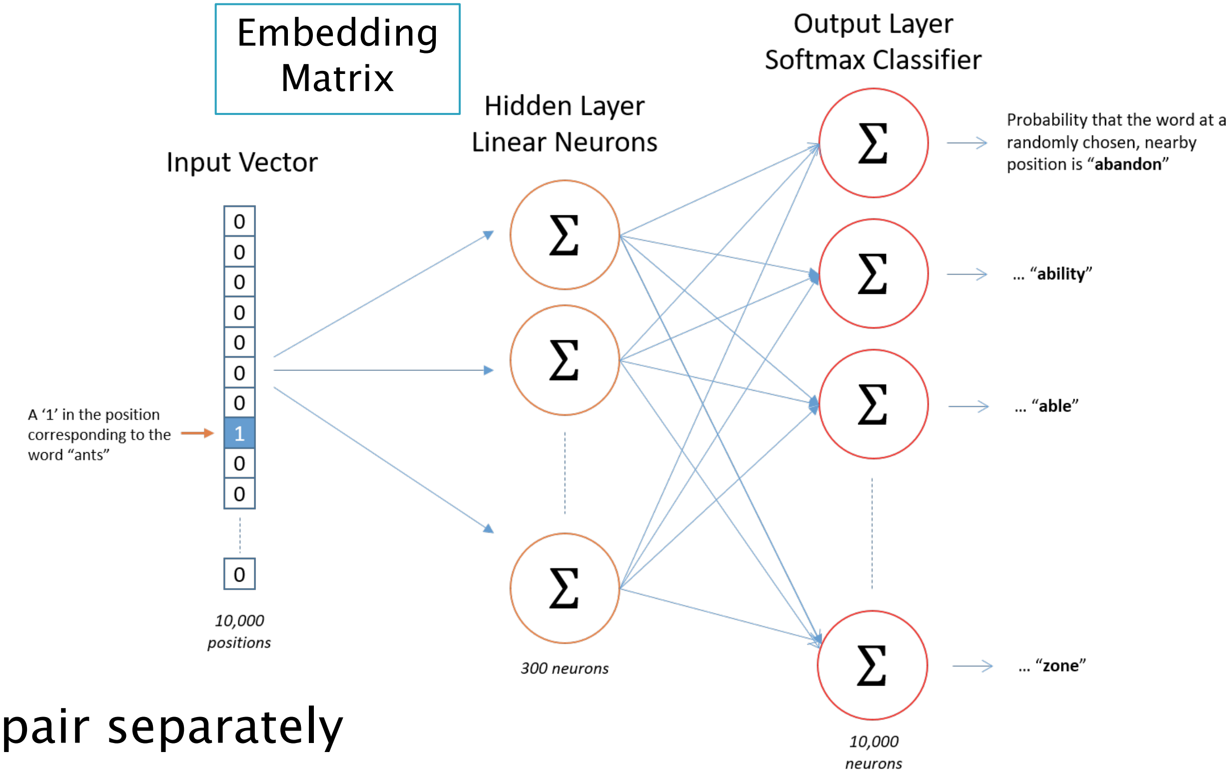
V: Number of words in corpus
N: Size of Embedding Vector

Word2Vec: Skip-Gram

Transform+Softmax

Embedding Matrix

dog



cat

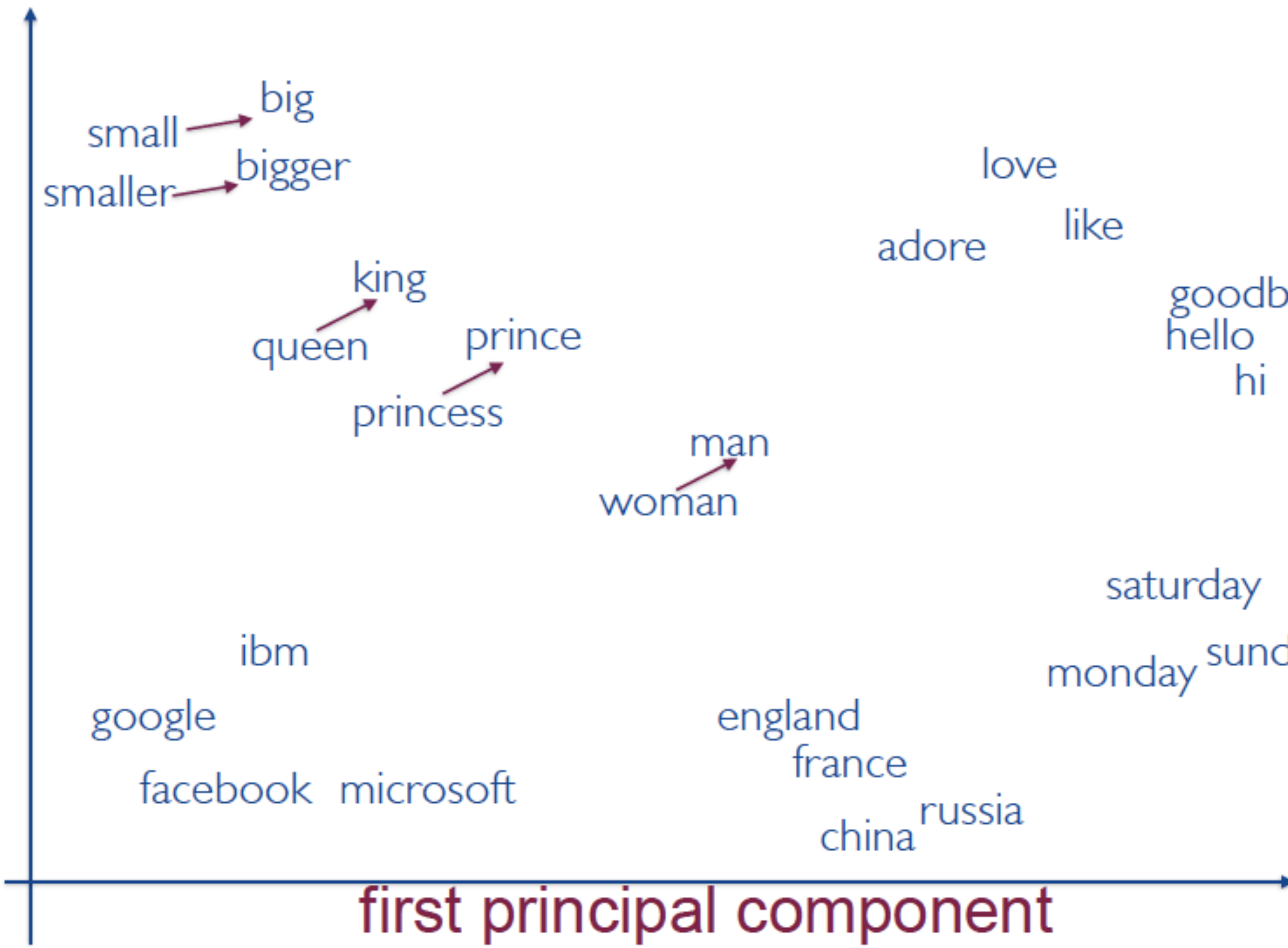
chased

Train each pair separately
(dog, cat)
(dog, chased)

"the $\boxed{\text{dog}}$ chased the cat"

