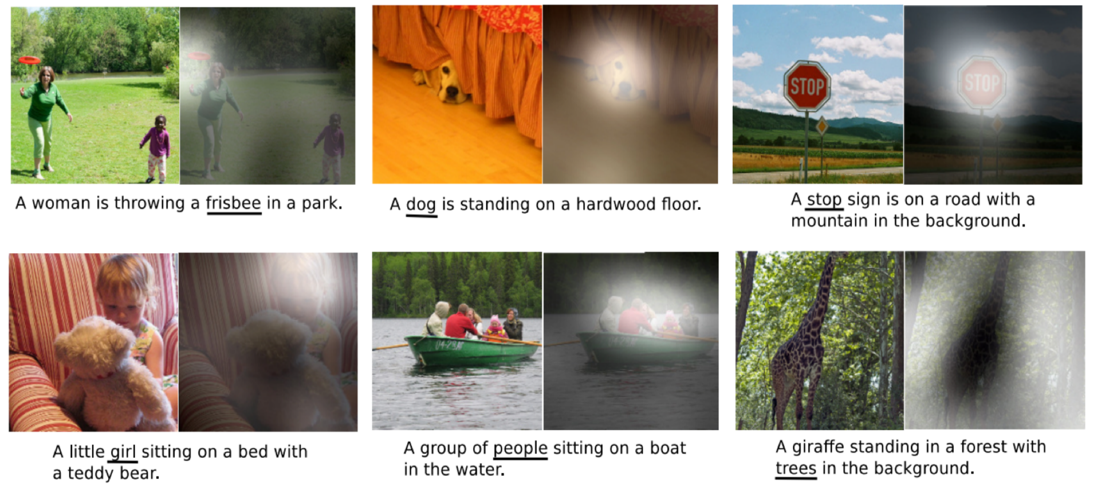
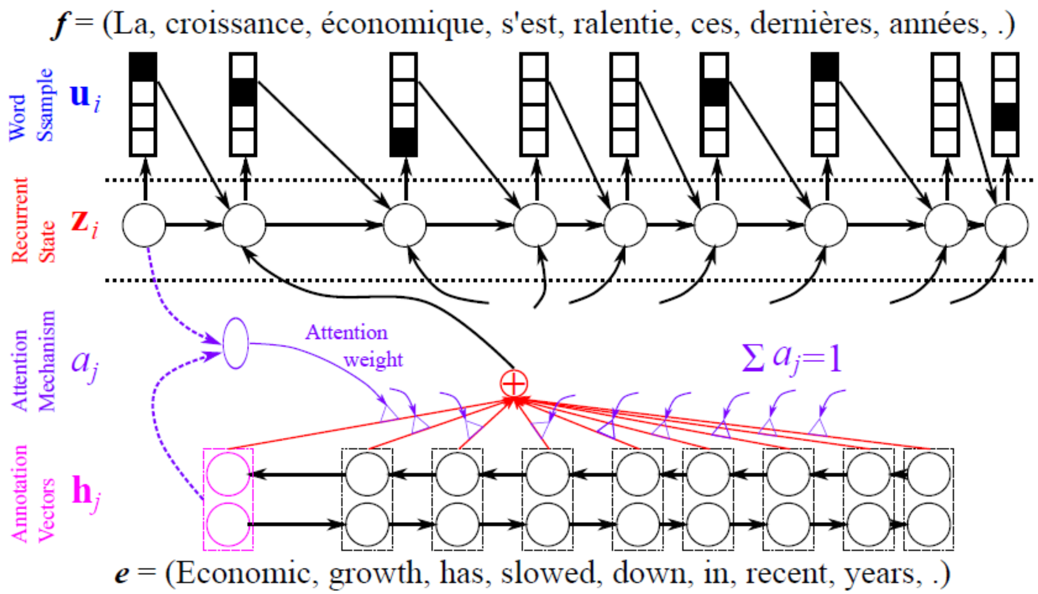
Sequence-to-sequence models with attention



