Deep Learning with Neural Networks

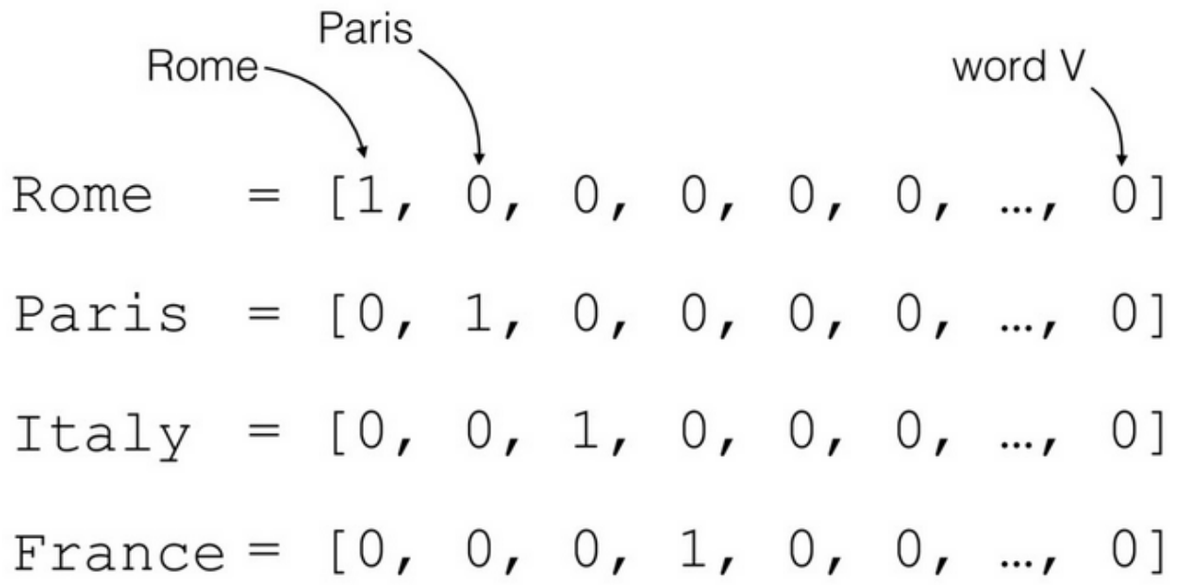
Word Embeddings,
Seq2Seq, Attention Models

Pablo Martínez Olmos, pamartin@ing.uc3m.es

Word Embeddings

Word Embeddings

- One-hot encoding does not scale well with dictionary size
- All one-hot vectors are equally distant! Neighbours do not mean anything!



Word Embeddings

- With word embeddings (WE) our goals are

  1. Find vector representation of words in an **unsupervised manner**

  2. **Semantically meaningful** → Neighbours matter!

- There are several variants of WE, here we will present the most common one:
  The Skip-gram model (a.k.a. Word2Vec).

Distributed Representations of Words and Phrases and their Compositionality

Tomas Mikolov
Google Inc.
Mountain View
mikolov@google.com

Ilya Sutskever
Google Inc.
Mountain View
ilyasu@google.com

Kai Chen
Google Inc.
Mountain View
kai@google.com

Greg Corrado
Google Inc.
Mountain View
gcorrado@google.com

Jeffrey Dean
Google Inc.
Mountain View
jeff@google.com

The Wor2Vec model (I)

- We are given a specific word in the middle of a sentence (**the input word**).
- We will train a simple NN to predict the probability for every word in our vocabulary of **being nearby the input word**.
- We want to **predict neighbours in a window-size of W** words.
- Typically W is 3 or 5.
- Train the neural network by feeding it with word pairs found in our training documents.
- **Unsupervised problem trained as a supervised one**!!

The Wor2Vec model (II)

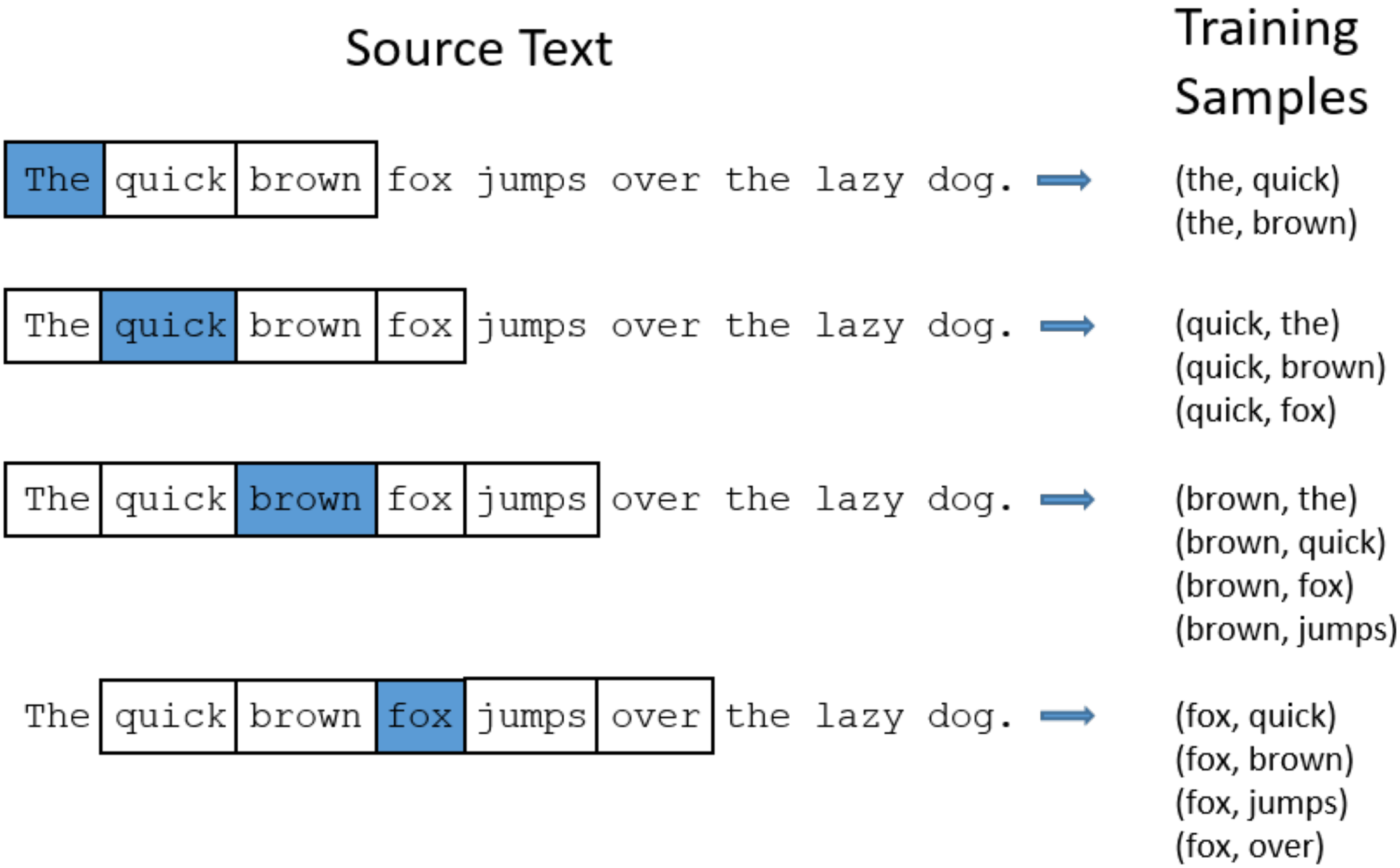


Figure source: Chris McCormick's blog entry on Wor2Vec

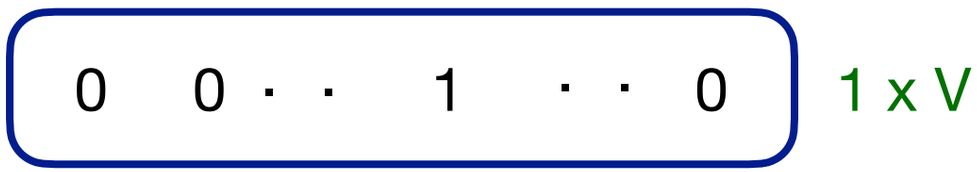
The Wor2Vec model (III)

The Wor2Vec model (III)

Input

One-hot

Assume *i*-th position is 1



The Wor2Vec model (III)

Input

One-hot

Assume *i*-th position is 1

0 0 · · 1 · · 0

1 x V

x

Weight Matrix W

V x K

The Wor2Vec model (III)

Input

One-hot

Assume *i*-th position is 1

0 0 · · 1 · · 0

1 x V

x

Weight Matrix W

V x K

=

Word Embedding

(*i*-th row of W)

$\mathbb{R}^K$

1 x K

The Wor2Vec model (III)

Input

One-hot

Assume *i*-th position is 1

0 0 · · 1 · · 0 — 1 x V

x

Weight Matrix W

V x K

= **Word Embedding** (*i*-th row of W)

$\mathbb{R}^K$

1 x K

x

Weight Matrix W'

K x V

The Wor2Vec model (III)

Input
One-hot
Assume *i*-th position is 1

0 0 · · 1 · · 0

1 x V

x

Weight Matrix W

V x K

=

Word Embedding
(*i*-th row of W)

$\mathbb{R}^K$

1 x K

x

Weight Matrix W'

K x V

=

1 x V

Output
(softmax)

0.01 0.0015 ⋯

The Wor2Vec model (III)

Input
One-hot
Assume *i*-th position is 1

0 0 · · 1 · · 0

1 x V

x

Weight Matrix W

V x K

=

Word Embedding
(*i*-th row of W)

$\mathbb{R}^K$

1 x K

x

Weight Matrix W'

K x V

=

Output
(softmax)

1 x V

0.01 0.0015 ···

Multi-class LR with input a WE!!