Target Word

Word Vectors

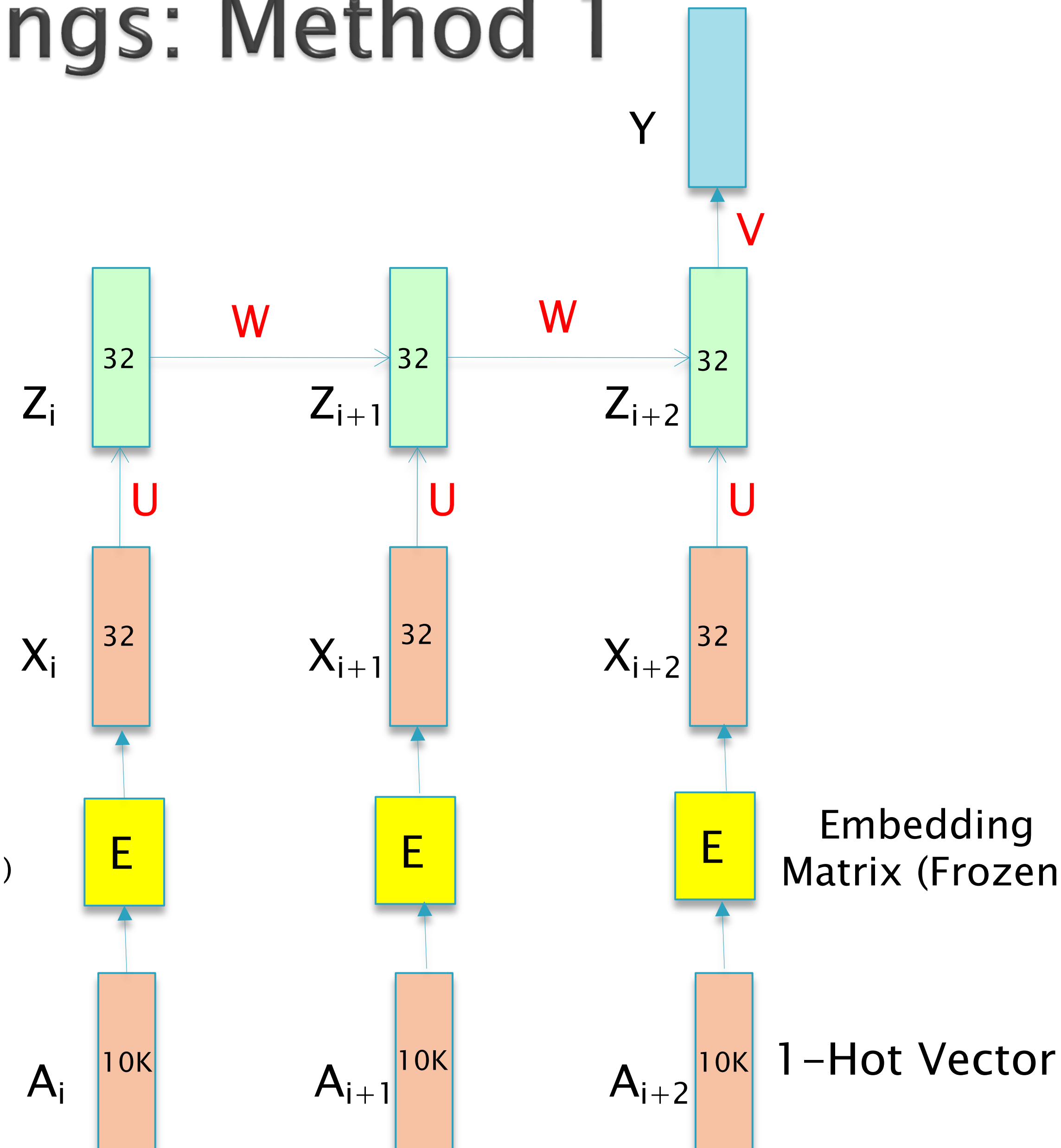


king – queen = man – woman

Using Embeddings: Method 1

Use a Frozen Embedding Matrix
- Glove
- Word2Vec

$A^{(1*10K)}E^{(10K*32)}=X^{(1*32)}$

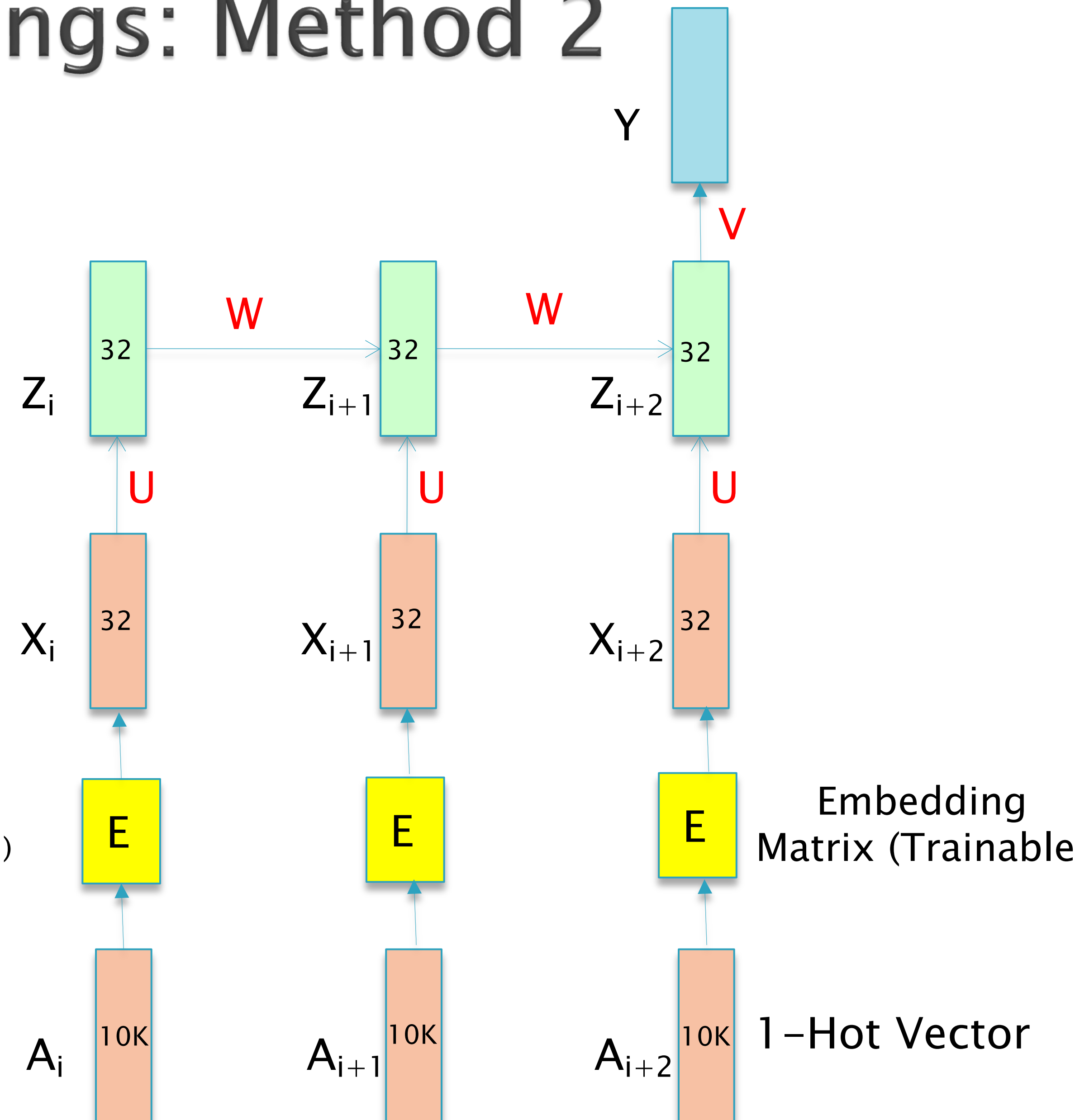


Chollet: Page 334

Using Embeddings: Method 2

Use a Trainable Embedding Matrix

$A^{(1*10K)}E^{(10K*32)}=X^{(1*32)}$



Chollet: Page 331

Learning Task Based Embeddings

- Embedding matrix can be learnt from scratch or initialized with pre-learned embeddings
- Using pre-learned embeddings (Word2Vec or Glove) is a type of Transfer Learning
- If enough training data available, then the embeddings can be computed during the training process (using backprop)
  - These capture embeddings that are relevant to the task

Text Classification

Applications of Text Classification

<u>K-ary Classification</u>

- Is this email spam?
- Positive or Negative Review?
- What is the topic of this article?
- What language is this article in?
- Who is the Author of this article?