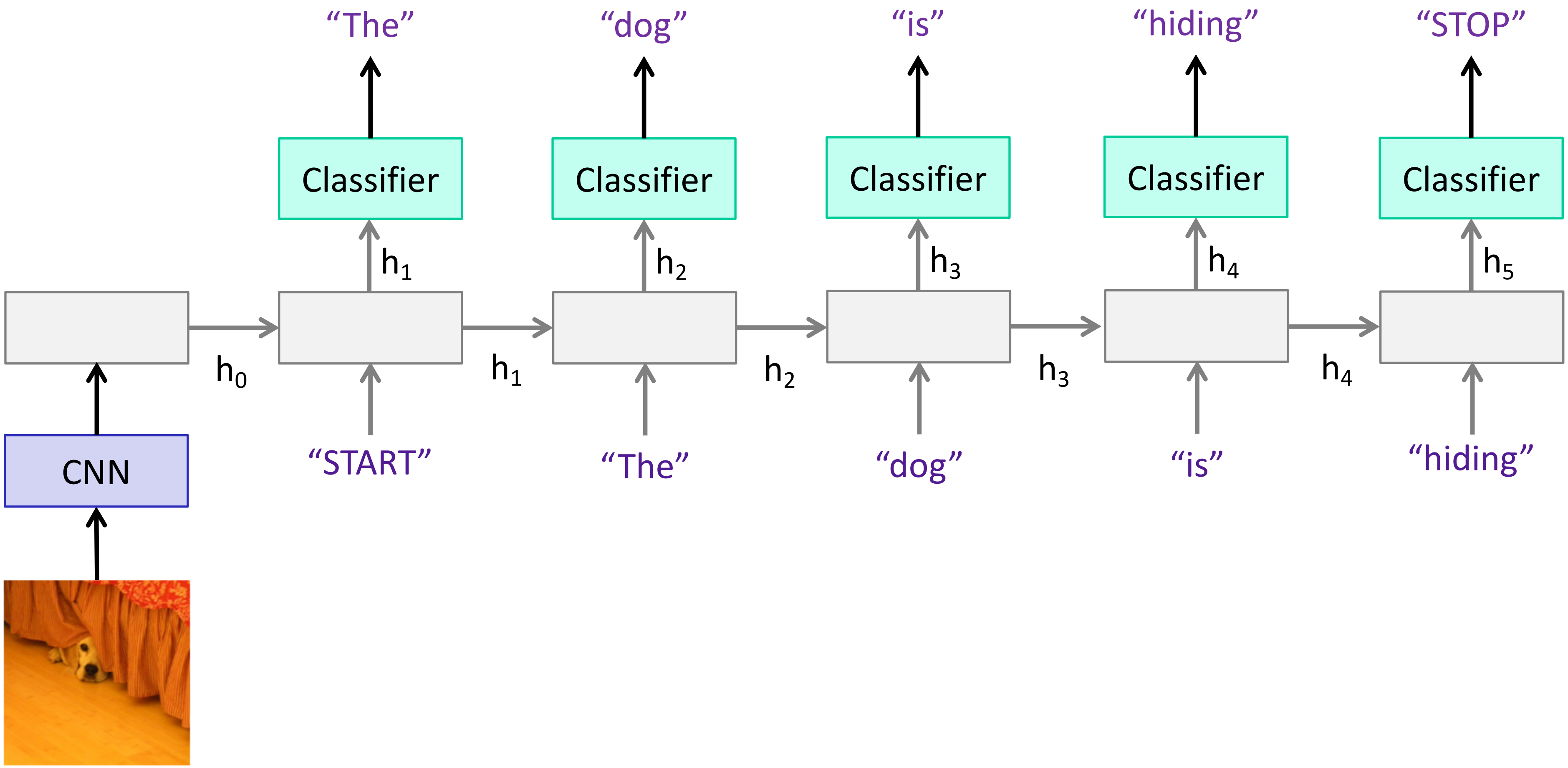


Slides L. Lazebnik and others

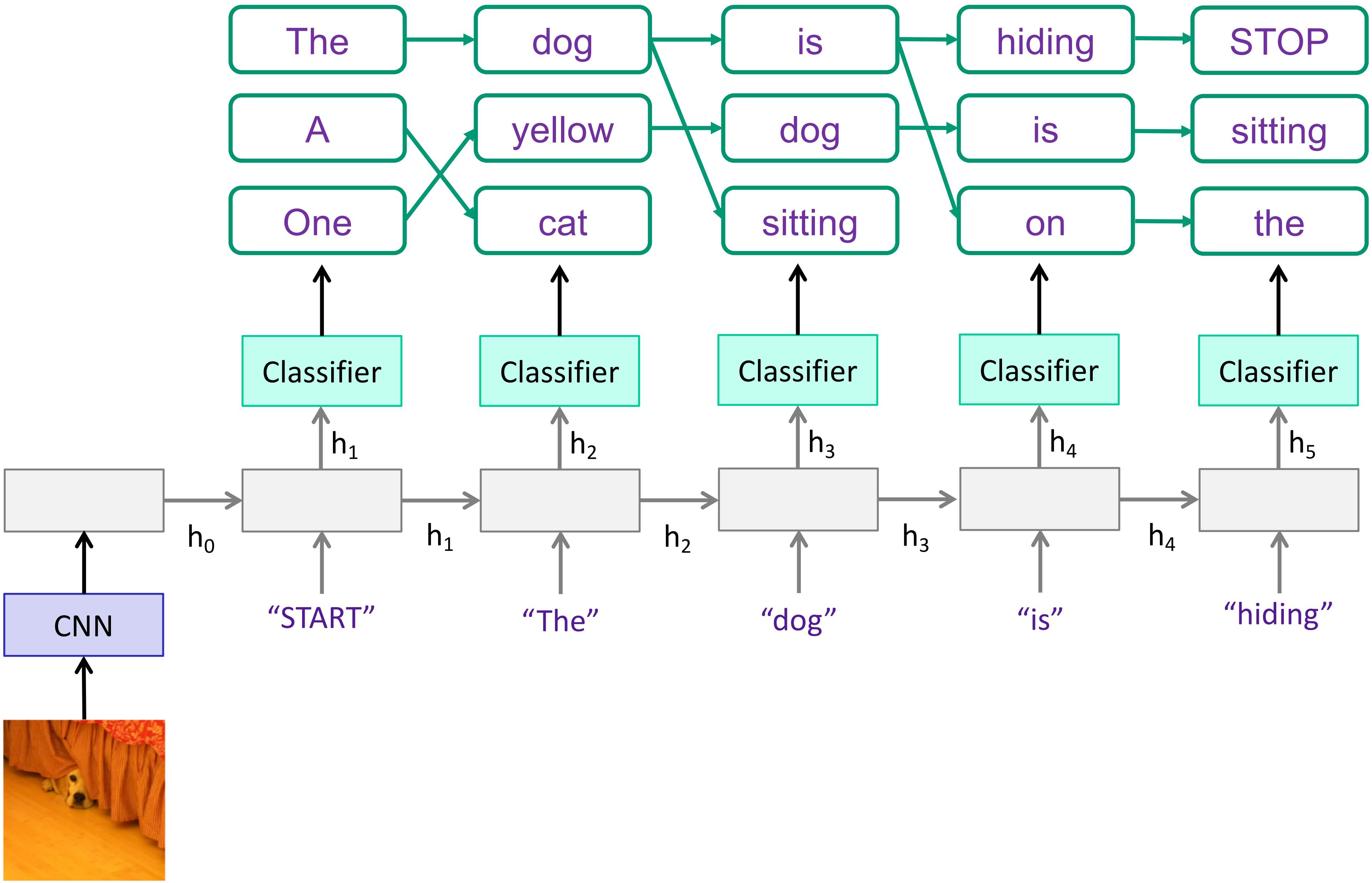
Overview

- Image captioning with attention
- Neural machine translation with attention
  - Recurrent models with global and local attention
  - Google Neural Machine Translation
  - Convolutional sequence to sequence models
  - Attention without recurrence or convolutions