The Wor2Vec model (IV)

K=300, input word = “ants”, output word = “words”

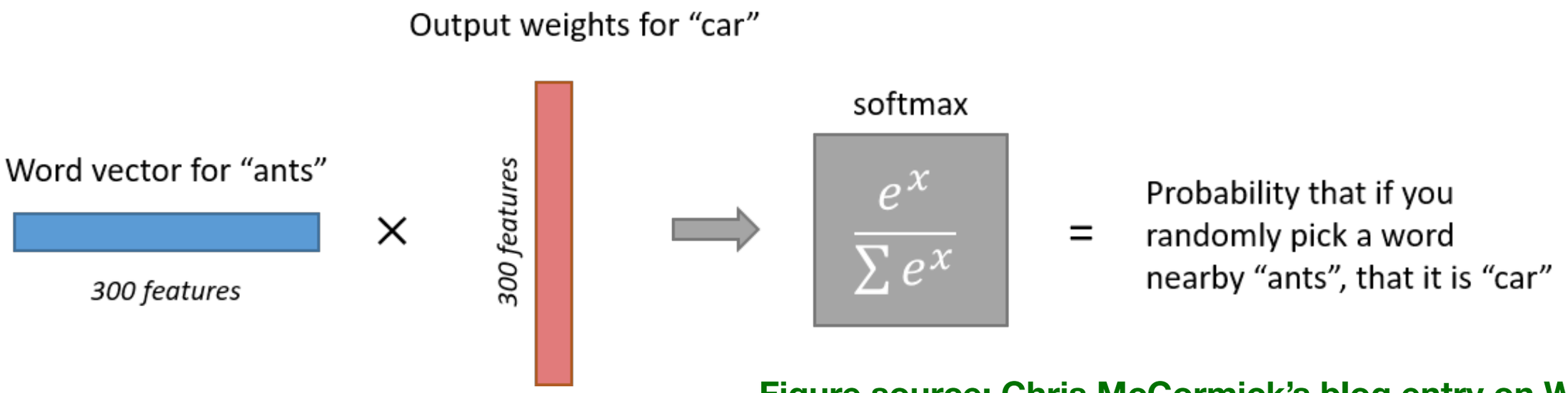


Figure source: Chris McCormick’s blog entry on Wor2Vec

The Wor2Vec model (IV)

K=300, input word = “ants”, output word = “words”

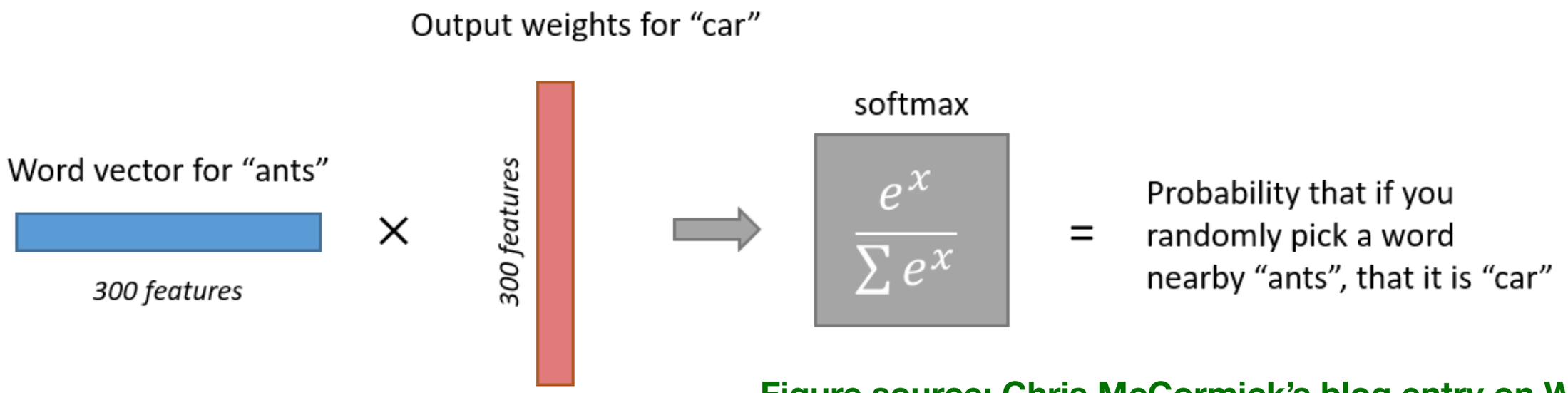


Figure source: Chris McCormick’s blog entry on Wor2Vec

- If two different words have very **similar contexts** (that is, what words are likely to appear around them), then our model needs to output very similar results for these two words.

- And one way for the network to output similar context predictions for these two words is if the **word vectors are similar**.

- The network will likely learn similar word vectors for the words ant and ants because these should have similar contexts.

Efficient Training of Wor2Vec

Efficient Training of Wor2Vec

- Word embeddings 300 components, and a vocabulary of 10,000 words
- Weight matrices of size 300 x 10,000 = 3 million weights each!
- Two main techniques to improve training efficiency:

1. **Subsampling frequent words**
2. **Negative Sampling**

Subsampling Frequent Words

Subsampling Frequent Words

- **Remove neutral frequent words** such as ”the”, ”and”, ”then ...”. They appear in the context of pretty much every word.

- Very frequent words will appear too often as an input to the WE classifier, introducing a bias.

- Very frequent words are **subsampled with a probability that is inversely proportional to the fraction of times this word appears in the corpus**.

Negative Sampling

Negative Sampling

- When training the network on the word pair (fox, quick),
  recall that the label or correct output of the network is a one-hot vector.

- Each training sample only modify a **small percentage of the weights**, rather than all of them.

- With Negative sampling we **select just a small number of negative words**
  to update the weights for.

- Namely, **we evaluate the softmax for only a few words!**

- Milokov's paper says that selecting **5-20 words works well for smaller datasets**, and you ca
  get away with only 2-5 words for large datasets.

- The probability for selecting a word as a negative sample is related to its frequency.

Model Evaluation

Journal of Machine Learning Research 1 (2008) 1-48 Submitted 4/00; Published 10/00

Visualizing Data using t-SNE

Laurens van der Maaten L.VANDERMAATEN@MICC.UNIMAAS.NL
MICC-IKAT
Maastricht University
P.O. Box 616, 6200 MD Maastricht, The Netherlands

Geoffrey Hinton HINTON@CS.TORONTO.EDU
Department of Computer Science
University of Toronto
6 King's College Road, M5S 3G4 Toronto, ON, Canada

Model Evaluation

- Once the model is trained, we can evaluate the space of input embeddings
- For instance, look at the closest neighbors of a given word.
- Use dimensionality reduction to visualize the embedding space. **TSNE is recommended** (it keeps local structure)!

Journal of Machine Learning Research 1 (2008) 1-48 Submitted 4/00; Published 10/00