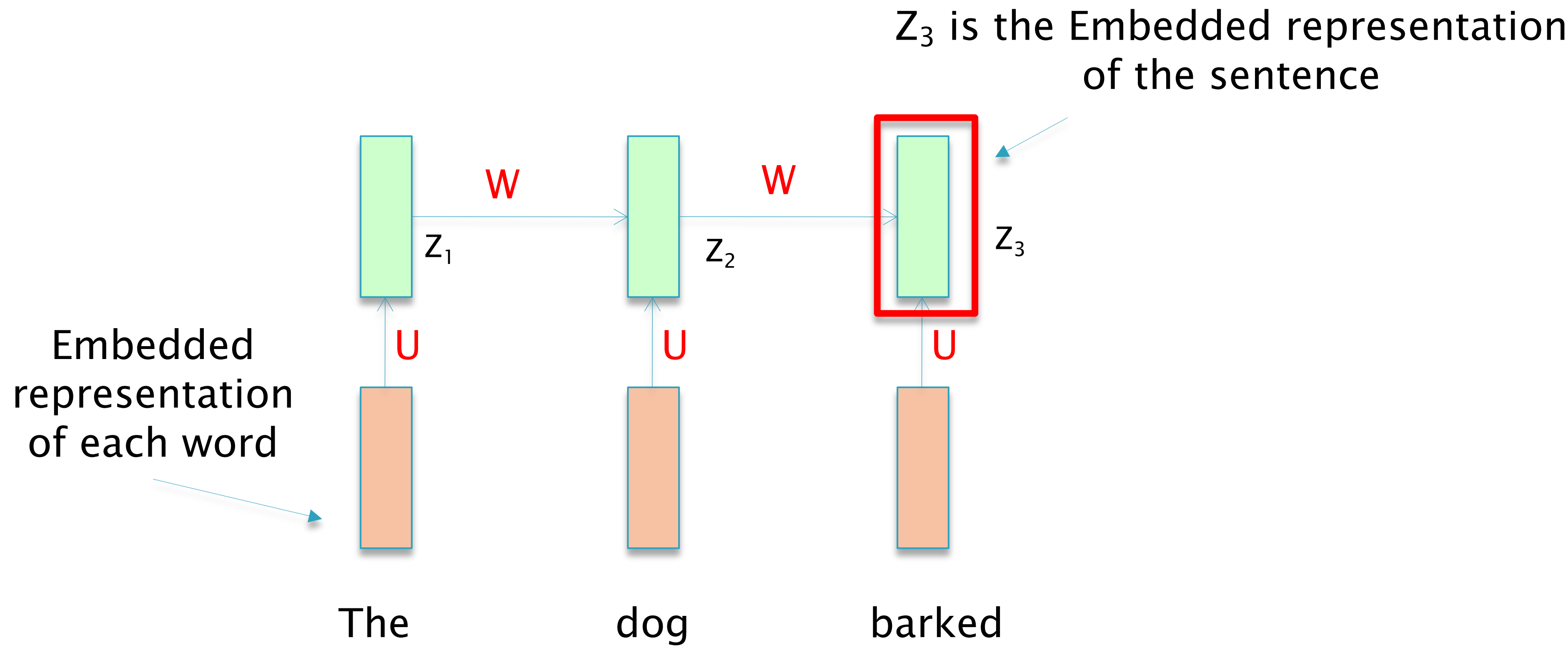
<u>Regression</u>

- What is the age/gender etc of the author

<u>Multi-Label Classification</u>

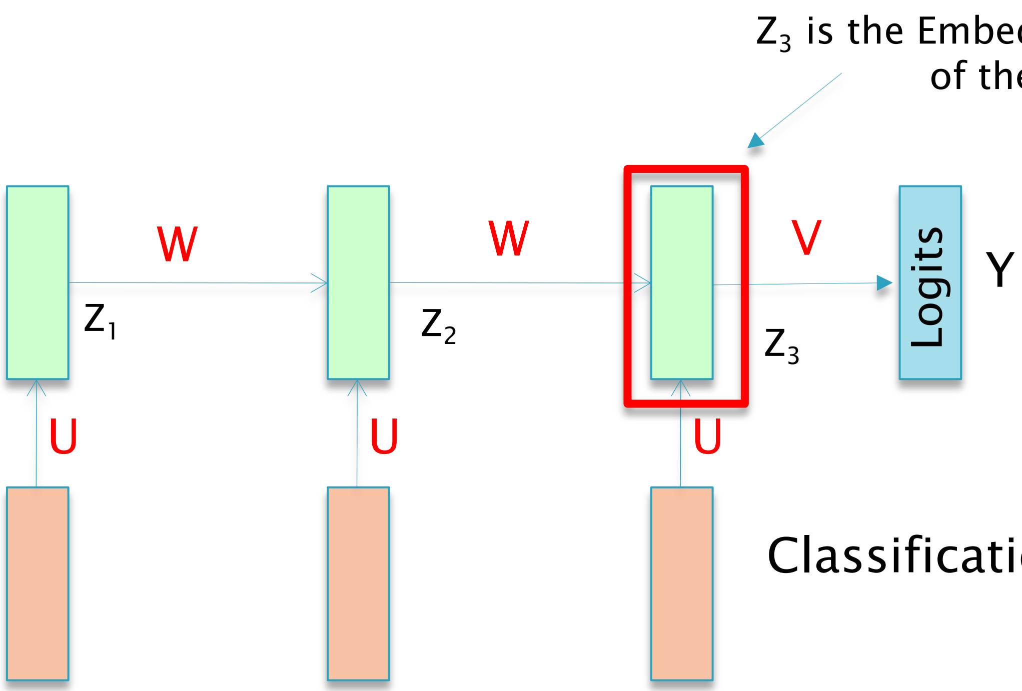
- Predict hashtags for a tweet

Representing Text Using RNNs



Z_3 contains information about all the text in the sequence

Classification/Regression



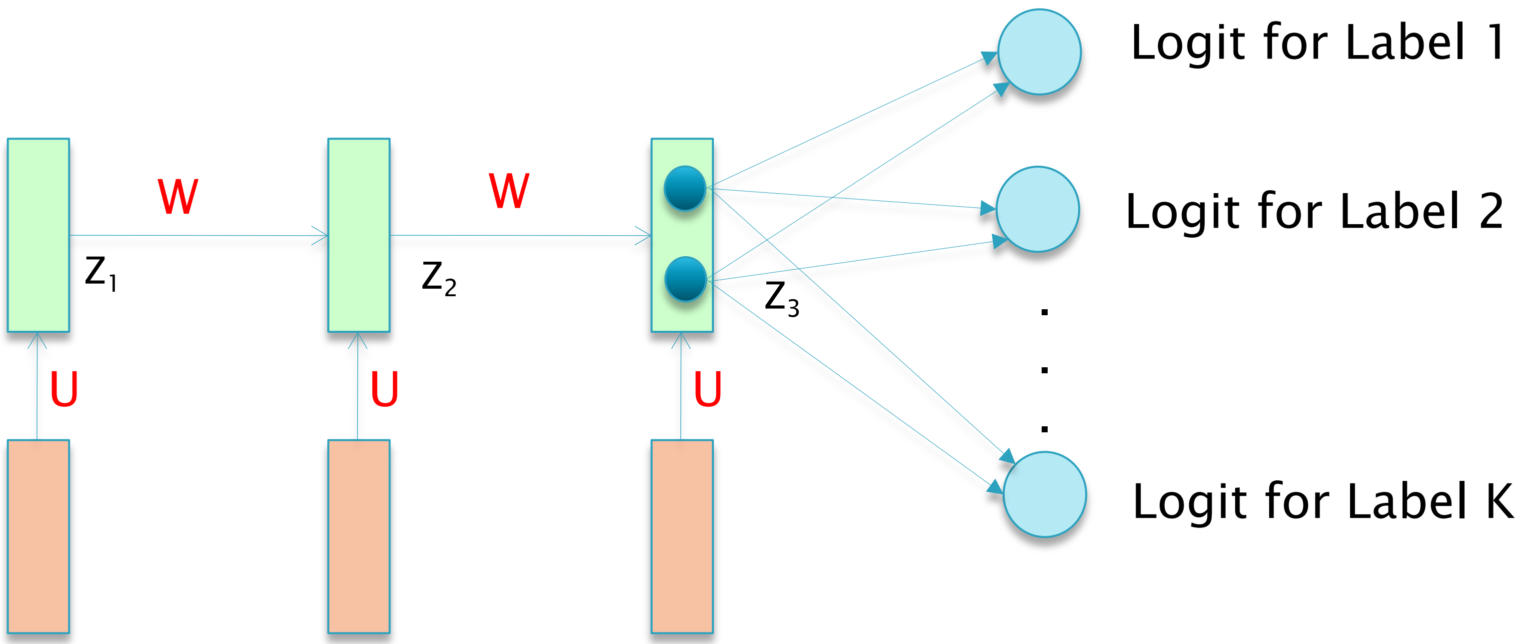
$$Y = \text{softmax}(A_4)$$

Classification: $$\mathcal{L} = -\sum_{k=1}^{K} t_k \log y_k$$

Regression: $$\mathcal{L} = \frac{1}{2}\sum_{k=1}^{K} (t_k - a_k)^2$$

Multi-Label Classification

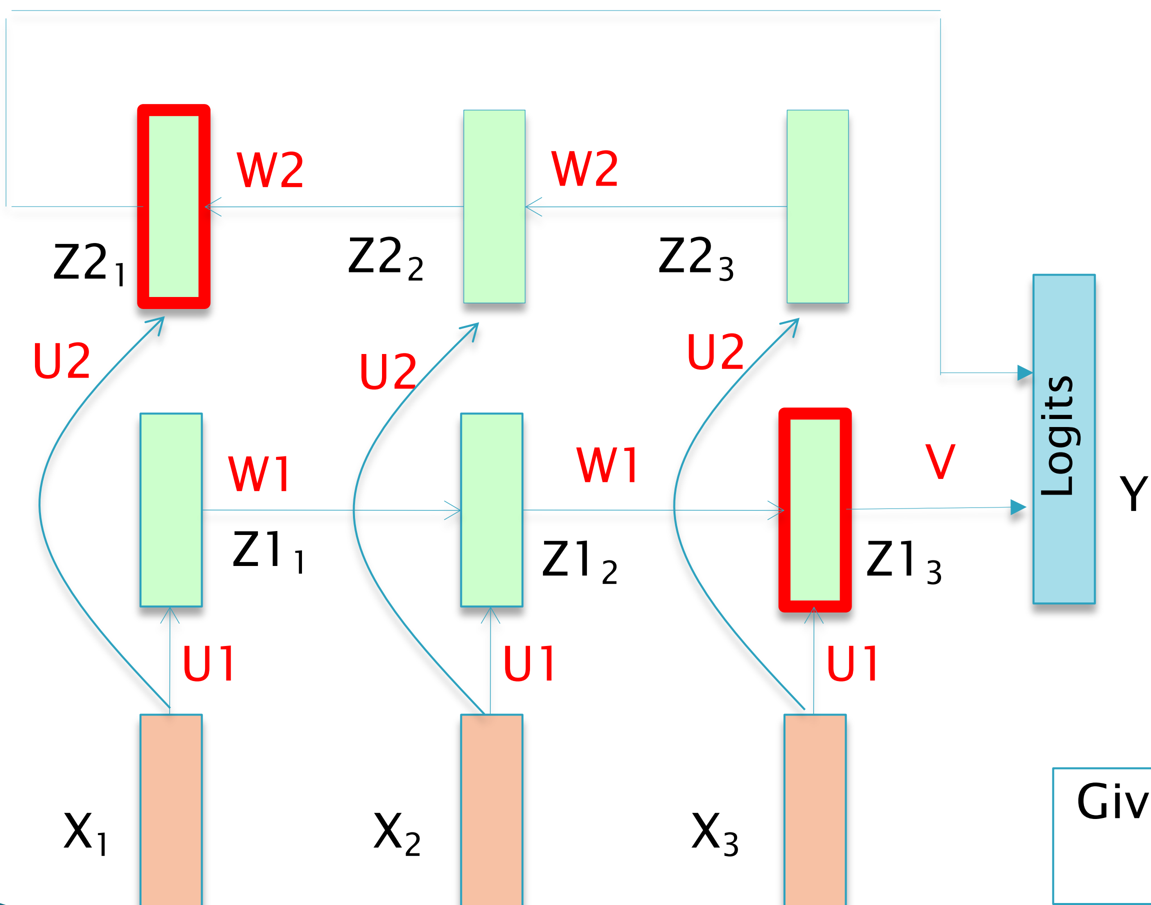
$$\mathcal{L} = -\sum_{k=1}^{K}[t_k \log y_k + (1 - t_k)\log(1 - y_k)]$$

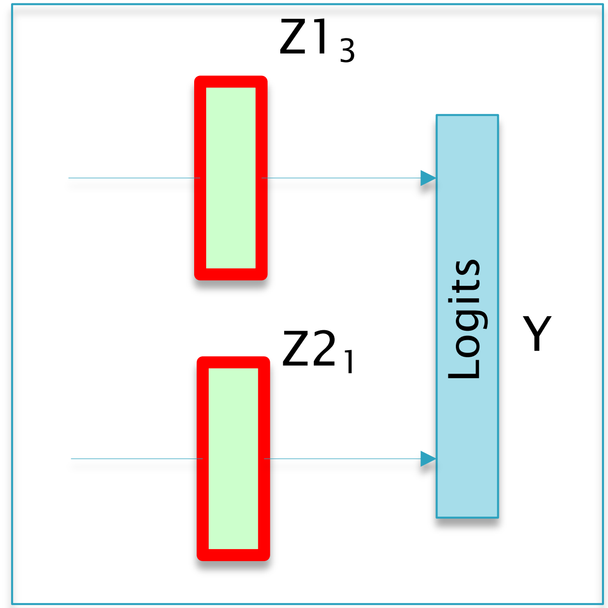


A single sequence has multiple correct labels

Problem reduced to K separate Yes/No decisions
With K Binary Classifiers operating in Parallel

Text Representation with Bi-Directional RNNs



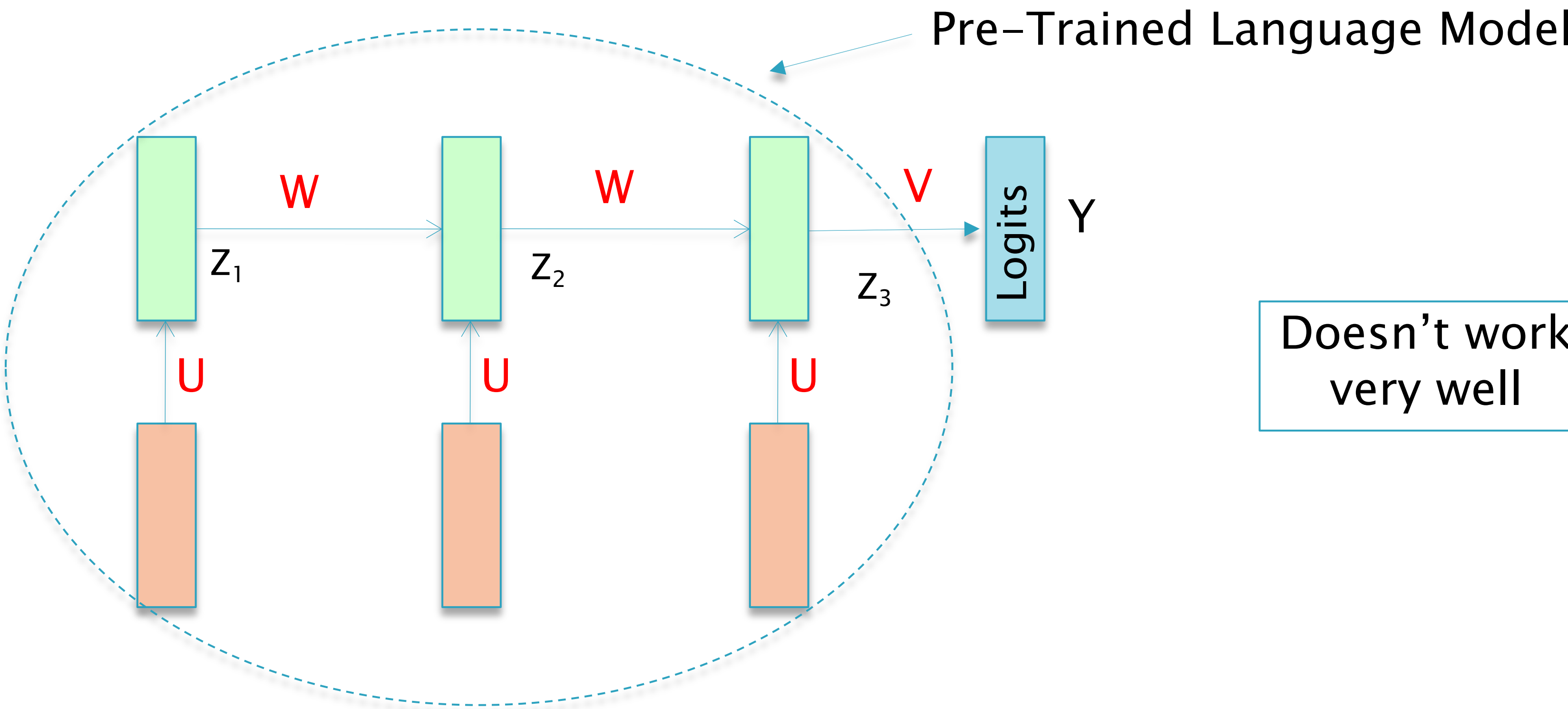


Final State
$Z1_3 \,||\, Z2_1$

Gives an extra 2-3% increase in accuracy

In Keras: model.add(layers.Bidirectional(layers.LSTM(32))

Using Pre-Trained Language Models: Transfer Learning



Doesn’t work very well

W, U are Frozen (with pre-trained weights)
Only V needs to be trained

Benefits:
- Can potentially classify sentences with words not in the classifier training dataset
- Smaller training dataset needed

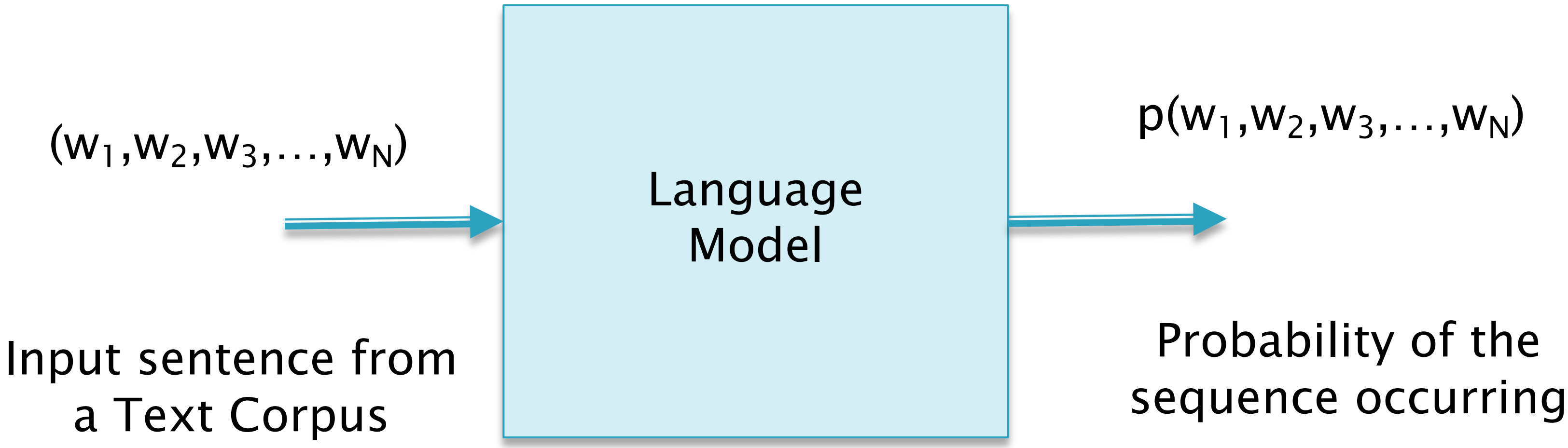
Language Models

What is a Language Model?

- Definition 1: Given a sequence of words $(w_1,\ldots,w_N)$, a Language Model predicts the most probable next word w_{N+1} in the sequence.
- Definition 2: Given a sequence of words $(w_1,\ldots,w_N)$, a Language Model can be used to compute the probability $p(w_1,\ldots,w_N)$ of that sequence occurring in the language

Language Models: Definition 1

Language Models: Definition 2



A language model assigns a probability to a sequence of words, such that $\sum_{w \in \Sigma^*} p(w) = 1$:

Given the observed training text, how probable is this new utterance?

Why are Language Models Useful?

(1) we can compare different orderings of words (e.g. Translation):

$$p(\text{he likes apples}) > p(\text{apples likes he})$$

Syntactically less probable

(2) or choice of words (e.g. Speech Recognition):

$$p(\text{he likes apples}) > p(\text{he licks apples})$$

Syntactically correct but Semantically less probable

Language Models can also be used to Generate new text!

How are Language Models Used?

