University of Zurich[UZH]

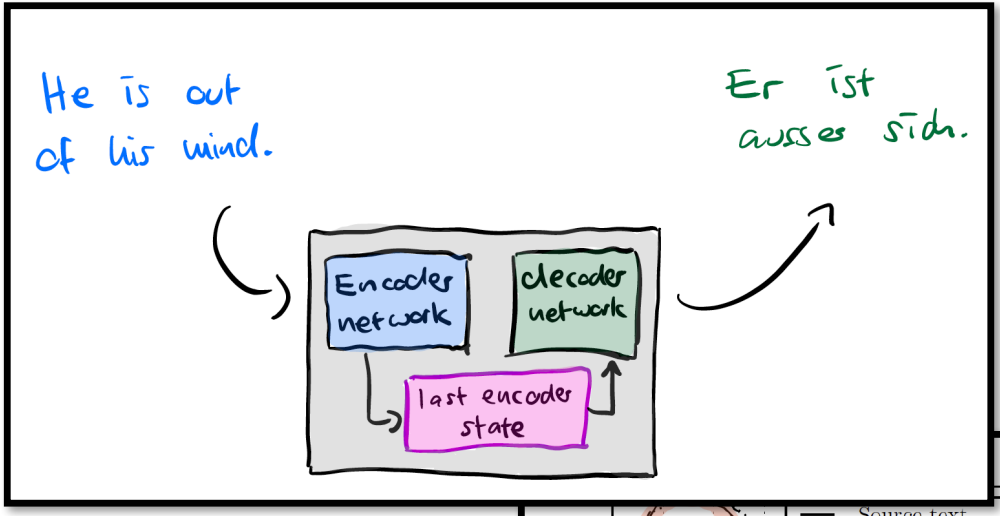
Institute of Computational Linguistics

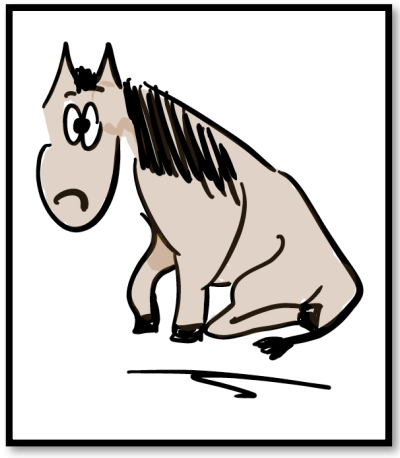
Machine Translation

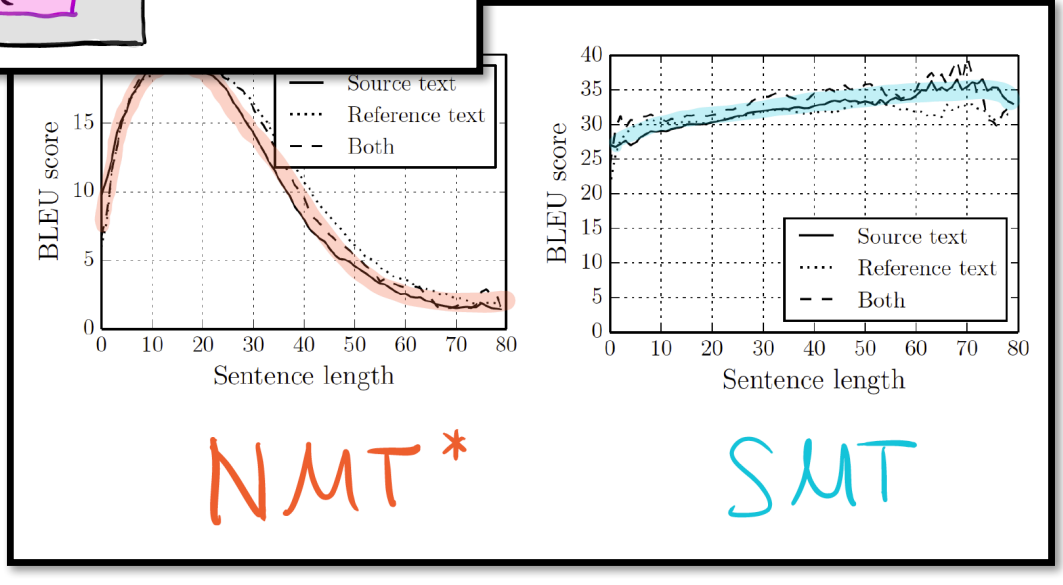
11 Attention
Bi-directional Encoding
Byte-pair Encoding

Mathias Müller

Last time


He is out of his mind.
Er ist ausser sich.
Encoder network
decoder network
last encoder state




BLEU score
Sentence length
Source text
Reference text
Both
NMT*
SMT

Topics of Today

- **Bi-directional encoding**: read source sequences in two directions

- **Attention models**: circumvent the problem of having to cram a %$! sentence into one %$! vector

- **Byte-pair encoding**: solve the problem of vocabulary size and unknown words

Bi-directional Encoding

Bi-directional encoding

- Bi-directional encoding is a change to the **encoder:**
  - one RNN reads the source sentence left-to-right
  - another RNN reads right-to-left

- Early research found that reversing the input sequence improves translation quality