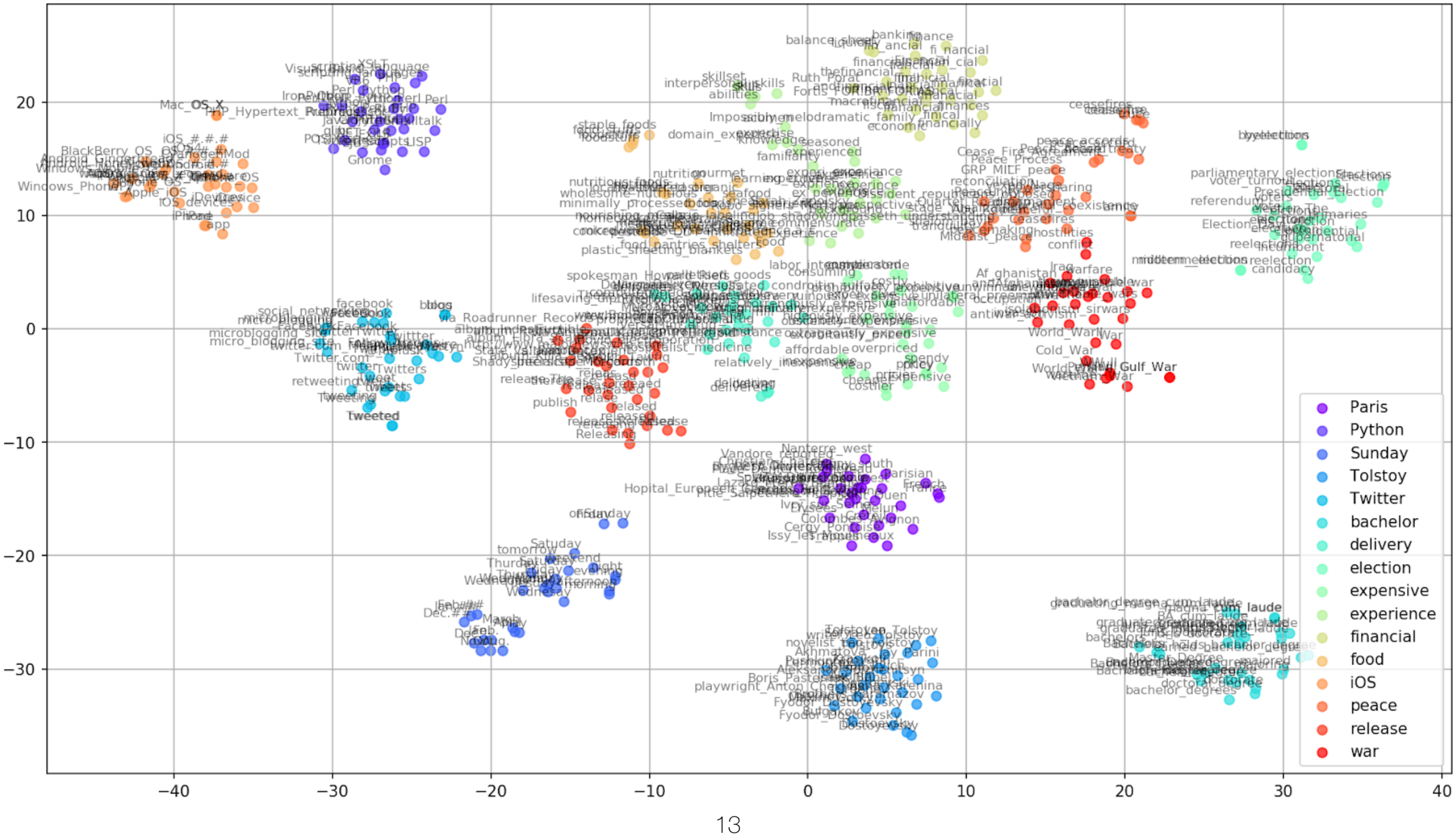
Visualizing Data using t-SNE

Laurens van der Maaten L.VANDERMAATEN@MICC.UNIMAAS.NL
MICC-IKAT
Maastricht University
P.O. Box 616, 6200 MD Maastricht, The Netherlands

Geoffrey Hinton HINTON@CS.TORONTO.EDU
Department of Computer Science
University of Toronto
6 King's College Road, M5S 3G4 Toronto, ON, Canada

WE + 2-dimensional TSNE

- Word2vec (300 hundred dimensions) trained over Google News Dataset (>1million news headlines)



Vector operations with WE

Linguistic Regularities in Continuous Space Word Representations

Tomas Mikolov*, Wen-tau Yih, Geoffrey Zweig
Microsoft Research
Redmond, WA 98052

Vector operations with WE

- It has been shown that the word vectors capture many linguistic regularities.
- **vector('Paris') - vector('France') + vector('Italy')** results in a vector that is very close to **vector('Rome')**.
- **vector('king') - vector('man') + vector('woman')** is close to **vector('queen')**
- To observe strong regularities in the word vector space, it is needed to train the models on large data set, with sufficient vector dimensionality

Linguistic Regularities in Continuous Space Word Representations

Tomas Mikolov*, Wen-tau Yih, Geoffrey Zweig
Microsoft Research
Redmond, WA 98052

Pre-trained Word Embeddings

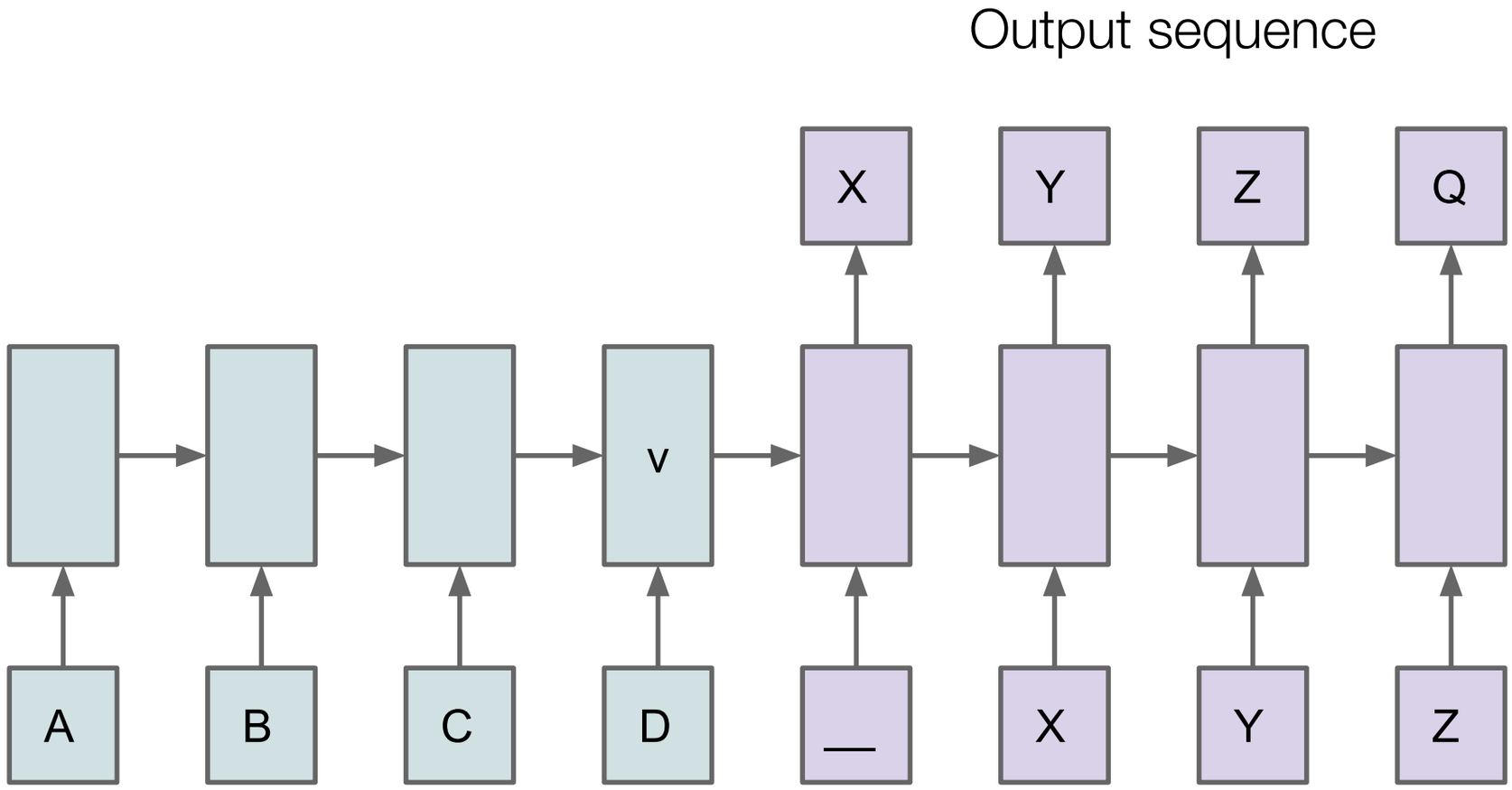
Pre-trained Word Embeddings

- Pre-trained word embeddings are available in most current NLP libraries.
  - **Gensim**
  - **Torchtext**
  - **Spacy**

among many others … (these are just the three ones that I know)