Much of NLP can be structured as Conditional Language Modeling:

Translation:

p_{LM}(*Les chiens aiment les os* | Dogs love bones)

The translation is the sentence that has the maximum probability in Language 2, given the sentence in Language 1

Question Answering:

p_{LM}(Answer | Document, Question)

The Answer is the word (or words) with the maximum probability of occurring given the Question and a reference Document

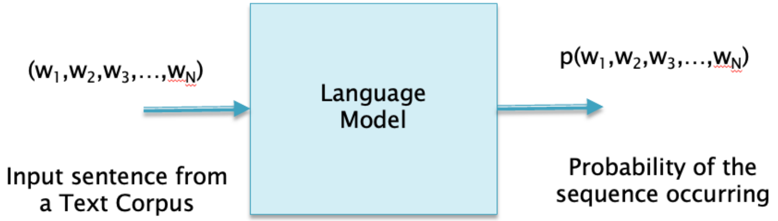
Computing the Probability

- Using the Chain Rule of Probabilities

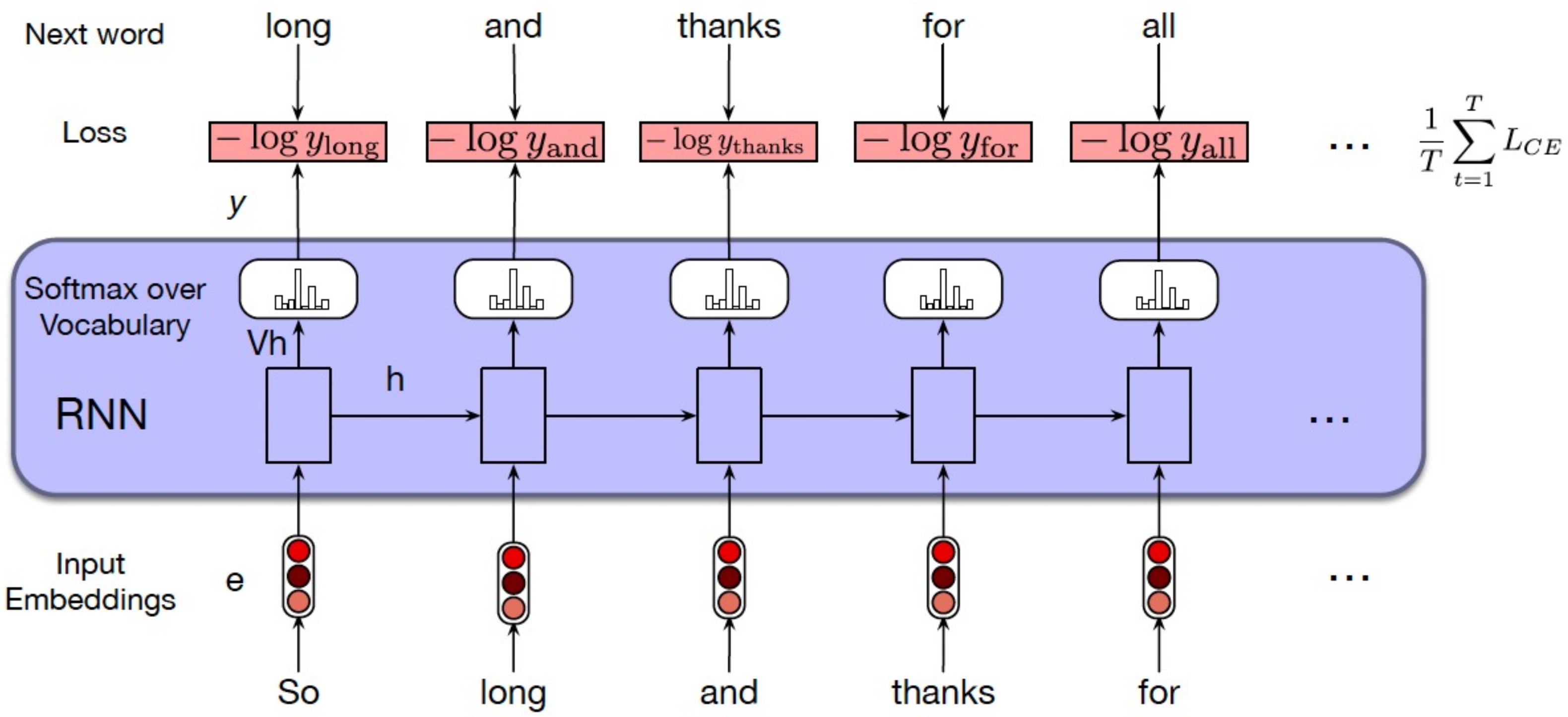
$$P(w_1, w_2, \ldots, w_n) = P(w_1)P(w_2 | w_1)P(w_3 | w_2, w_1), \ldots, P(w_n | w_{n-1}, \ldots, w_1)$$

Language Modeling reduces to the problem of computing these conditional probabilities

The Conditional Probabilities can be computed with a RNN/LSTM



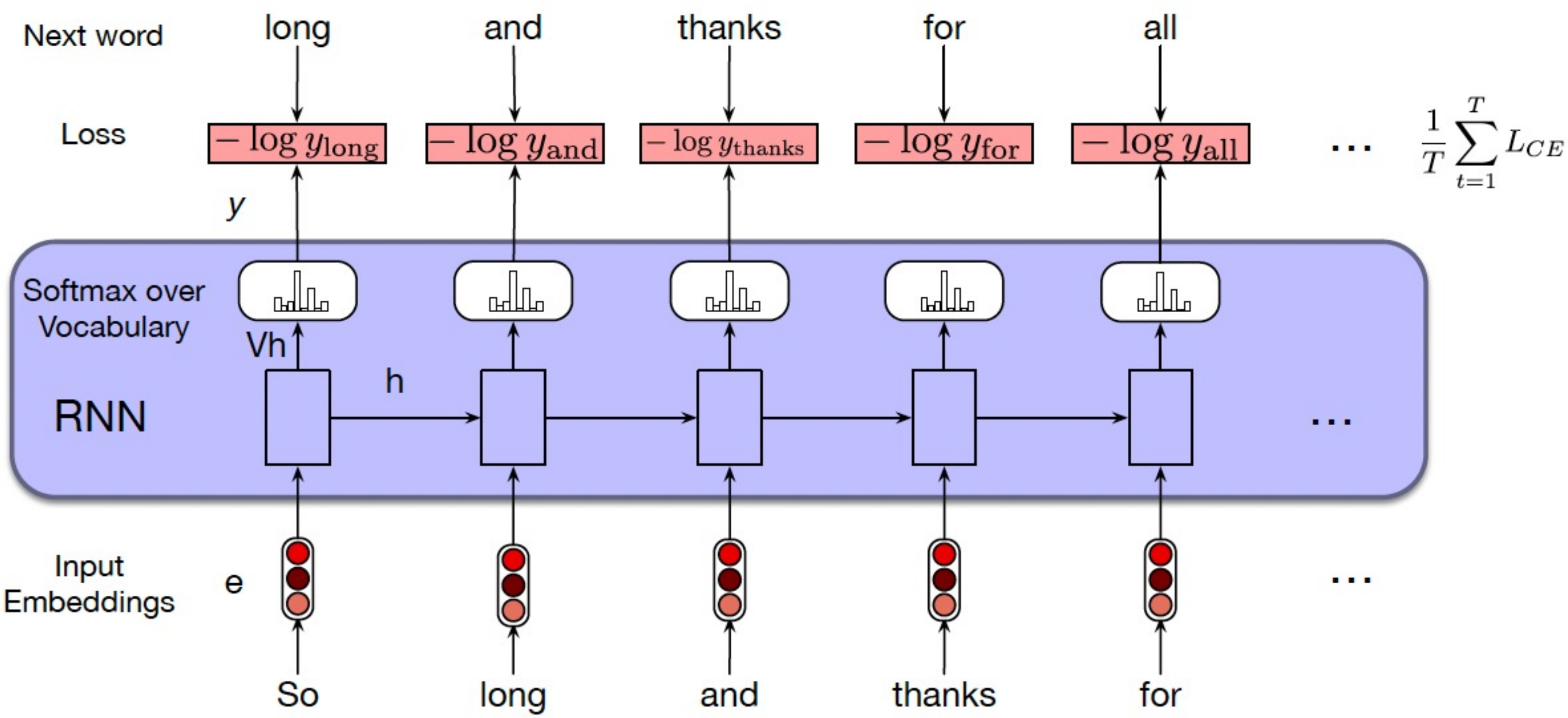
Training the Language Model



Training the RNN
by trying to predict next word

Training the Language Model

$P(Y_1|X_1)$ $P(Y_2|X_1,X_2)$ $P(Y_3|X_1,X_2,X_3)$



X_1 X_2 X_3 X_4

Using a Trained Model we can compute
$P(X_1,X_2,X_3)=P(X_1)P(Y_1=X_2|X_1)P(Y_2=X_3|X_1,X_2)$

Can a Language Model also be used to Generate Sentences?

Back to the Chain Rule of Probabilities

$$P(X_1,X_2,X_3) = P(X_1)\ P(X_2|X_1)\ P(X_3|X_1,X_2)$$

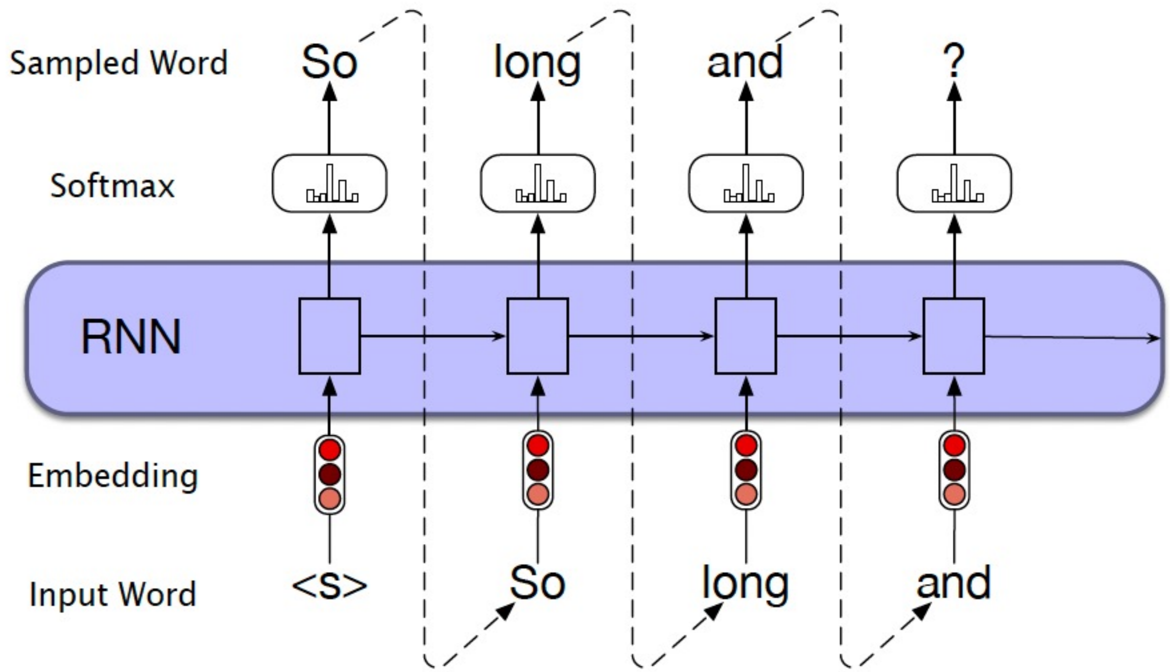
Start with X_1

Sample $(Y_1=X_2|X_1)$ to generate X_2

Sample $(Y_2=X_3|X_1, X_2)$ to generate X_3

.
.
.

Language Generation



Auto-Regressive Network!