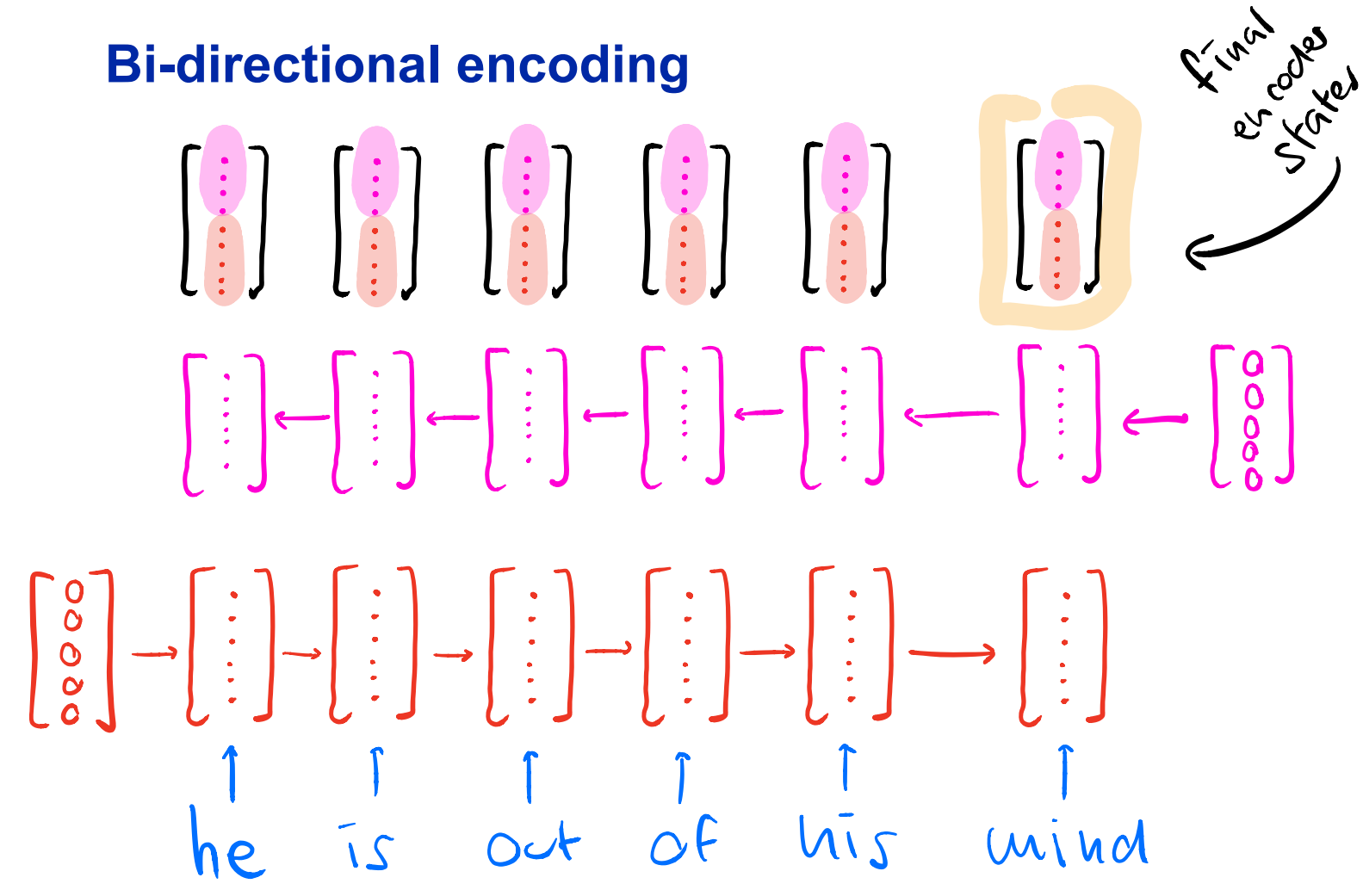
Why bi-directional encoding?

Early research found that **reversing the input sequence** improves translation quality

Surprisingly, the LSTM did not suffer on very long sentences, despite the recent experience of other researchers with related architectures [26]. We were able to do well on long sentences because we reversed the order of words in the source sentence but not the target sentences in the training and test set. By doing so, we introduced many short term dependencies that made the optimization problem much simpler (see sec. 2 and 3.3). As a result, SGD could learn LSTMs that had no trouble with long sentences. The simple trick of reversing the words in the source sentence is one of the key technical contributions of this work.

Sutskever et al (2014)

Bi-directional encoding



University of Zurich[UZH]