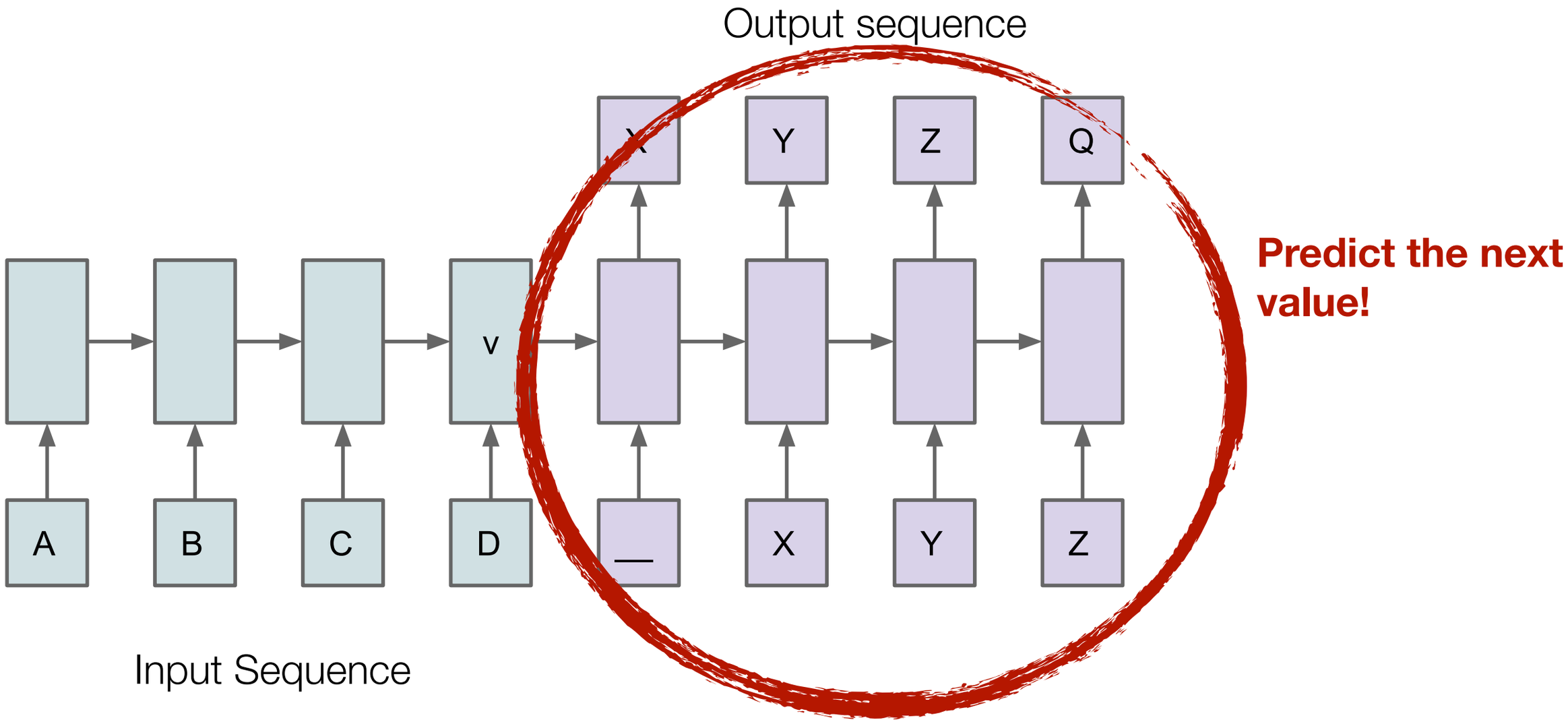
Seq2Seq & Attention Networks

Seq2Seq



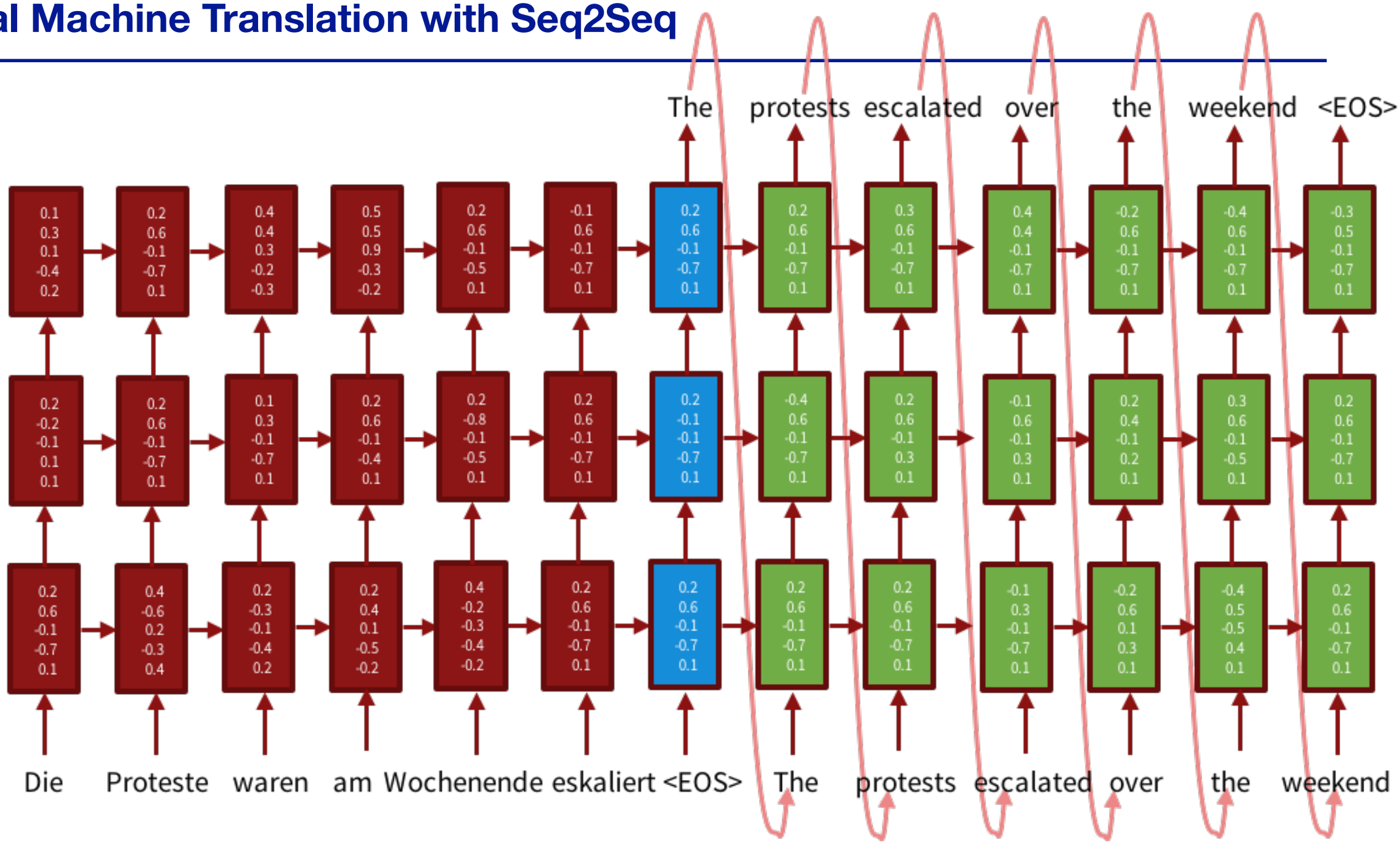
$$P\left(y_1, \ldots, y_{T'} | x_1, \ldots, x_T\right) = \prod_{t=1}^{T'} p\left(y_t | v, y_1, \ldots, y_{t-1}\right)$$

Seq2Seq



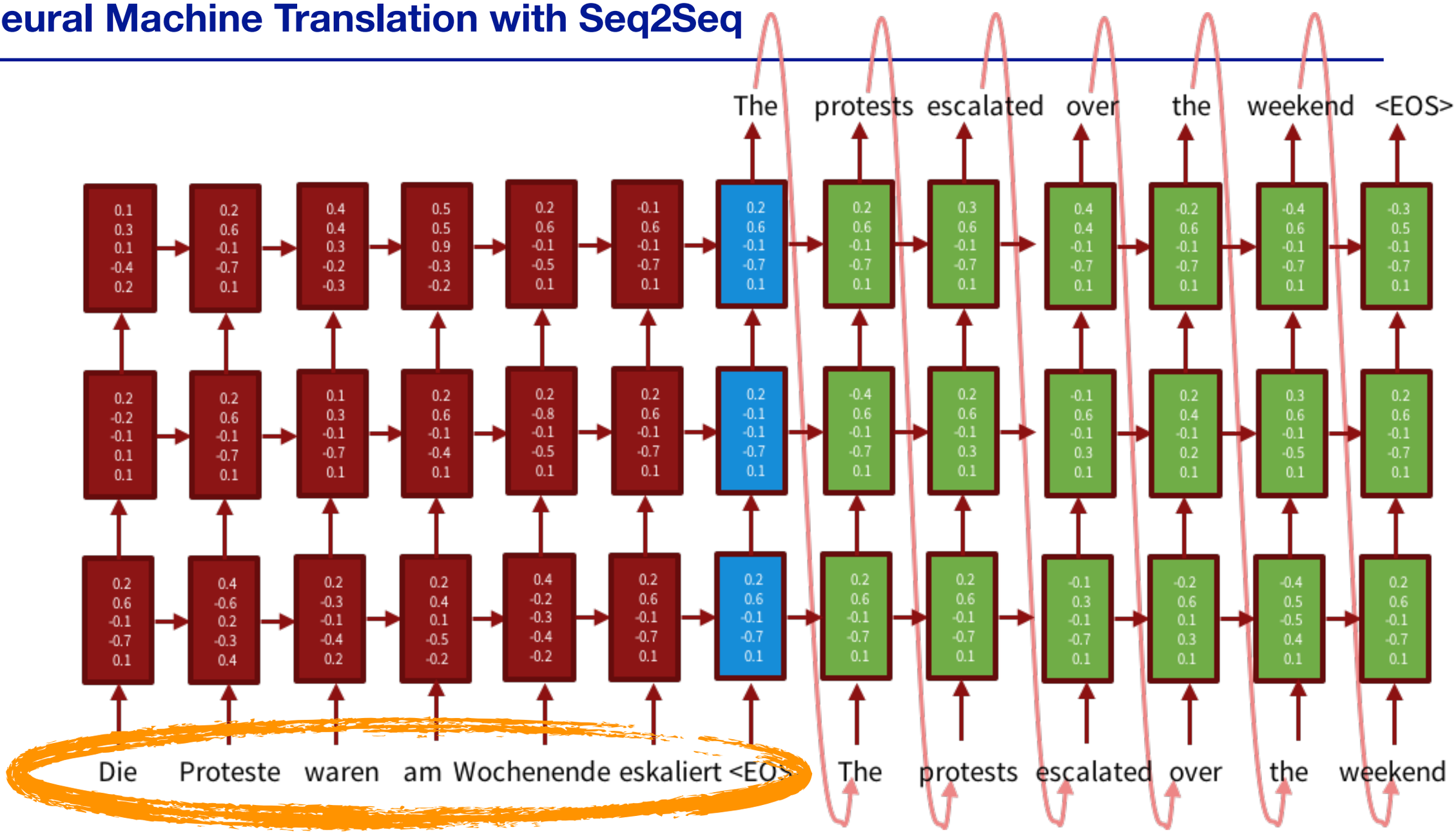
$$P\left(y_1, \ldots, y_{T'} | x_1, \ldots, x_T\right) = \prod_{t=1}^{T'} p\left(y_t | v, y_1, \ldots, y_{t-1}\right)$$