The output of the network serves as its next input

Sampling Methods

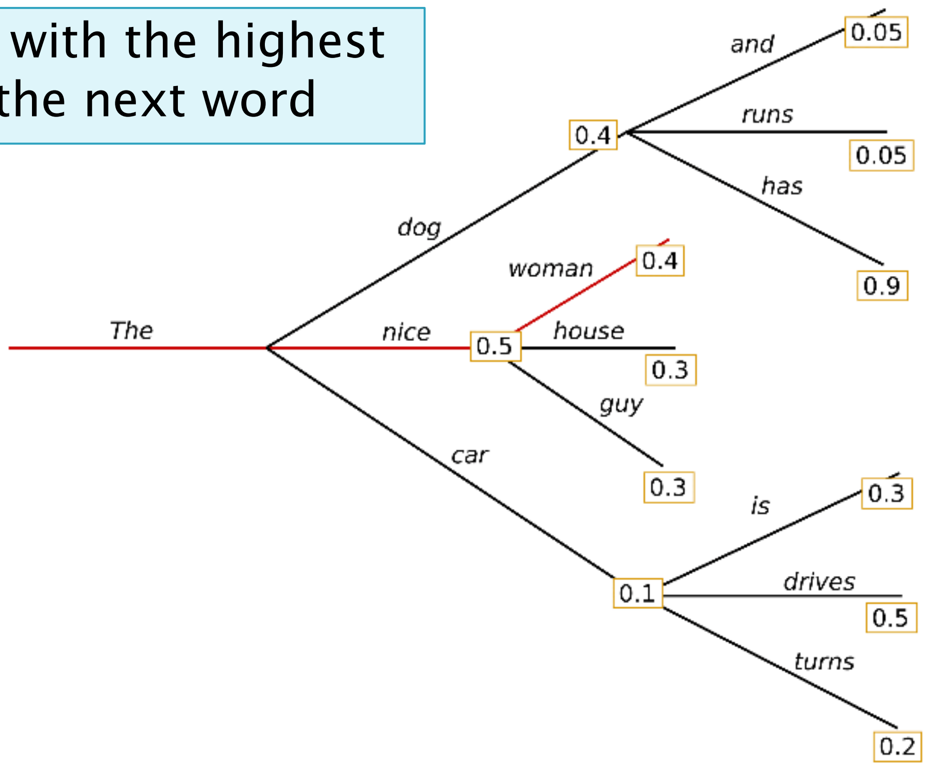
Generating some Randomness during Sampling

Techniques:

- Greedy Search
- Beam Search
- Sampling
- Sampling with Temperature
- Top-K Sampling
- Top-p (Nucleus) Sampling

Greedy Search

Chooses the word with the highest probability as the next word

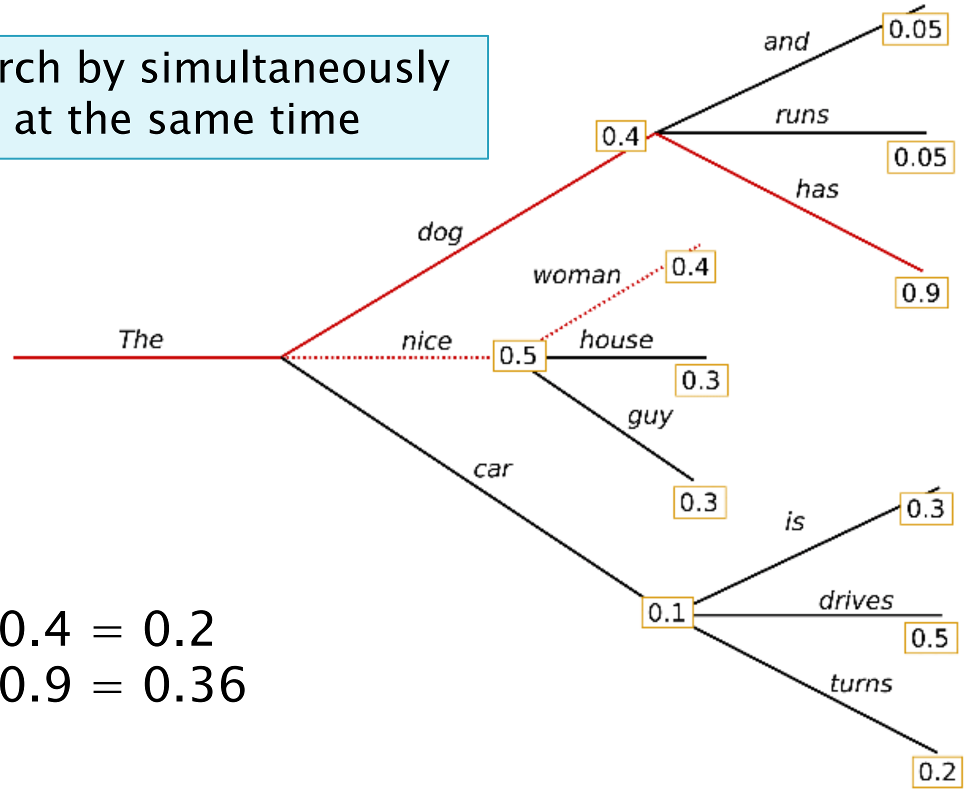


Output: I enjoy walking with my cute dog, but I'm not sure if I'll ever be able to walk with my dog. I'm not sure if I'll ever be able to walk with my dog. I'm not sure if I'll

https://huggingface.co/blog/how-to-generate

Beam Search

Tries to remedy Greedy Search by simultaneously generating B sentences at the same time



B=2

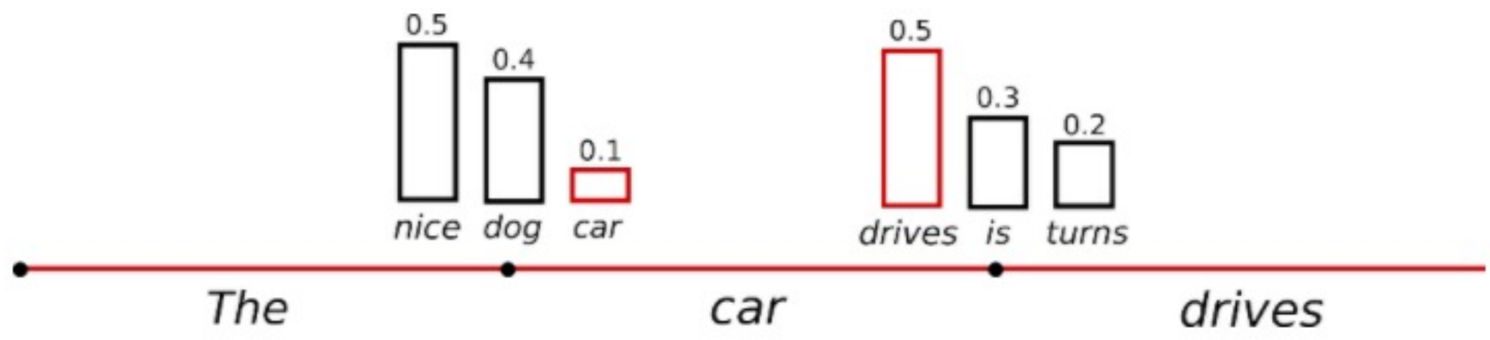
<u>Stage 1</u>
The nice – 0.5
The dog – 0.4

<u>Stage 2</u>
The nice woman – 0.5 x 0.4 = 0.2
The dog has – 0.4 x 0.9 = 0.36

Output: I enjoy walking with my cute dog, but I'm not sure if I'll ever be able to walk with him again. I'm not sure if I'll ever be able to walk with him again. I'm not sure if I'll

Sampling

At each stage of the Language Model, we sample from the output distribution to generate the next word



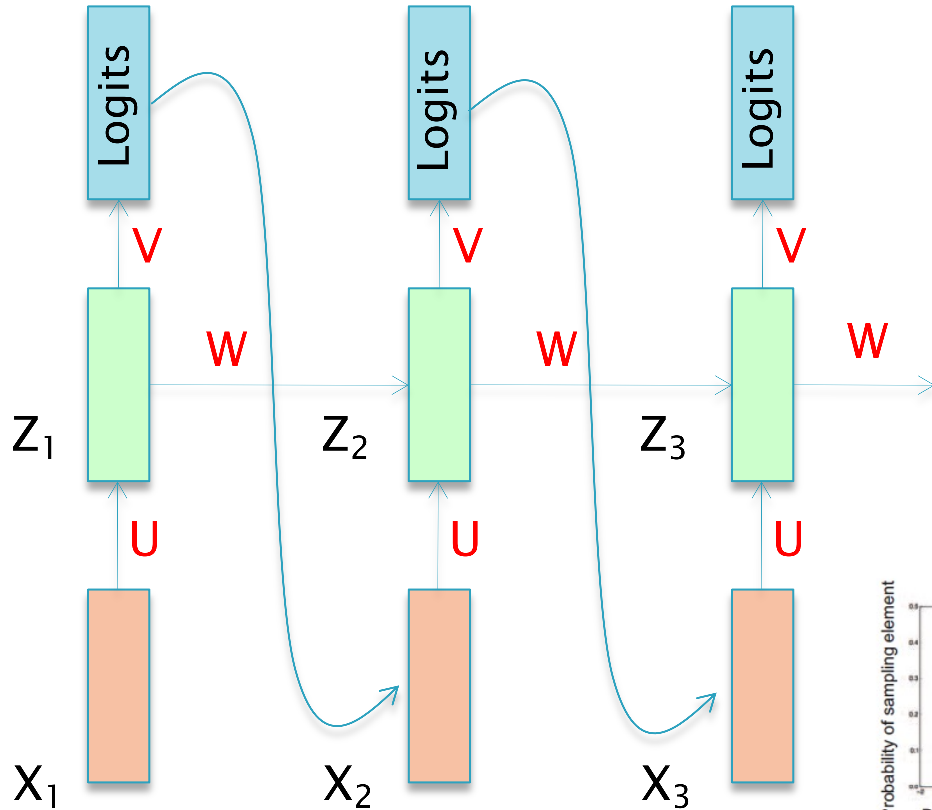
Output: I enjoy walking with my cute dog. He just gave me a whole new hand sense."
But it seems that the dogs have learned a lot from teasing at the local batte harness once they take on the outside. "I take

Using Softmax Temperature

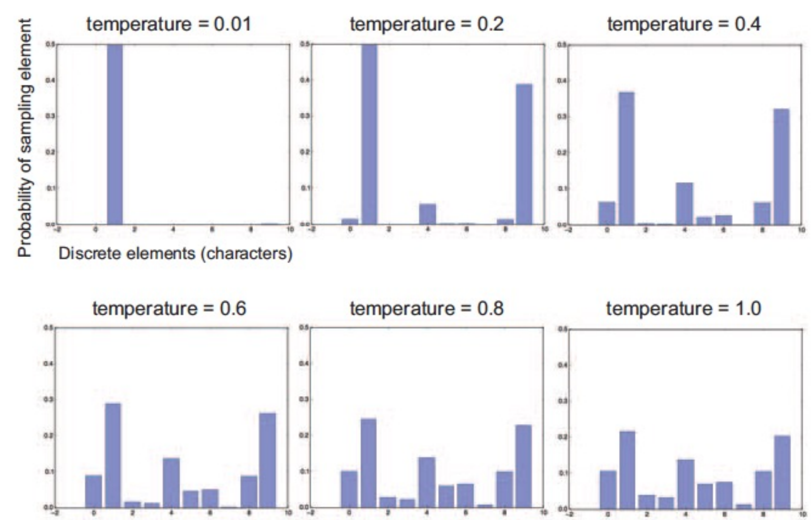
$X_3 = sample\ P(Y_2|X_1, X_2)$

$X_2 = sample\ P(Y_1|X_1)$

$X_4 = sample\ P(Y_3|X_1, X_2, X_3)$



$$P(b_i) = \frac{\exp(\frac{a_i}{T})}{\sum \exp(\frac{a_i}{T})}$$

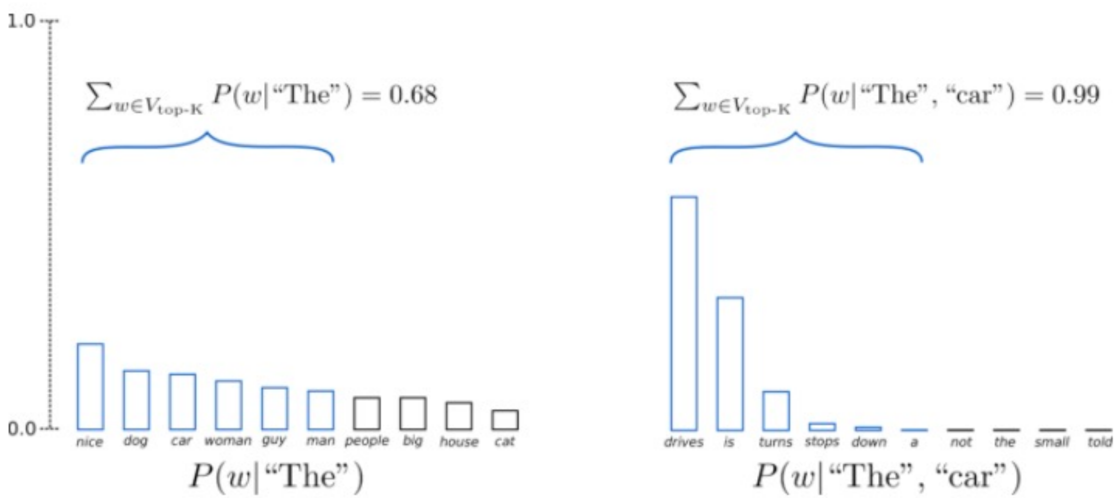


Sampling with Low Softmax Temperature

Output: I enjoy walking with my cute dog, but I don't like to be at home too much.
I also find it a bit weird when I'm out shopping.
I am always away from my house a lot, but I do have a few friends

Top-K Sampling (2018)

In *Top-K* sampling, the *K* most likely next words are filtered and the probability mass is redistributed among only those *K* next words.