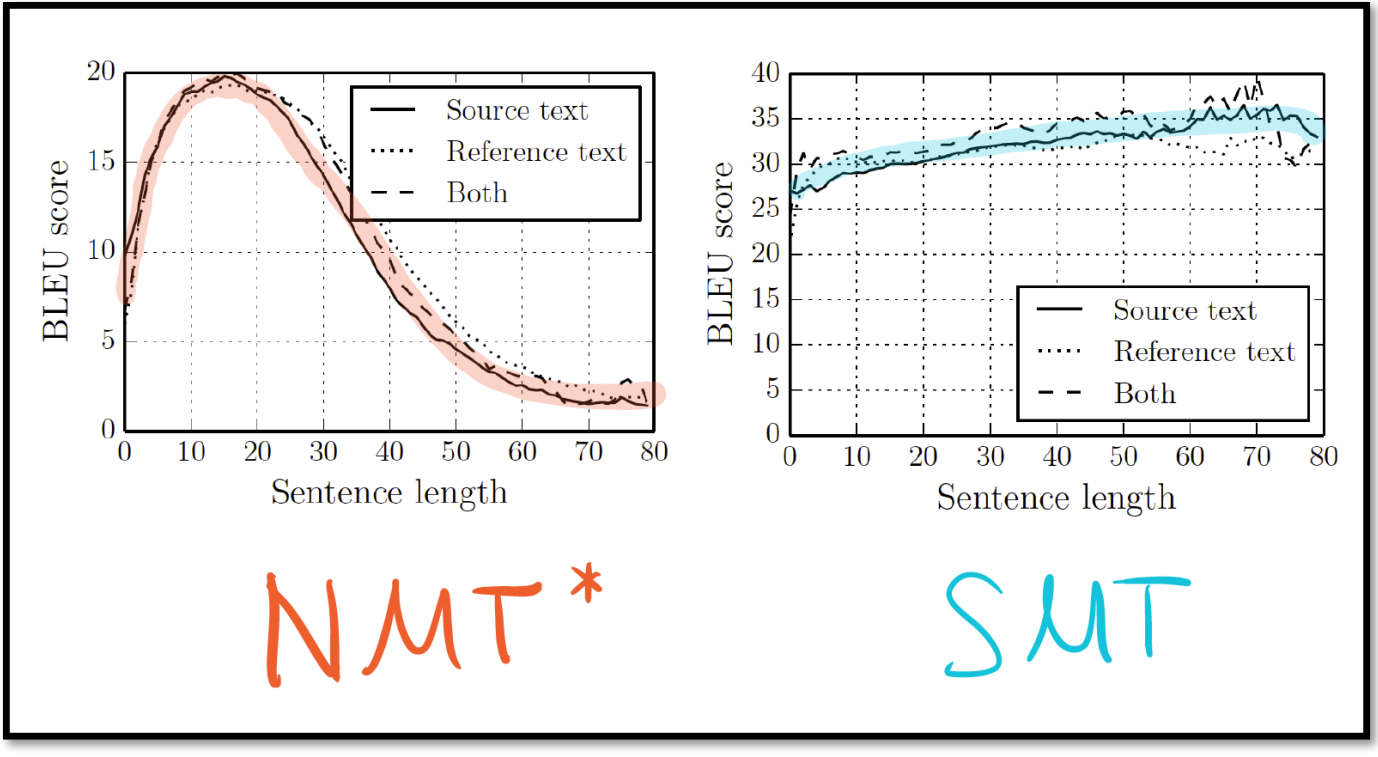
Institute of Computational Linguistics

Attention Networks

Attention networks

- Address the problem of **long sentences**
- Change to the decoder architecture
- Intuitively: instead of computing a "summary" of the source sentence once (last encoder state), compute it again at each decoding step

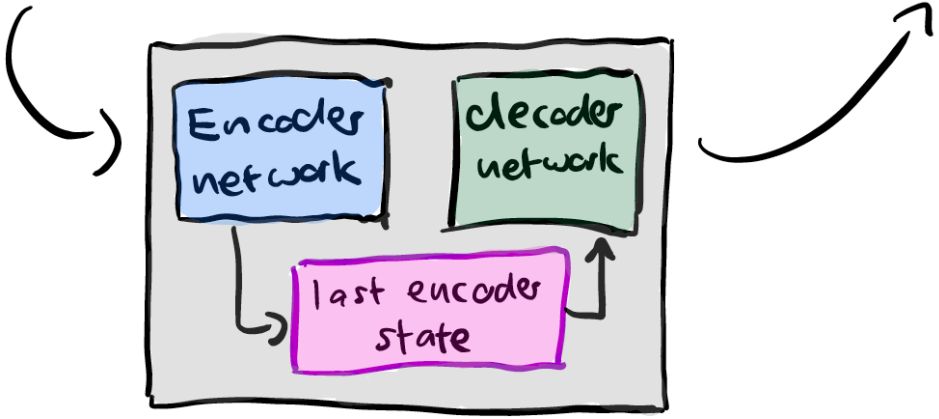
Long sentences lead to low translation quality



Reason for this problem: bottleneck between encoder and decoder

He is out of his mind.

Er ist ausser sich.



Visual attention in image captioning

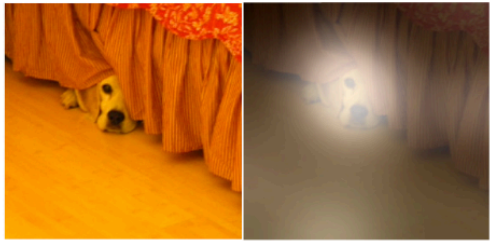
Attention network

RNN

A woman is throwing a frisbee

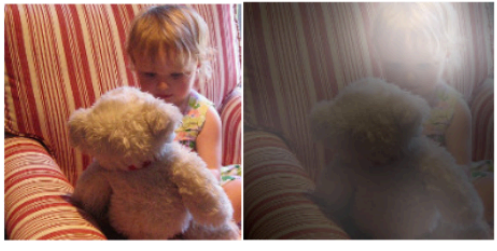
Visual attention in image captioning

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.