Neural Machine Translation with Seq2Seq



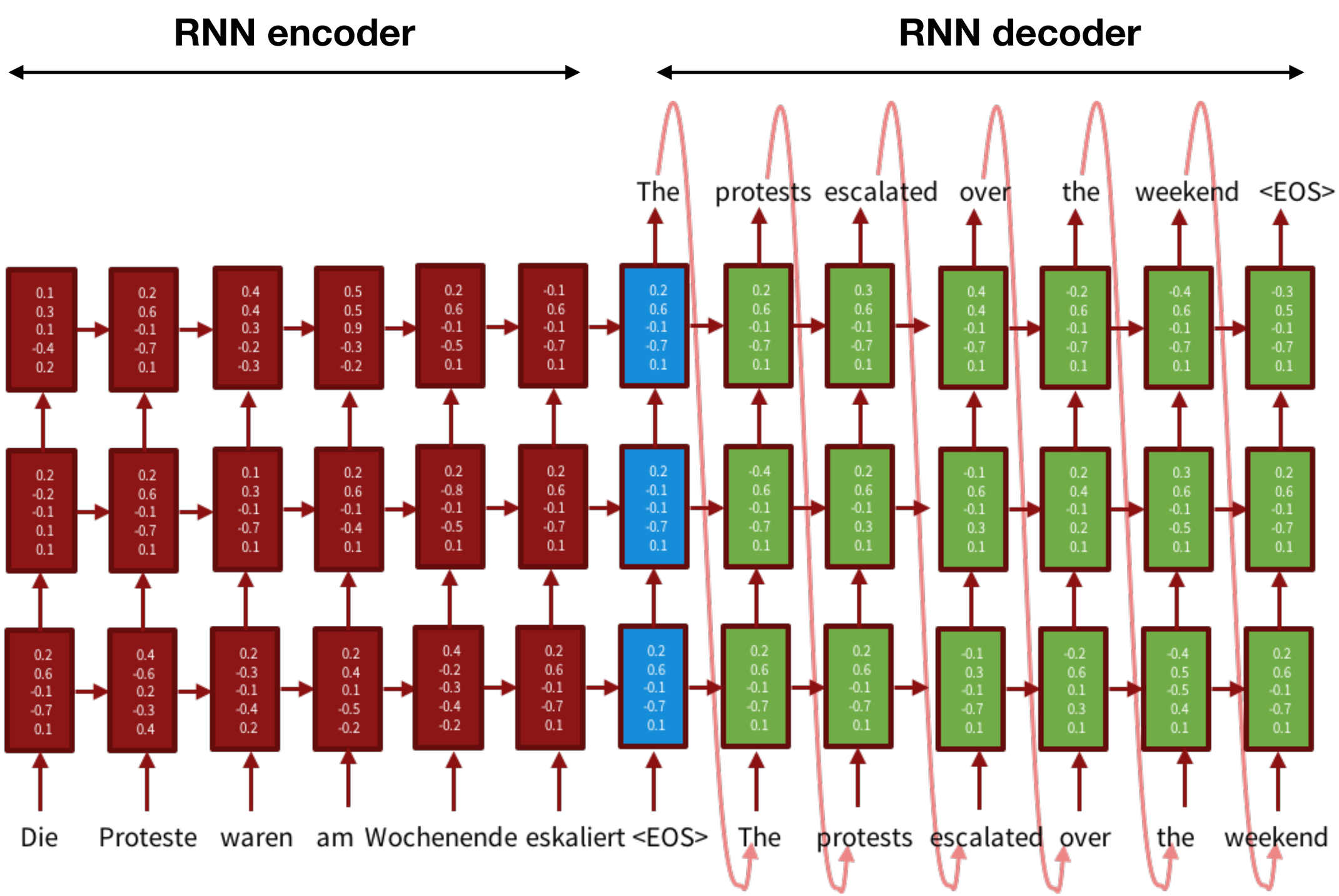
1. Auli, M., et al. "Joint Language and Translation Modeling with Recurrent Neural Networks." *EMNLP (2013)*
2. Kalchbrenner, N., et al. “Recurrent Continuous Translation Models.” *EMNLP (2013)*
3. Cho, K., et al. “Learning Phrase Representations using RNN Encoder-Decoder for Statistical MT.” *EMNLP (2014)*
4. Sutskever, I., et al. “Sequence to Sequence Learning with Neural Networks.” *NIPS (2014)*

Neural Machine Translation with Seq2Seq

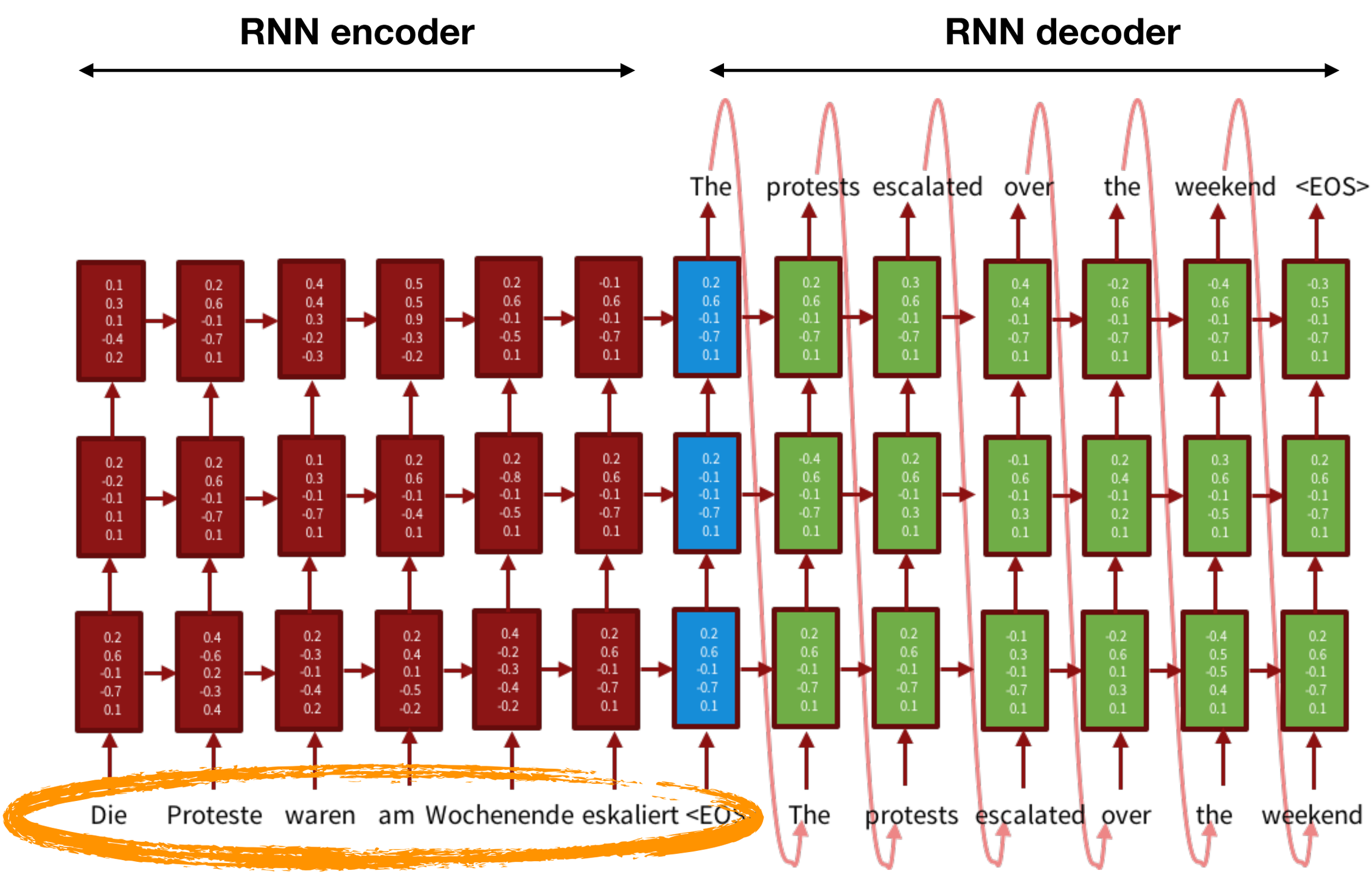


Word Embeddings!

1. Auli, M., et al. "Joint Language and Translation Modeling with Recurrent Neural Networks." *EMNLP (2013)*
2. Kalchbrenner, N., et al. “Recurrent Continuous Translation Models.” *EMNLP (2013)*
3. Cho, K., et al. “Learning Phrase Representations using RNN Encoder-Decoder for Statistical MT.” *EMNLP (2014)*
4. Sutskever, I., et al. “Sequence to Sequence Learning with Neural Networks.” *NIPS (2014)*



During training, the loss function is evaluated at the decoder by the prediction loss (cross entropy) of the next word given the current word and the decoder RNN state, always feeding the decoder RNN with the true word (teacher forcing).