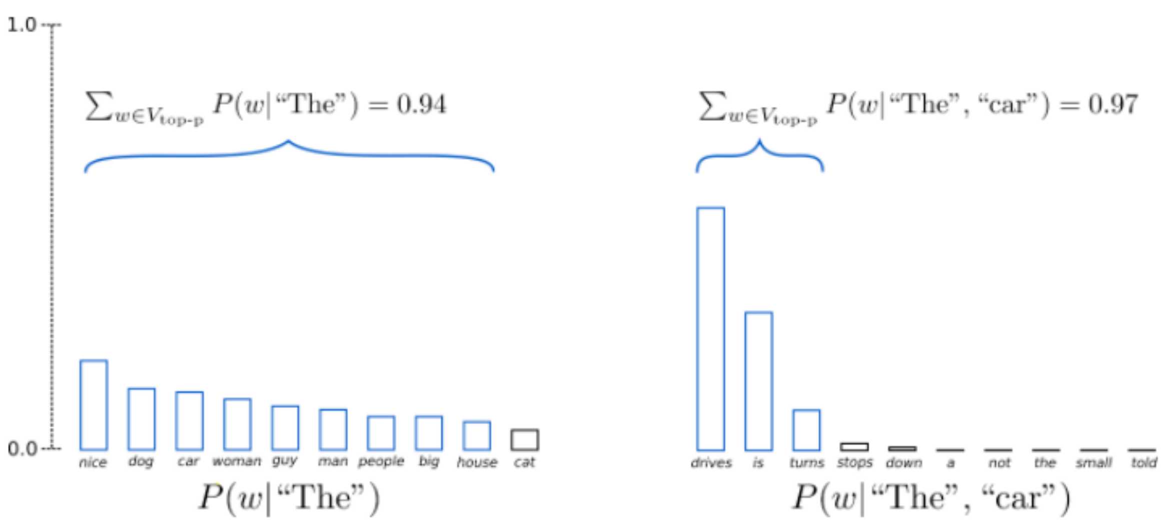


K = 6

Output: I enjoy walking with my cute dog. It's so good to have an environment where your dog is available to share with you and we'll be taking care of you. We hope you'll find this story interesting! I am from

Top-p (Nucleus) Sampling (2019)

Instead of sampling only from the most likely *K* words, in *Top-p* sampling chooses from the smallest possible set of words whose cumulative probability exceeds the probability *p*. The probability mass is then redistributed among this set of words. This way, the size of the set of words (*a.k.a* the number of words in the set) can dynamically increase and decrease according to the next word's probability distribution.



p=0.92

Output: I enjoy walking with my cute dog. He will never be the same. I watch him play. Guys, my dog needs a name. Especially if he is found with wings.
What was that? I had a lot of

Top-K + Top-p

K=50, p=0.95

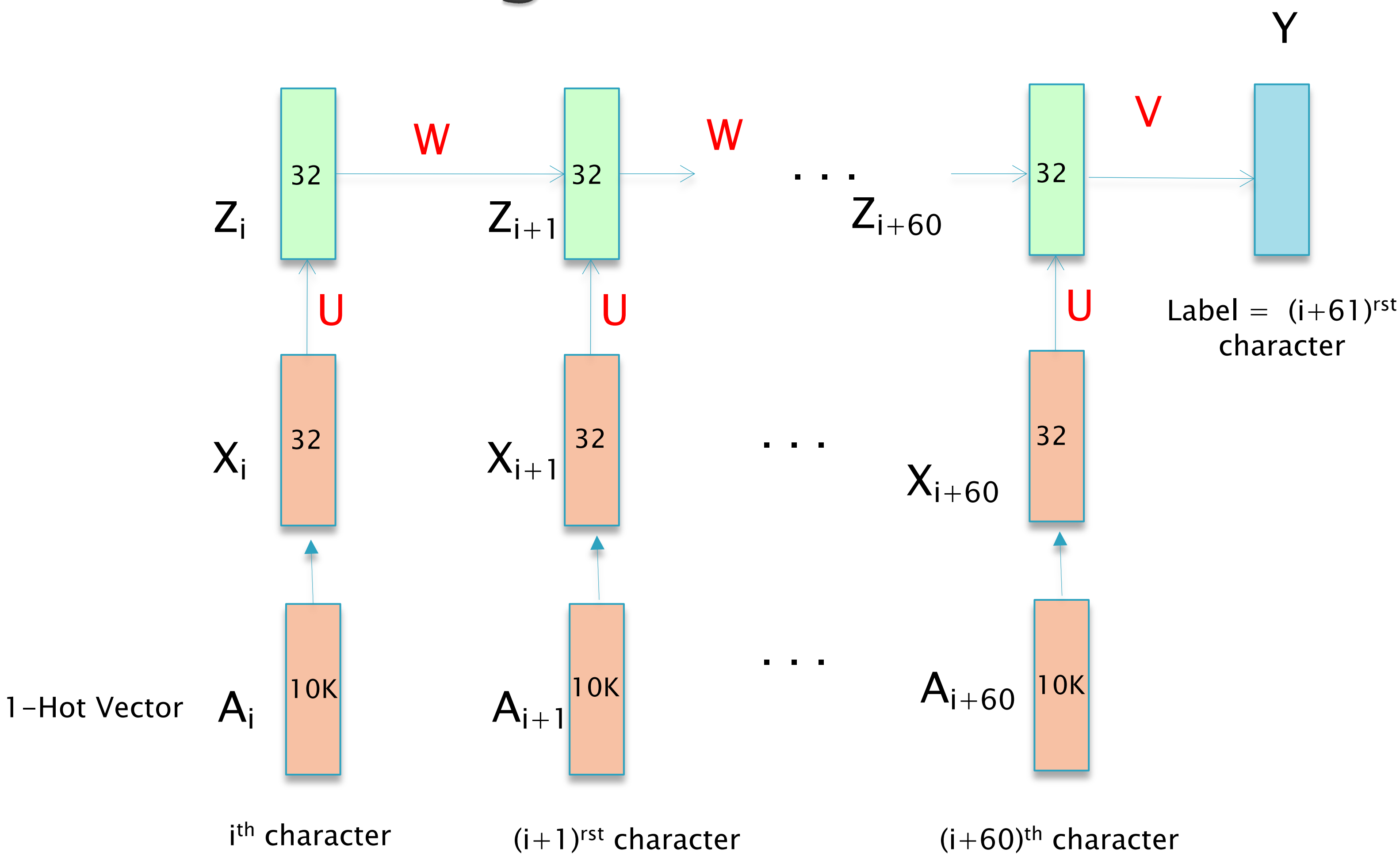
0: I enjoy walking with my cute dog. It's so good to have the chance to walk with a dog. But I have this problem with the dog and how he's always looking at us and always trying to make me see that I can do something

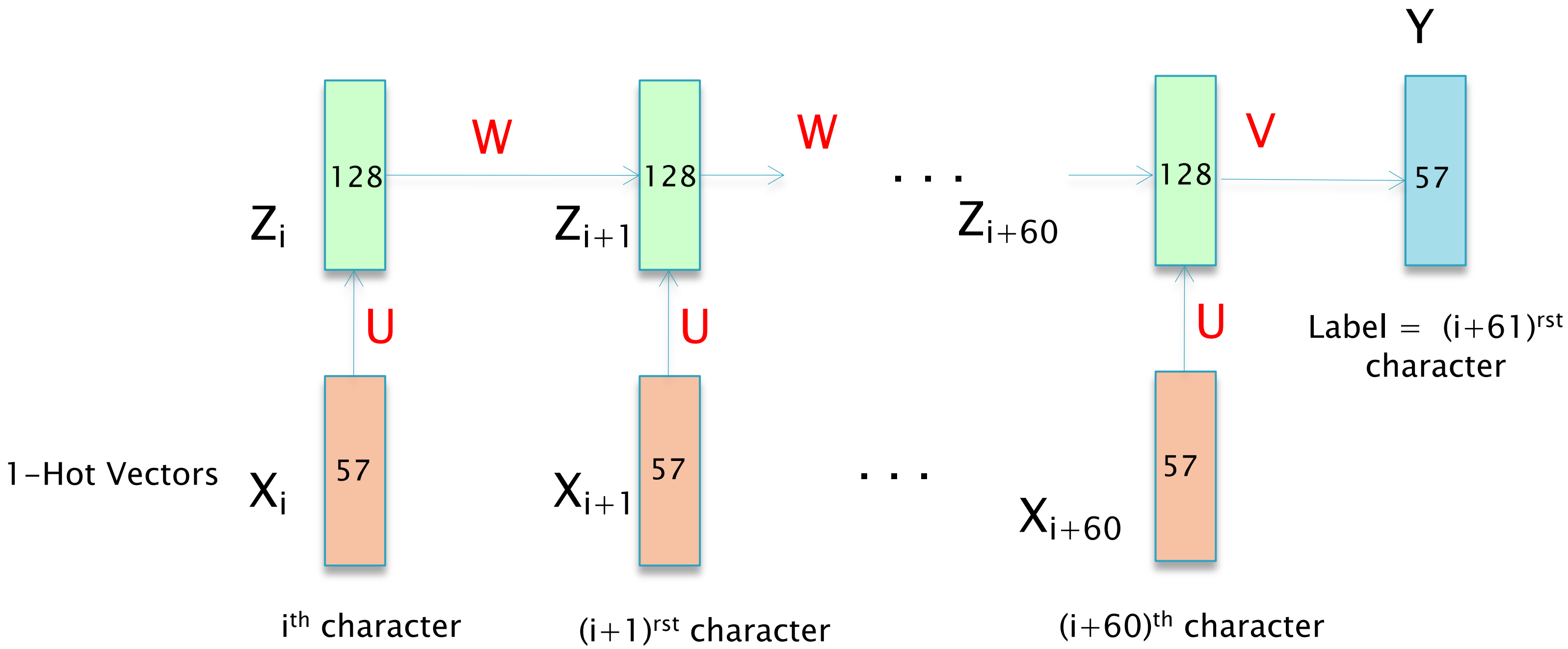
1: I enjoy walking with my cute dog, she loves taking trips to different places on the planet, even in the desert! The world isn't big enough for us to travel by the bus with our beloved pup, but that's where I find my love

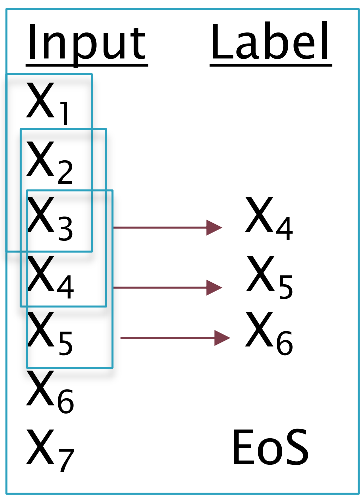
2: I enjoy walking with my cute dog and playing with our kids," said David J. Smith, director of the Humane Society of the US. "So as a result, I've got more work in my time," he said.