Xu et al (2015)

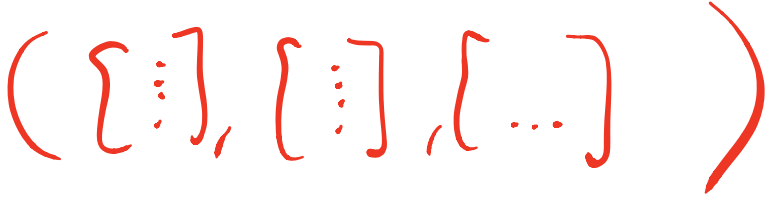
Textual attention

Eine Frau wirft einen Frisbee in einem Park

A woman is throwing a **frisbee** in a park.

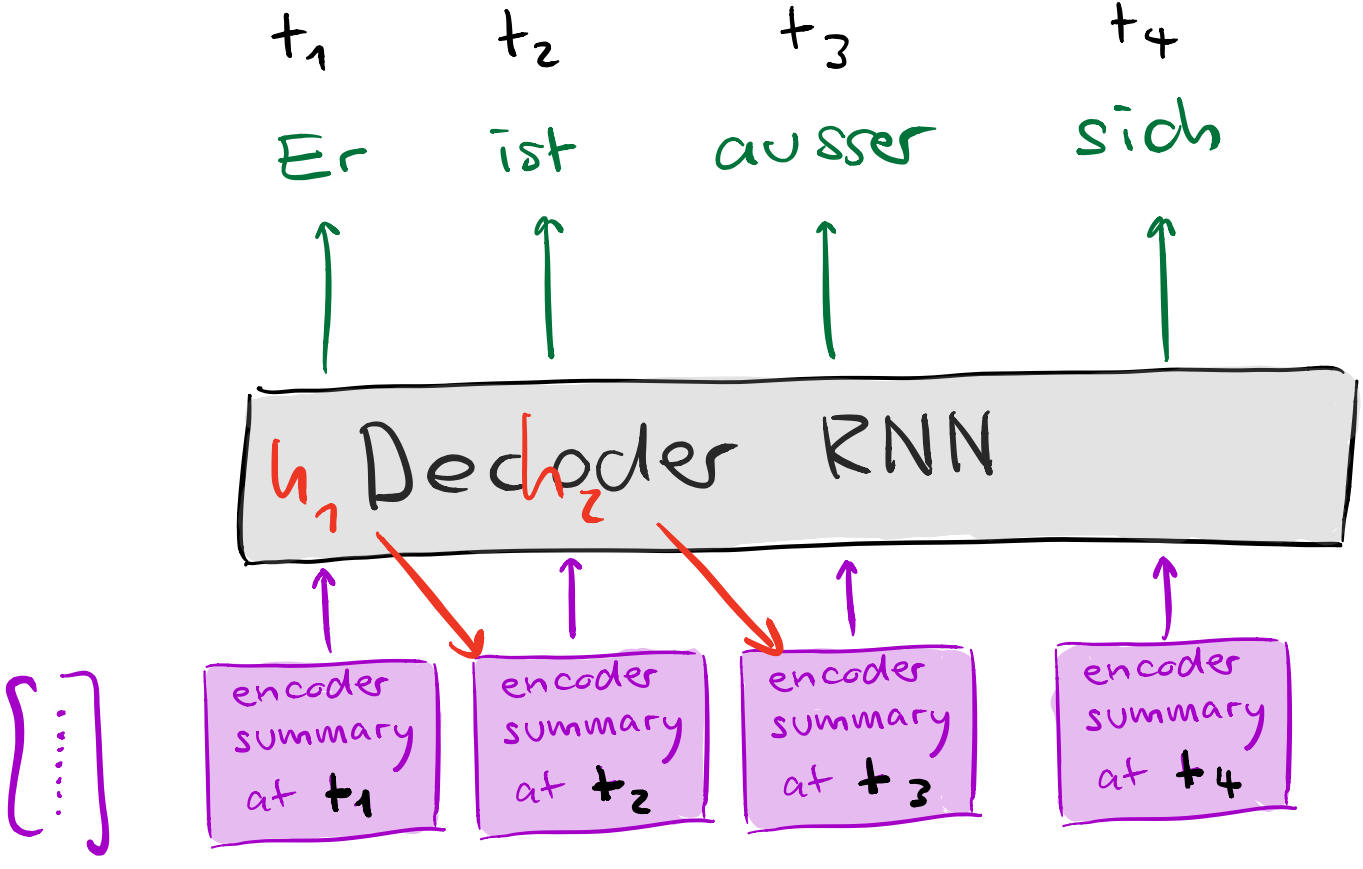
Attention

- Instead of only seeing the final encoder state, the decoder is **allowed to see all encoder states**