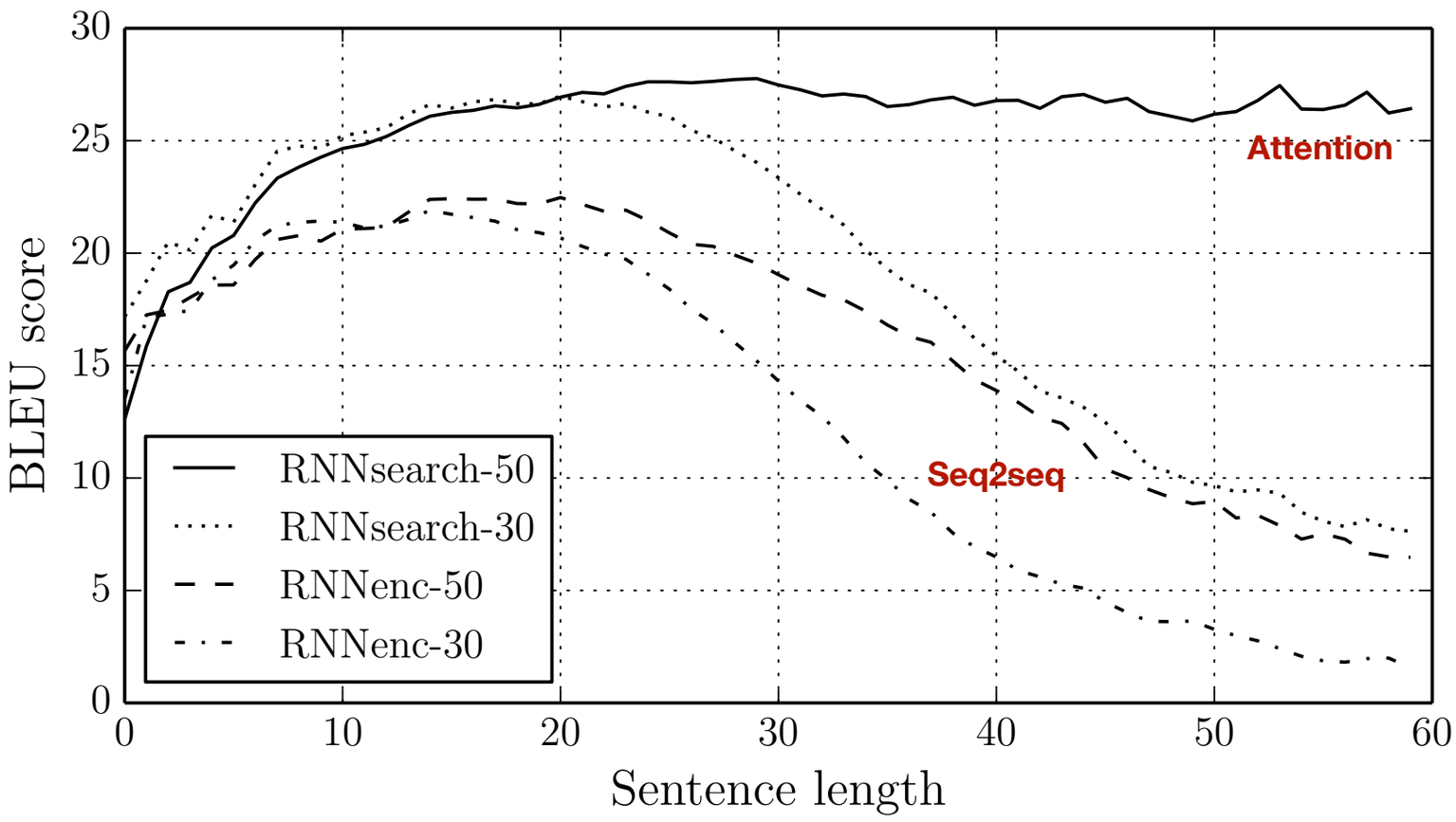


Word Embeddings!

During training, the loss function is evaluated at the decoder by the prediction loss (cross entropy) of the next word given the current word and the decoder RNN state, always feeding the decoder RNN with the true word (teacher forcing).

NMT with Seq2Seq: Limitations

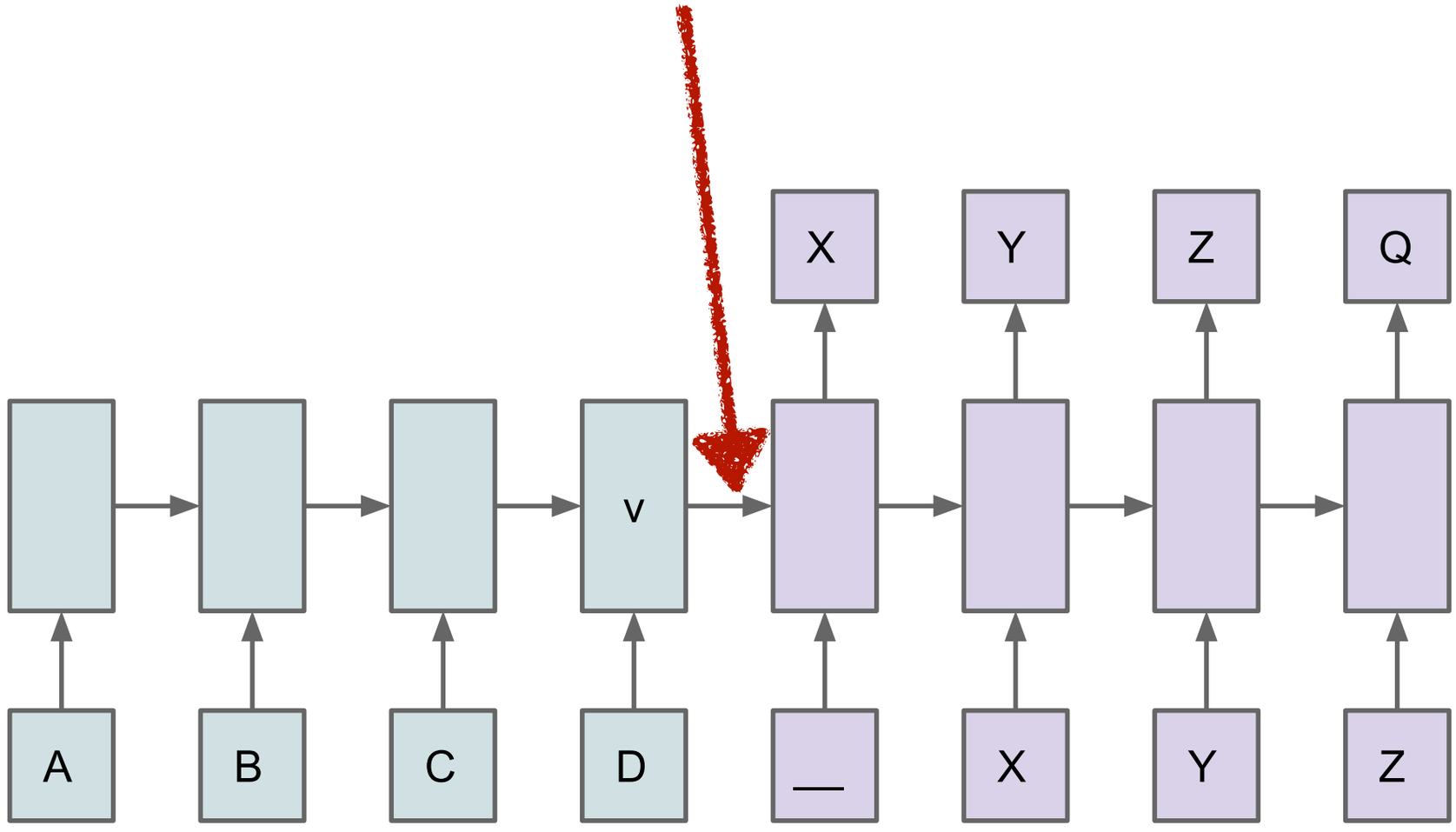
- Fixed Size Embeddings are easily overwhelmed by long inputs or long outputs
- BLEU (bilingual evaluation understudy) is an algorithm for evaluating the **quality of text which has been machine-translated** from one natural language to another.