Character Based Language Model

Character Based Language Model:Training







A single input Sample

char index

$$
\begin{array}{l c}
 & \begin{array}{cccc} 1 & 2 & 3 & \ldots 57 \end{array} \\
\begin{array}{l} \text{char 1} \\ \text{char 2} \\ \text{char 3} \\ \vdots \\ \text{char 60} \end{array} &
\begin{pmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ & & \vdots & & \\ 0 & 0 & 0 & \ldots & 1 \end{pmatrix}
\end{array}
$$

Example: Generating Shakespeare

```
PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.

Second Lord:
They would be ruled after this chamber, and
my fair nues begun out of the fact, to be conveyed,
Whose noble souls I'll have the heart of the wars.

Clown:
Come, sir, I will make did behold your worship.

VIOLA:
I'll drink it.
```

- Trained using all the works of Shakespeare concatenated into a single (4.4MB) file.
- Using a 3 layer LSTM with 512 nodes per layer

Example: Generating Tolstoy

at first:

```
tyntd-iafhatawiaoihrdemot  lytdws  e ,tfti, astai f ogoh eoase rrranbyne 'nhthnee e
plia tklrgd t o idoe ns,smtt   h ne etie h,hregtrs nigtike,aoaenns lng
```

train more

```
"Tmont thithey" fomesscerliund
Keushey. Thom here
sheulke, anmerenith ol sivh I lalterthend Bleipile shuwy fil on aseterlome
coaniogennc Phe lism thond hon at. MeiDimorotion in ther thize."
```

train more

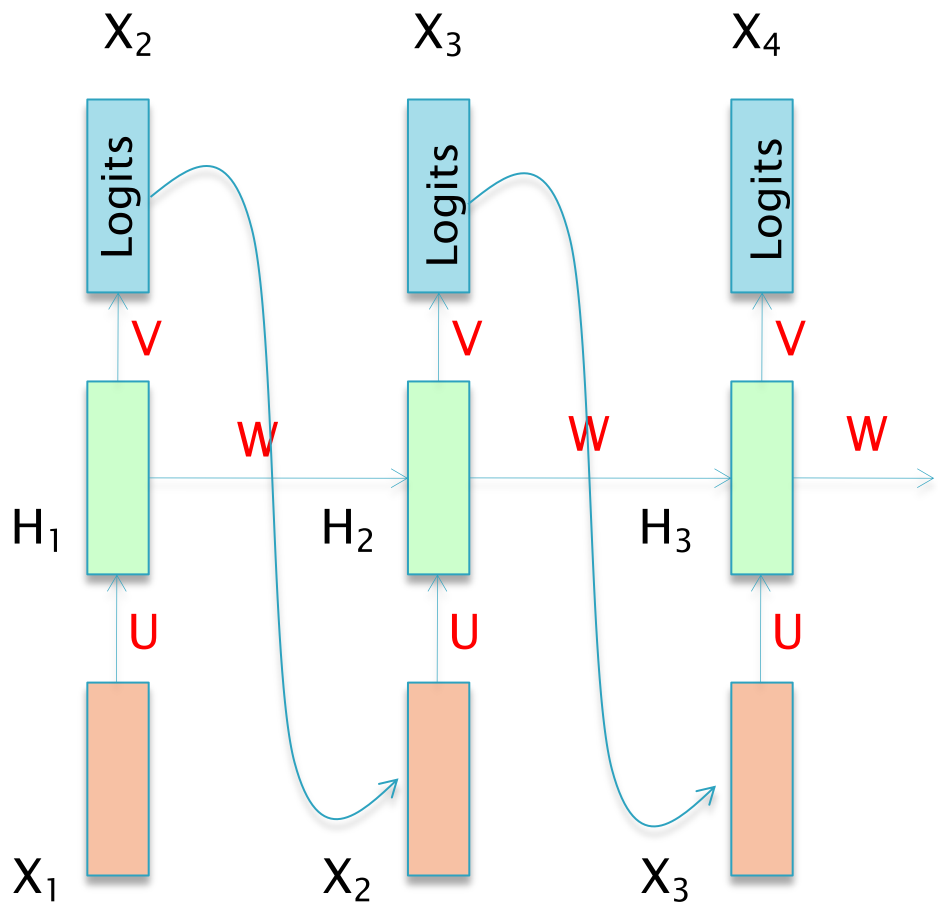
```
Aftair fall unsuch that the hall for Prince Velzonski's that me of
her hearly, and behs to so arwage fiving were to it beloge, pavu say falling misfort
how, and Gogition is so overelical and ofter.
```

train more

```
"Why do what that day," replied Natasha, and wishing to himself the fact the
princess, Princess Mary was easier, fed in had oftened him.
Pierre aking his soul came to the packs and drove up his father-in-law women.
```

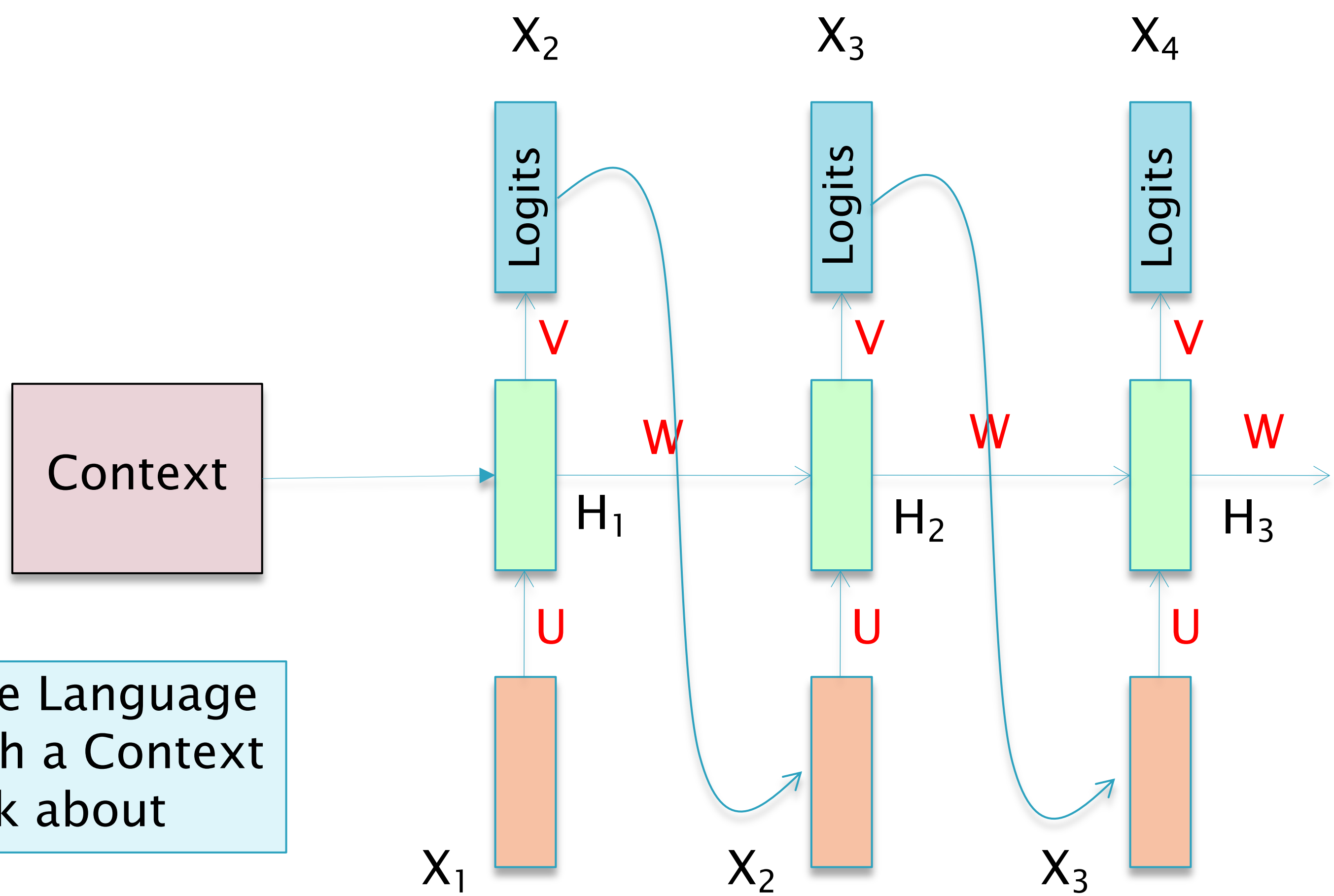
Encoder Decoder Systems

So Far..



We know how to generate sentences but
What is the sentence talking about?

Conditional Language Models

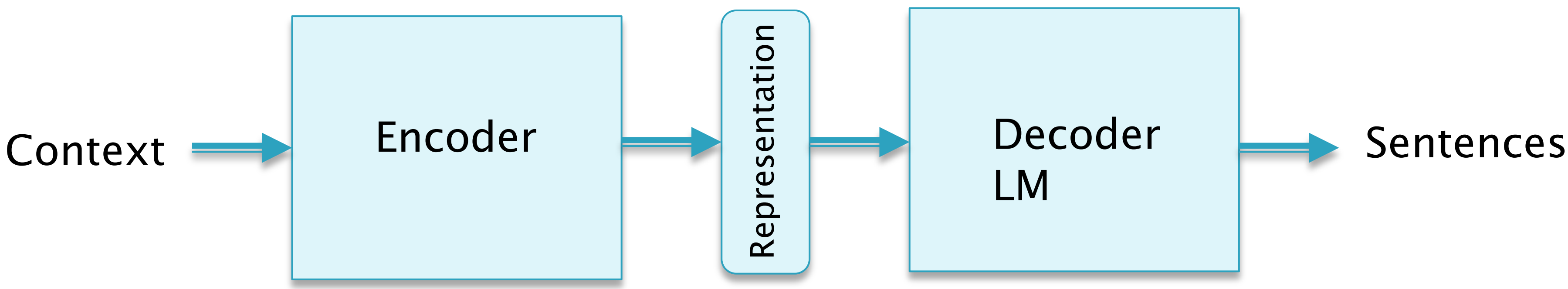


Supply the Language Model with a Context to talk about

Applications of Conditional LMs

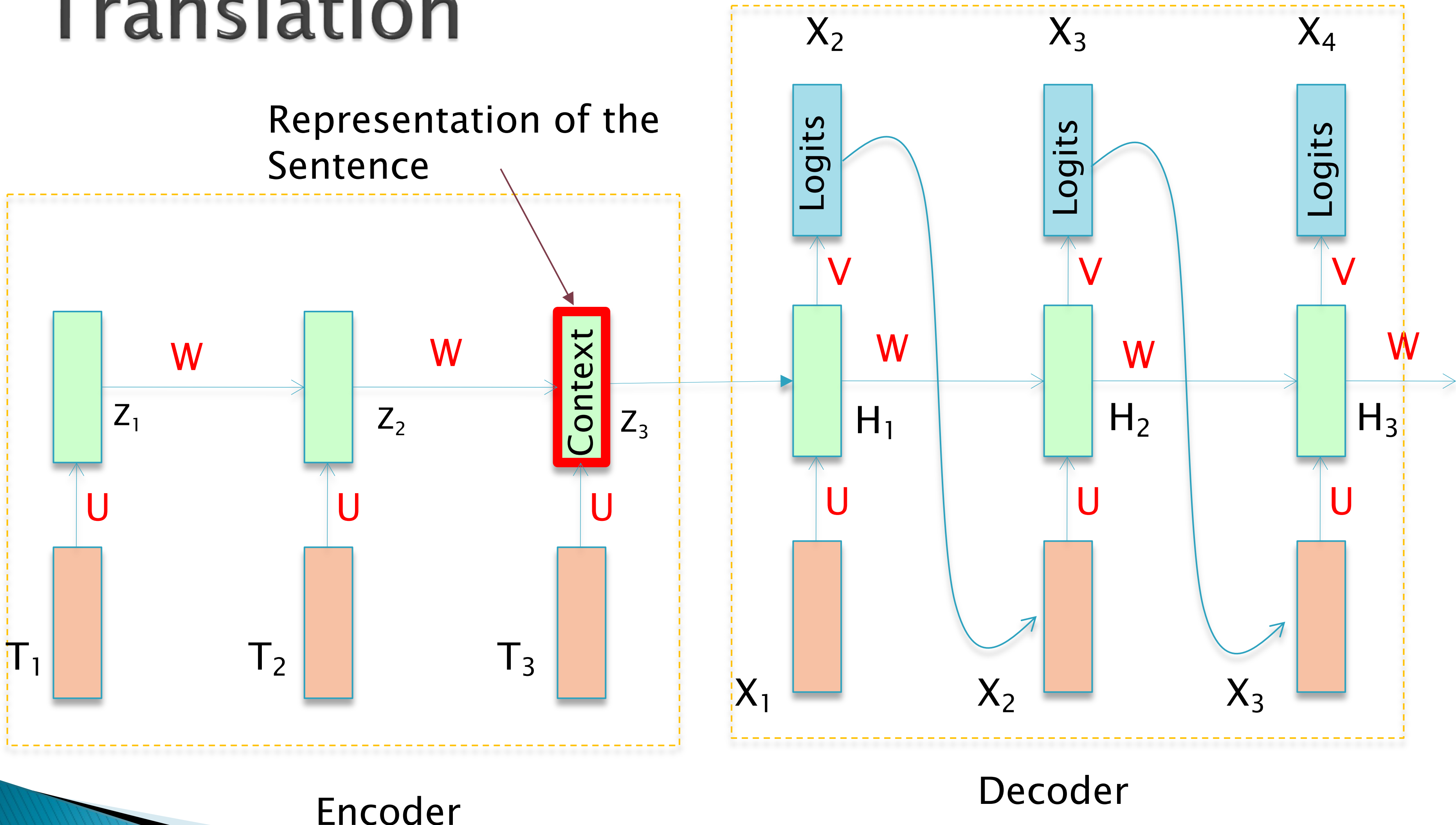
Context	Language Model Output
A sentence in French	Its English Translation
An image	A text description of the image
A document	Its Summary
An acoustic signal	Transcription of Speech
A question + Document	Its Answer
A question + Image	Its Answer
Meteorological Measurements	A weather report
Conversational History + Database	Dialogue system response
An Email	Auto Reply to the Email