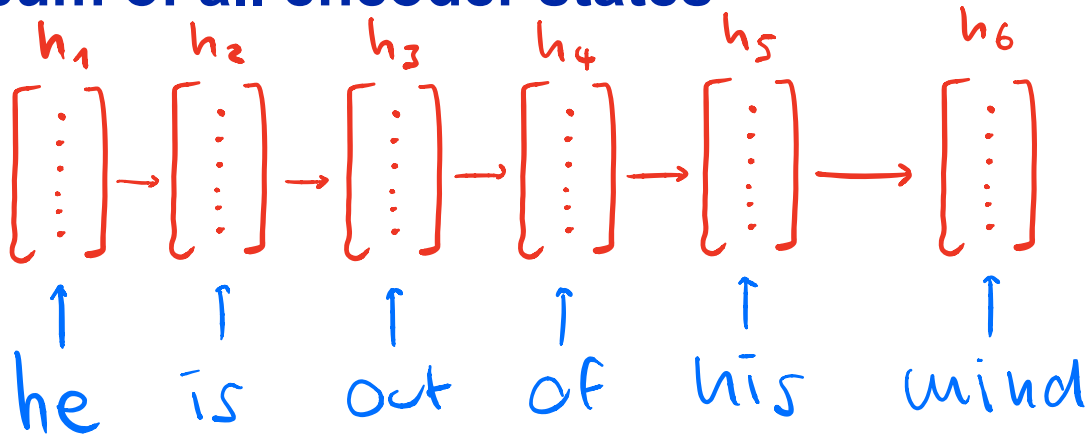


- At every time step, the decoder is fed an **additional input that is a weighted sum of all encoder states**