1. Sutskever, I., et al. “Sequence to Sequence Learning with Neural Networks.” *NIPS (2014)*
2. Bahdanau, D., et al. “Neural Machine Translation by Jointly Learning to Align and Translate.” *ICLR (2015)*

Seq2Seq: The issue with long inputs

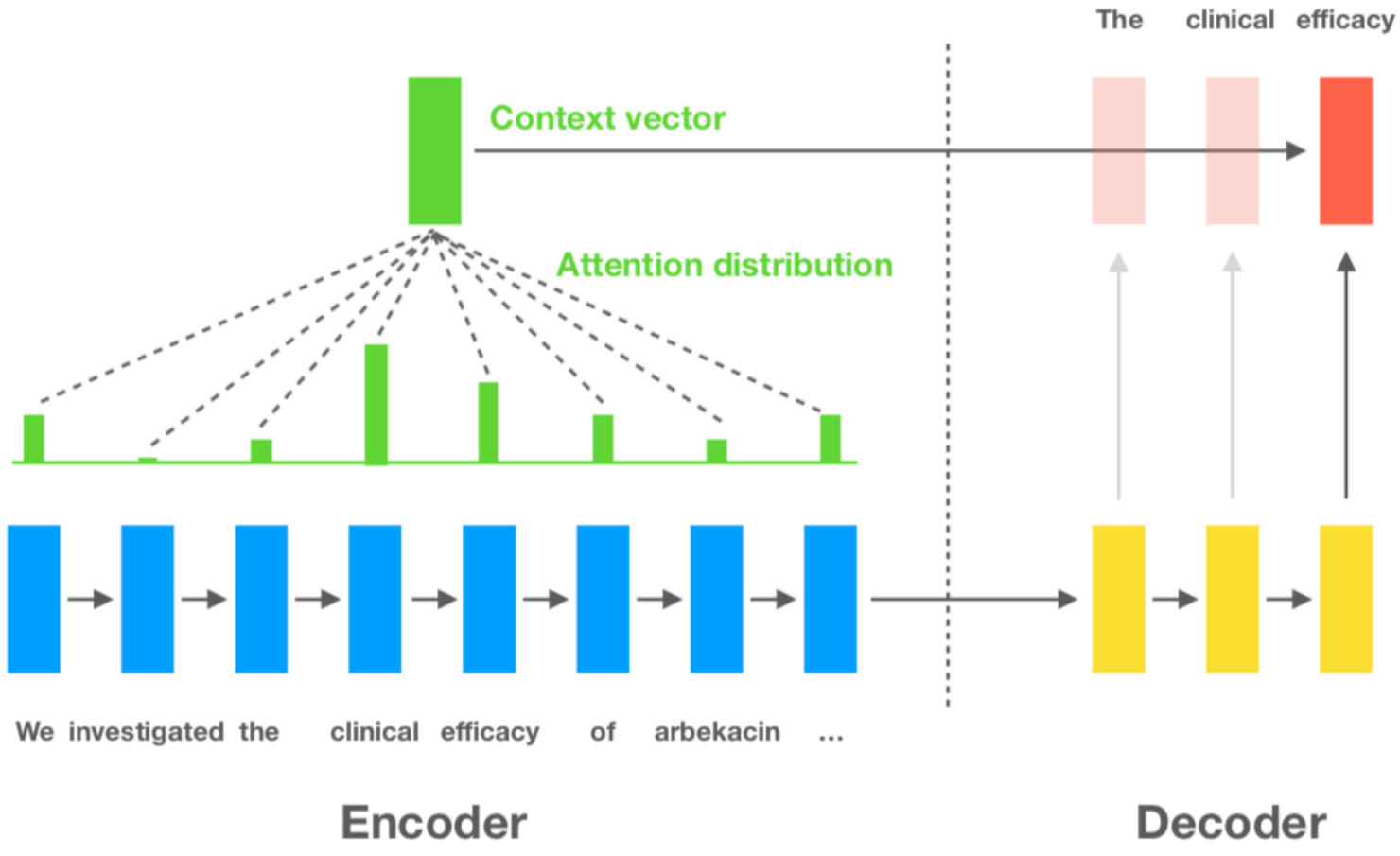
- Same embedding informs the entire output
- Needs to capture all the information about the input regardless of its length



Is there a better way to pass the information from encoder to the decoder ?

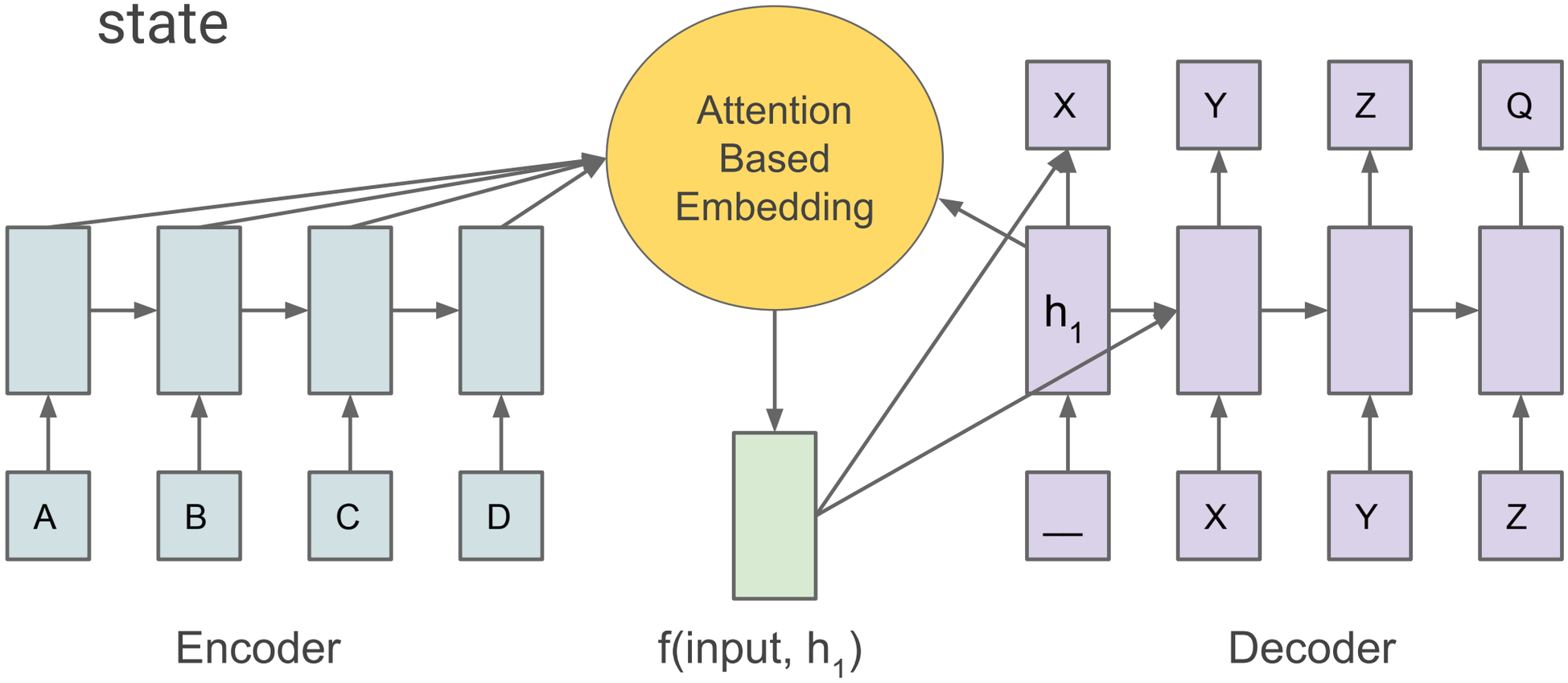
Seq2Seq with attention mechanisms

- Attention mechanisms have been shown to enhance the performance of seq2seq architectures **across almost all tasks**
- We allow the decoder to **“attend” to different parts of the source sentence** at each step of the output generation
- Each word that is generated by the decoder will be conditioned on a **unique weighted representation** of the source words