Encoder–Decoder Systems



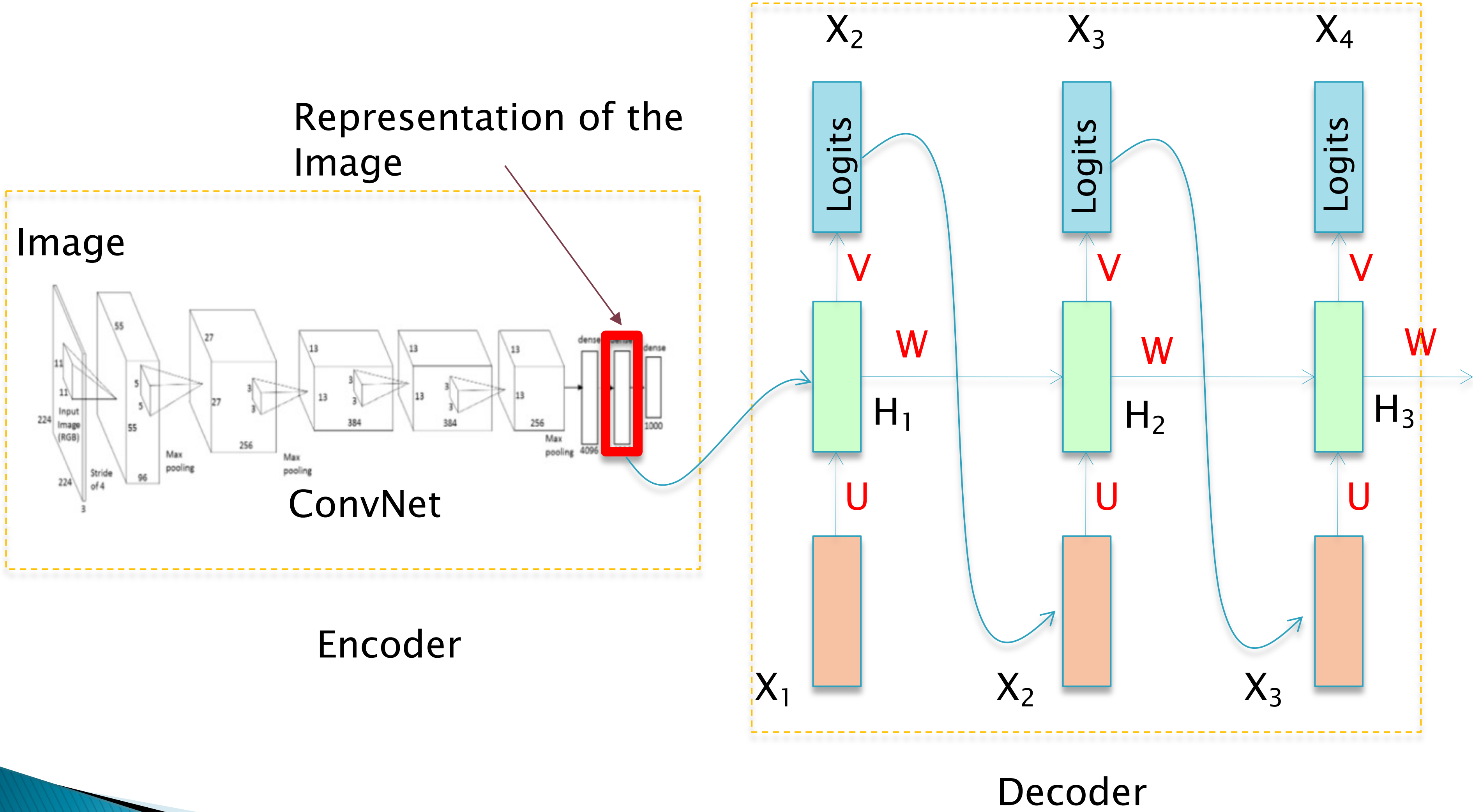
https://keras.io/examples/lstm_seq2seq/

Generating a Context: M/C Translation

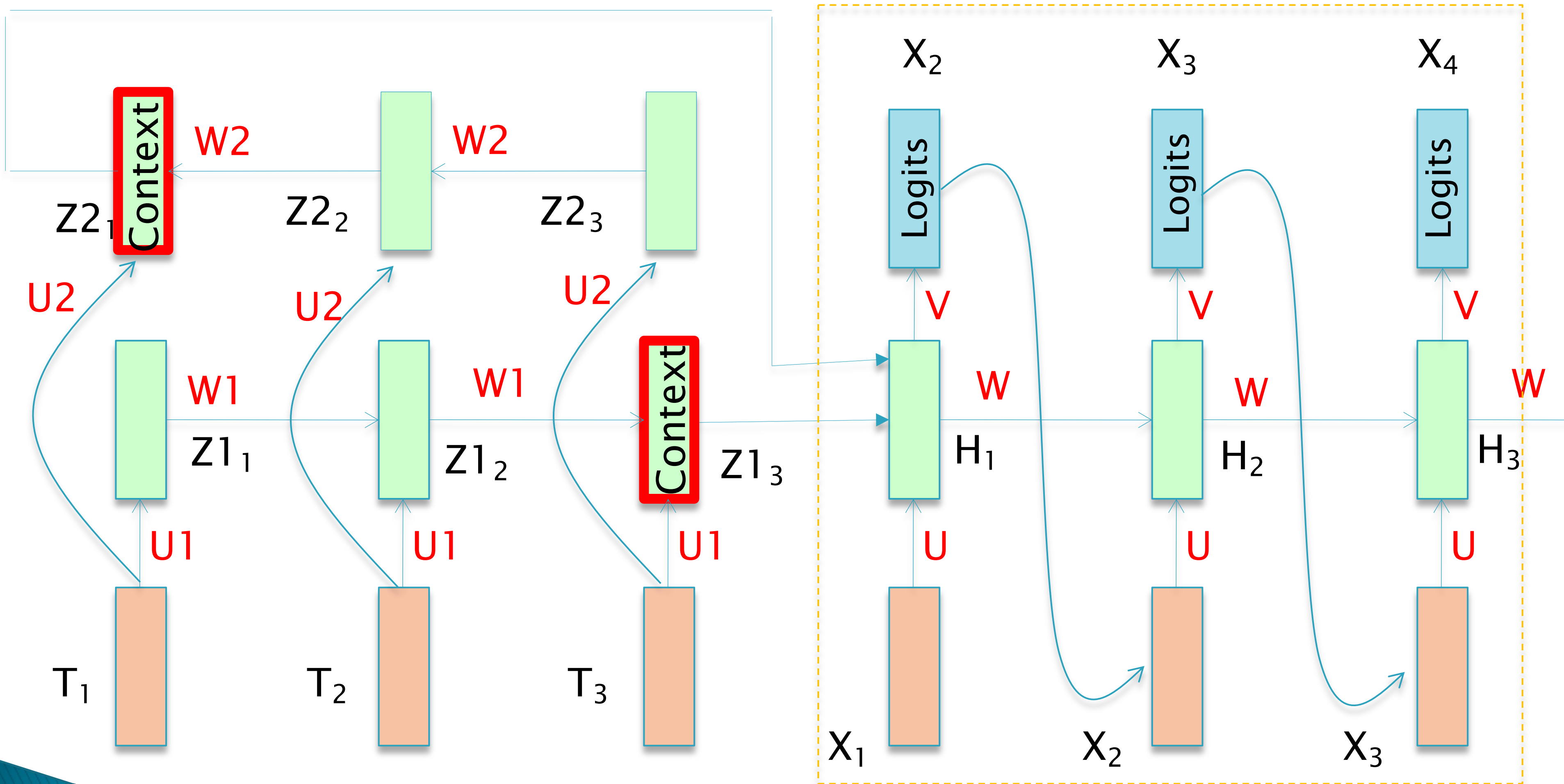


Encode Decoder systems
Sequence to Sequence Systems

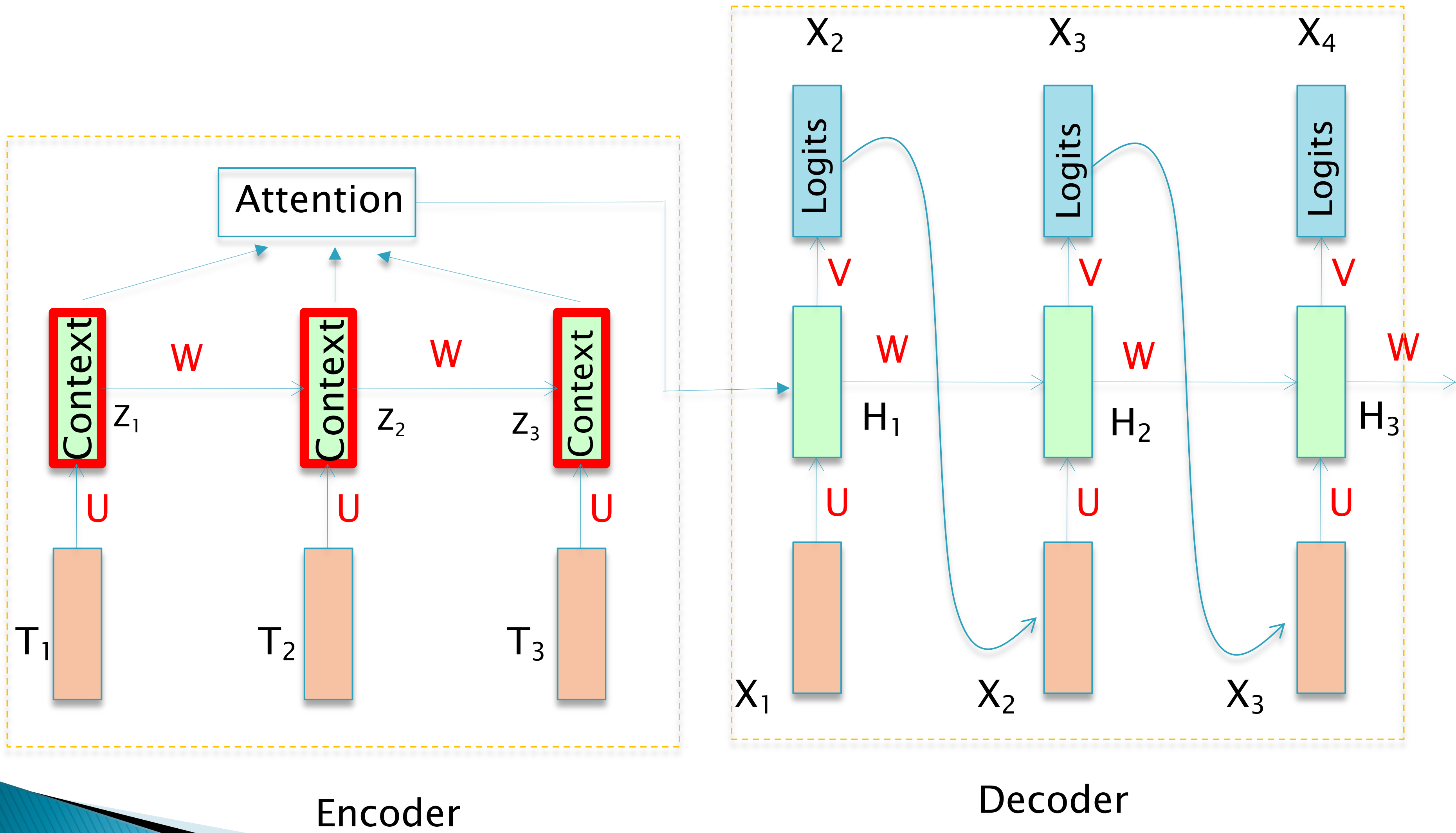
Generating a Context: Captioning



Generating a Context



Generating a Context



Further Reading

- Das and Varma: ChapterNLP
- Chollet: Chapter 11, Sections 11.1, 11.2, 11.3
  Chapter 12, Section 12.1

  For a deeper dive into NLP:
- Jurafsky and Martin: Speech and Language Processing, 3rd Edition