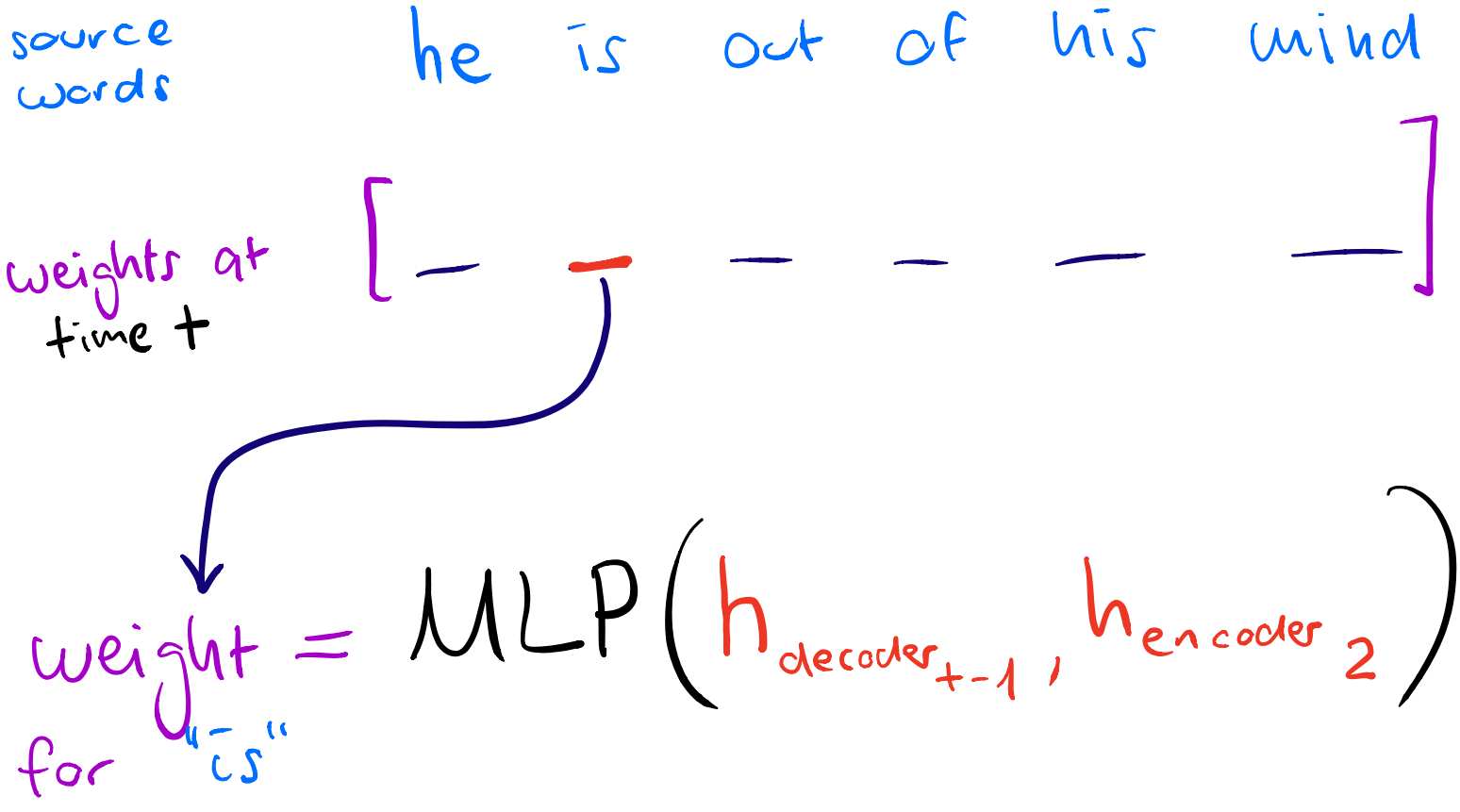
Weighted sum of all encoder states



Weighted sum of all encoder states



How attention weights are computed



How attention weights are computed

weights predicted by MLP

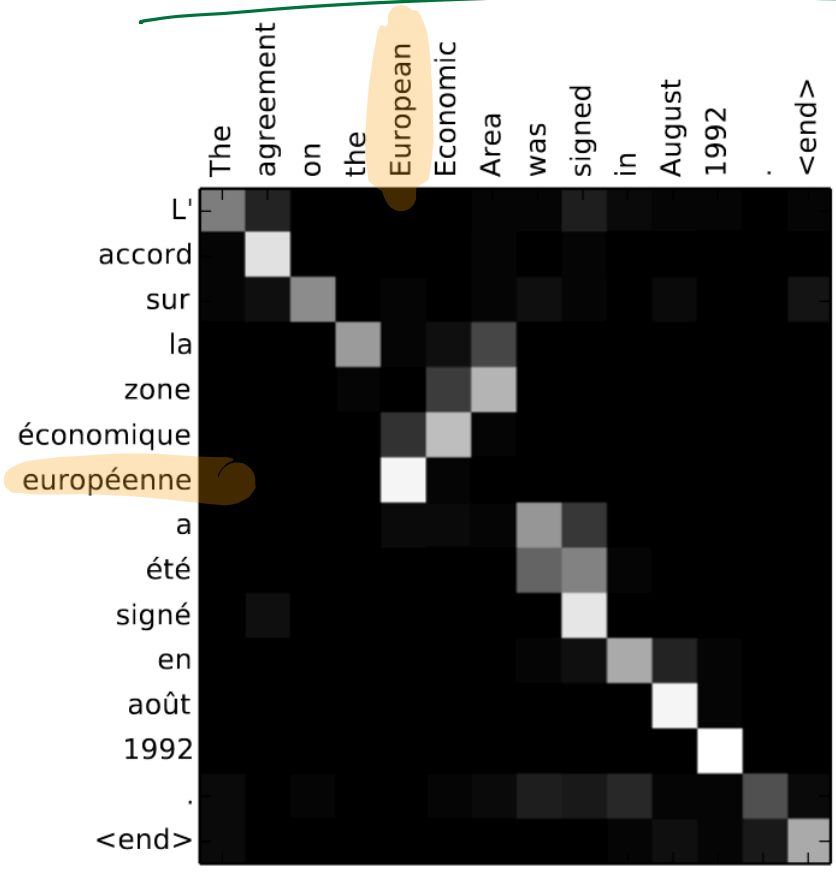
$$[\,1.1 \quad 1.7 \quad -0.2 \quad -0.2 \quad -0.01 \quad -0.01\,]$$

+ softmax

$$[\,0.2 \quad 0.5 \quad 0.1 \quad 0.1 \quad 0.05 \quad 0.05\,]$$

new decoder state h_t $= \mathrm{RNN}\left(h_{t-1}, y_{t-1}, \boxed{\text{encoder summary at } t}\right)$

Visualization of attention weights



Attention weights are not word alignments!

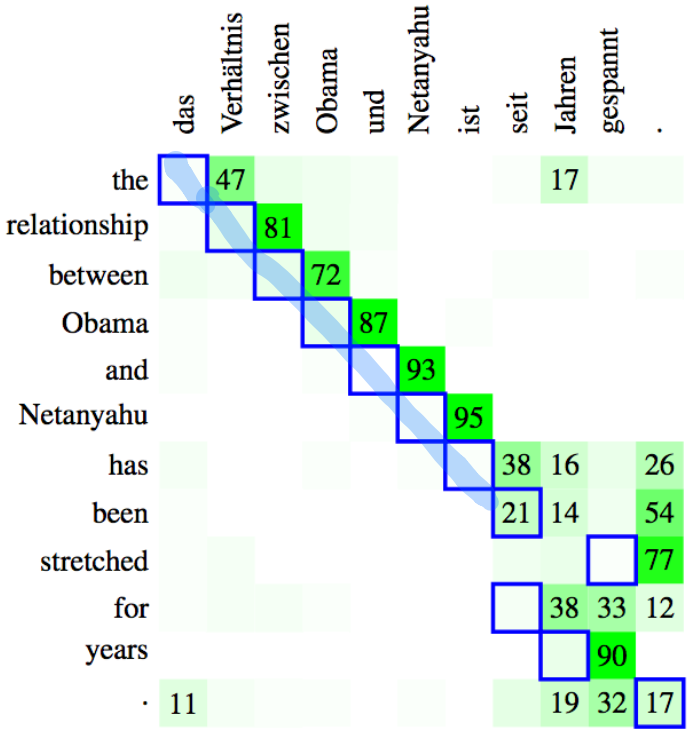


Figure 9: Mismatch between attention states and desired word alignments (German–English).