Seq2Seq - Attention models

- Embedding used to predict output, and compute next hidden state



- A different embedding computed for every output step

Seq2Seq with attention mechanisms

Published as a conference paper at ICLR 2015

NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE

Dzmitry Bahdanau
Jacobs University Bremen, Germany

KyungHyun Cho **Yoshua Bengio***
Université de Montréal

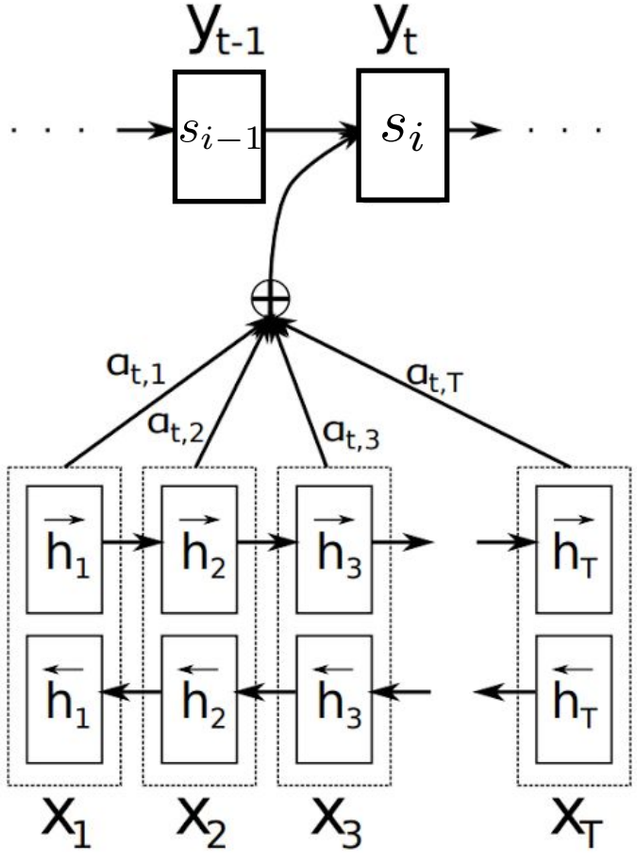
2015

Seq2Seq - Attention models

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

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

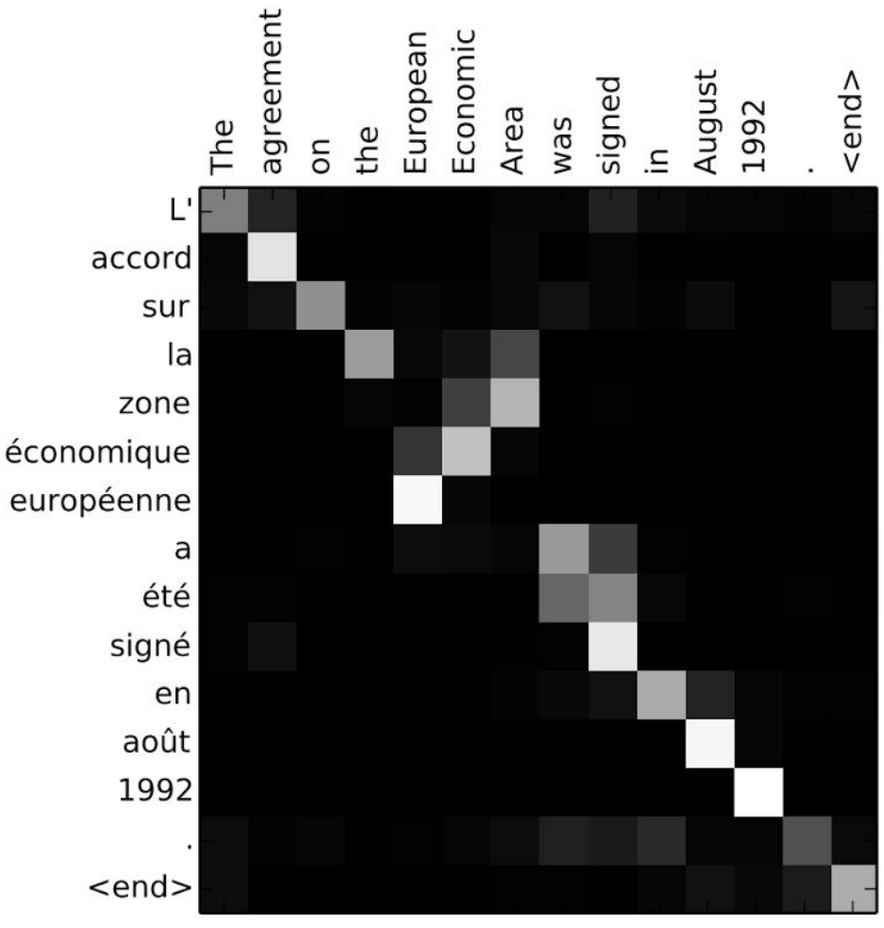
$$e_{ij} = a(s_{i-1}, h_j)$$



NEURAL MACHINE TRANSLATION
BY JOINTLY LEARNING TO ALIGN AND TRANSLATE

Dzmitry Bahdanau
Jacobs University Bremen, Germany

KyungHyun Cho **Yoshua Bengio***
Université de Montréal



Bahdanau, D., et al. “Neural Machine Translation by Jointly Learning to Align and Translate.” *ICLR (2015)*

Seq2Seq with attention mechanisms

- Large plethora of attention models
- Self-attention, two-way attention, key-value-predict models and hierarchical attention …
- Attention can also be used on the interface between a convolutional neural network and an RNN!

Seq2Seq with attention mechanisms

- Large plethora of attention models
- Self-attention, two-way attention, key-value-predict models and hierarchical attention …
- Attention can also be used on the interface between a convolutional neural network and an RNN!

Show, Attend and Tell: Neural Image Caption Generation with Visual Attention

---

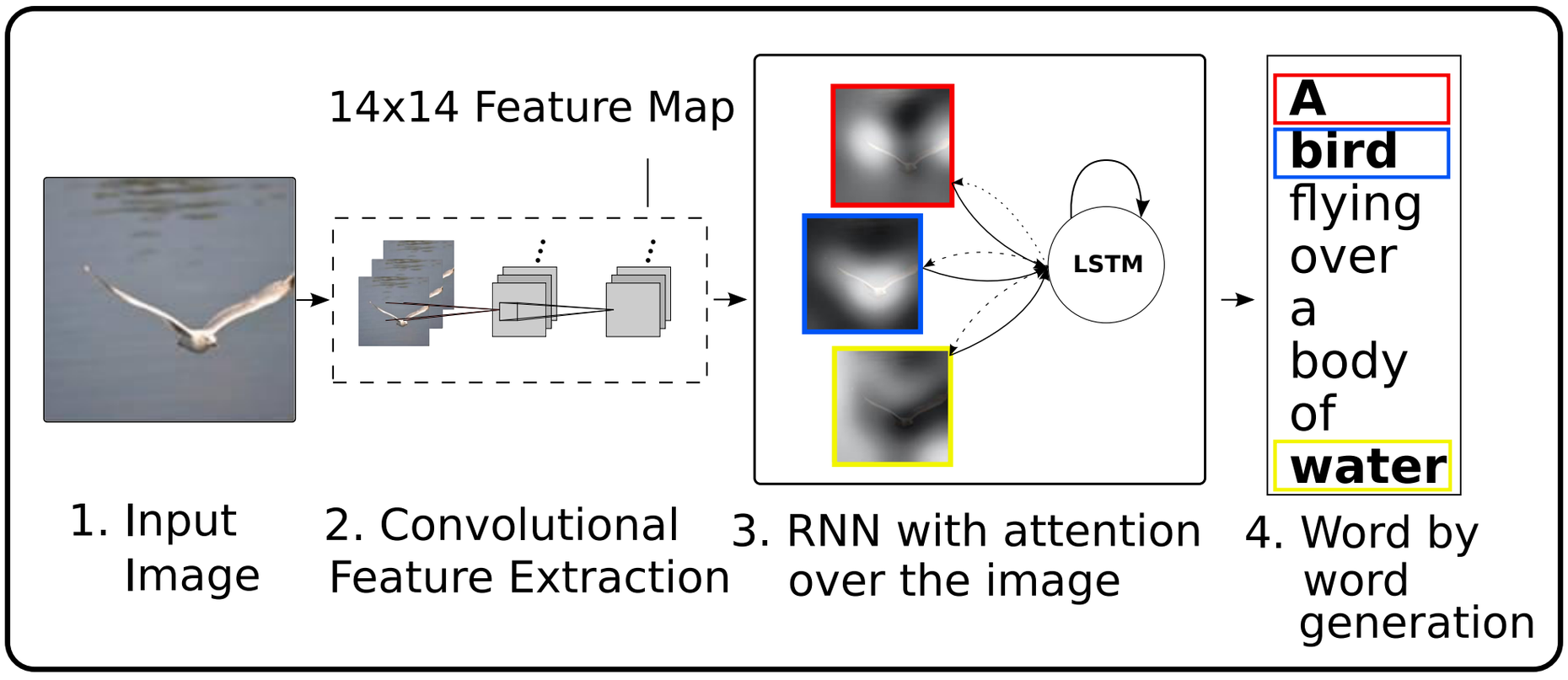
Kelvin Xu KELVIN.XU@UMONTREAL.CA
Jimmy Lei Ba JIMMY@PSI.UTORONTO.CA
Ryan Kiros RKIROS@CS.TORONTO.EDU
Kyunghyun Cho KYUNGHYUN.CHO@UMONTREAL.CA
Aaron Courville AARON.COURVILLE@UMONTREAL.CA
Ruslan Salakhutdinov RSALAKHU@CS.TORONTO.EDU
Richard S. Zemel ZEMEL@CS.TORONTO.EDU
Yoshua Bengio FIND-ME@THE.WEB

Seq2Seq with attention mechanisms

- Large plethora of attention models
- Self-attention, two-way attention, key-value-predict models and hierarchical attention …
- Attention can also be used on the interface between a convolutional neural network and an RNN!

Seq2Seq with attention mechanisms

- Large plethora of attention models
- Self-attention, two-way attention, key-value-predict models and hierarchical attention …
- Attention can also be used on the interface between a convolutional neural network and an RNN!



Seq2Seq with attention mechanisms

- Large plethora of attention models
- Self-attention, two-way attention, key-value-predict models and hierarchical attention …
- Attention can also be used on the interface between a convolutional neural network and an RNN!

Seq2Seq with attention mechanisms

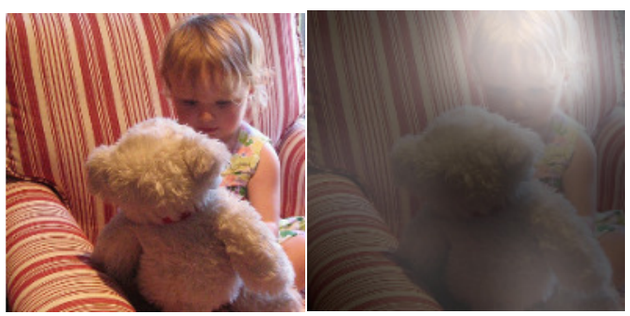
- Large plethora of attention models
- Self-attention, two-way attention, key-value-predict models and hierarchical attention …
- Attention can also be used on the interface between a convolutional neural network and an RNN!

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

A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.



A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.