Impact of attention on NMT translation quality

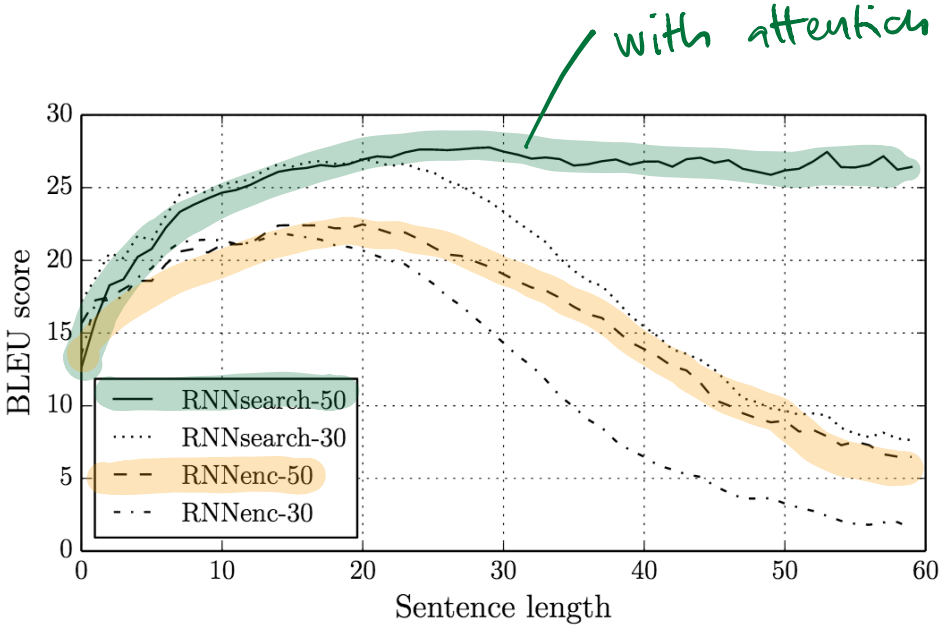


Figure 2: The BLEU scores of the generated translations on the test set with respect to the lengths of the sentences. The results are on the full test set which includes sentences having unknown words to the models.

After a usual tokenization[7], we use a shortlist of 30,000 most frequent words in each language to train our models. Any word not included in the shortlist is mapped to a special token ([UNK]). We do not apply any other special preprocessing, such as lowercasing or stemming, to the data.

Summary Attention

- Change to decoder to address bad translations for long sentences
- Intuitively: decoder is able to look at all encoder states, instead of just the last one
- Technically: at each decoding step, an attention network computes weights that are used to compute a weighted sum of all encoder states

Byte-pair Encoding (BPE)

Byte-pair encoding (BPE)

- Addresses the problem of vocabulary size and unknown words

<unk>

- BPE is a **segmentation algorithm**: it **segments words** into smaller pieces

BPE Algorithm

For training.

"d", "de", "denlcen"

Initial vocabulary: all individual characters are **symbols** in the vocabulary

Until vocabulary has size **N**:

- Look for most frequent symbols bigram (**s1**, **s2**) in the training data

("d", "e")

- Add the concatenation of **s1** and **s2** as new symbol to the vocabulary

V = V + "de"

BPE Example

Training data:

The methane lane is sane.
Sane is the methane lane.

Represent data as characters:

T h e </w> m e t h a n e </w> l a n e </w> i s </w> s a n e .

S a n e </w> l s </w> t h e </w> m e t h a n e </w> l a n e.

BPE Example

T h e </w> m e t h a n e </w> l a n e </w> i s </w> s a n e .

S a n e </w> l s </w> t h e </w> m e t h a n e </w> l a n e.

Initial vocabulary: all characters

$V = \{T, h, e, </w>, m, a, n, l, i, s, S, .\}$ + $|V| = 13$

Adding a symbol

T h e </w> m e t h a n e </w> l a n e </w> i s </w> s a n e .

S a n e </w> l s </w> t h e </w> m e t h a n e </w> l a n e.

Which symbol bigram is most frequent?

("a", "n") ⟶ "an"

Add this bigram as new symbol to the vocab:

V = {T, h, e, </w>, m, a, n, l, i, s, S, ., t, an}

Update representation:

T h e </w> m e t h **an** e </w> l **an** e </w> i s </w> s **an** e .

S **an** e </w> l s </w> t h e </w> m e t h **an** e </w> l **an** e.

Until vocabulary has desired size:

- Which symbol bigram is most frequent?
- Add this bigram as new symbol to the vocab

After adding 5 extra symbols

T h e </w> me thane </w> l ane </w> i s </w> s an e .

S ane </w> i s </w> th e </w> me thane </w> l an e .

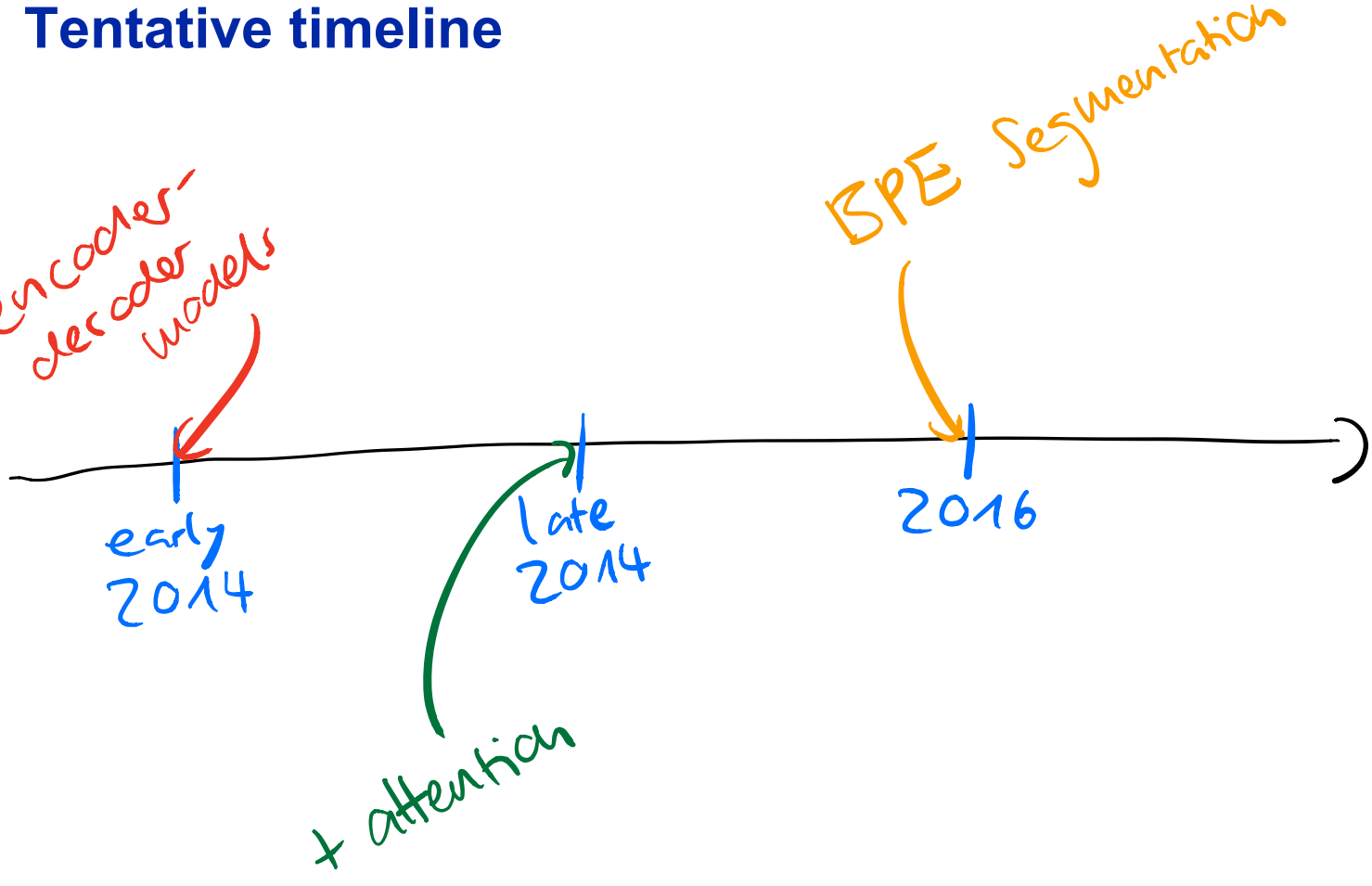
Merged symbols: **Vocabulary size =**

a n
an e</w>
t h
th ane</w>
m e

BPE Encoding

- BPE as a pre-processing step used in all current NMT systems, essential!
- Common vocabulary sizes: 10k - 50k, depending on data set

Tentative timeline



NMT state-of-the-art since ~ 2016

system	BLEU	official rank
uedin-nmt	34.2	1
metamind	32.3	2
uedin-syntax	30.6	3
NYU-UMontreal	30.8	4
online-B	29.4	5-10
KIT/LIMSI	29.1	5-10
cambridge	30.6	5-10
online-A	29.9	5-10
promt-rule	23.4	5-10
KIT	29.0	6-10
jhu-syntax	26.6	11-12
jhu-pbmt	28.3	11-12
uedin-pbmt	28.4	13-14
online-F	19.3	13-15
online-G	23.8	14-15

WMT16 EN→DE

system	BLEU	official rank
uedin-nmt	38.6	1
online-B	35.0	2-5
online-A	32.8	2-5
uedin-syntax	34.4	2-5
KIT	33.9	2-6
uedin-pbmt	35.1	5-7
jhu-pbmt	34.5	6-7
online-G	30.1	8
jhu-syntax	31.0	9
online-F	20.2	10

WMT16 DE→EN

- pure NMT

Summary Overall

- **Bi-directional encoding**: read source sequence from both sides, with two separate encoder RNNs

- **Attention networks:** at each decoding step, use attention weights to generate a new summary of the input sentence

- **BPE:** segment words into subwords to control vocabulary size and avoid unknown words

Further Reading / links

- Ilustrations by Jay Alammar:
  http://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/
- Influential paper 1:
  Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2014). "Neural machine translation by jointly learning to align and translate."
- Influential paper 2:
  Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le (2014). "Sequence to sequence learning with neural networks."
- BPE Implementation Repo:
  https://github.com/rsennrich/subword-nmt
- BPE Paper by Rico and Barry Haddow:
  Rico Sennrich, Barry Haddow and Alexandra Birch (2016): "Neural Machine Translation of Rare Words with Subword Units."
- Good tutorial that shows an implementation in TF:
  https://github.com/tensorflow/tensorflow/blob/r1.13/tensorflow/contrib/eager/python/examples/nmt_with_attention/nmt_with_attention.ipynb

Next time

	Cloud Platform		
30.04.	Encoder-Decoder-Modell	NMT Kapitel 5	Übung 5
07.05.	Attention-Mechanismus, bidirektionales Encoding, Byte Pair Encoding	NMT Kapitel 5-6	
14.05.	Decoding-Strategien	NMT Kapitel 5.4	Übung 6
21.05.	Maschinelle Übersetzung in der Praxis (Anwendungen)		
28.05.	Zusammenfassung, Q&A Prüfung		
Eventuell: Gastvortrag Prof. Artem Sokolov			
Cancelled! Prof. Sokolov had to decline the invitation.			
Prüfung (schriftlich)			
18.06., AND-2-48, 16.15 bis 18:00 Uhr			

Advance notice: exam questions

- On May 28, we will have an exam Q&A
- Until May 28, please post on OLAT:

  Exam question that would be fair in your opinion

- We will discuss exactly those questions that day.