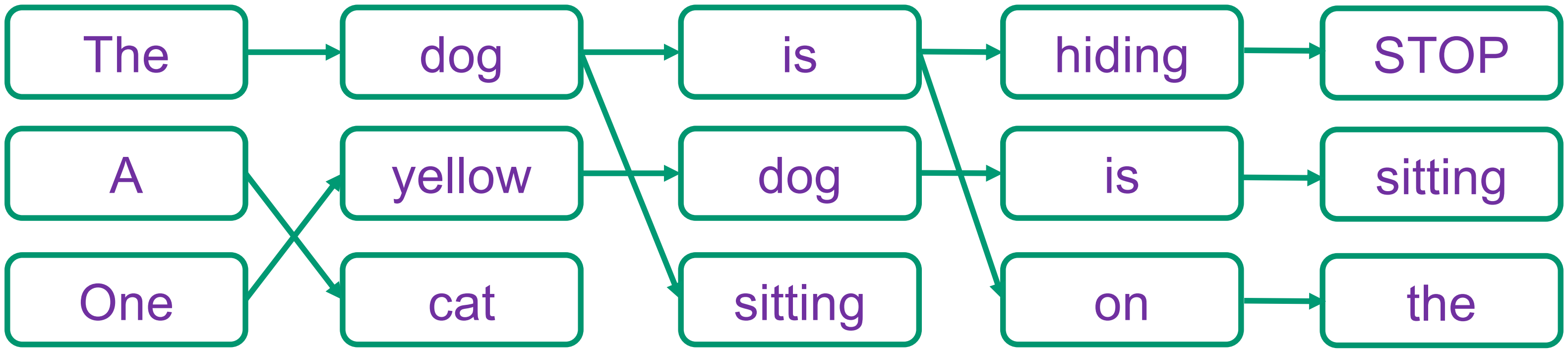
Review: Image captioning



Review: Image captioning



Beam search



- Maintain *k* top-scoring candidate sentences (according to sum of per-word log-likelihoods)
  - At each step, generate all their successors and reduce to *k* (*beam width*)

How to evaluate image captioning?

Reference sentences (written by human annotators):

- “A dog hides underneath a bed with its face peeking out of the bed skirt”
- “The small white dog is peeking out from under the bed”
- “A dog is peeking its head out from underneath a bed skirt”
- “A dog peeking out from under a bed”
- “A dog that is under a bed on the floor”

Generated sentence:

- “A dog is hiding”

BLEU: Bilingual Evaluation Understudy

- **N-gram precision**: count the number of n-gram matches between candidate and reference translation, divide by total number of n-grams in candidate translation
  - Clip counts by the maximum number of times an n-gram occurs in any reference translation
  - Multiply by *brevity penalty* to penalize short translations
- Most commonly used measure despite well-known shortcomings

K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, [BLEU: a Method for Automatic Evaluation of Machine Translation](), ACL 2002

Microsoft
Common Objects in Context

cocodataset@outlook.com

Home People Explore Dataset

Overview Challenges Download Evaluate Leaderboard

Table-C5 Table-C40 2015 Captioning Challenge

Last update: June 8, 2015. Visit CodaLab for the latest results.

	CIDEr-D	Meteor	ROUGE-L	BLEU-1	BLEU-2	BLEU-3	BLEU-4
m-RNN (Baidu/ UCLA)[16]	0.886	0.238	0.524	0.72	0.553	0.41	0.302
m-RNN[15]							0.299
MSR Captiva							.308
Google[4]							.309
Berkeley LR(							.277
Nearest Neig							.28
MSR[8]							.291
Montreal/Toronto[10]	0.85	0.243	0.513	0.689	0.515	0.372	0.268
PicSOM[13]	0.833	0.231	0.505	0.683	0.51	0.377	0.281
Tsinghua Bigeye[14]	0.673	0.207	0.49	0.671	0.494	0.35	0.241
MLBL[7]	0.74	0.219	0.499	0.666	0.498	0.362	0.26
Human[5]	0.854	0.252	0.484	0.663	0.469	0.321	0.217

Metrics

CIDEr-D	CIDEr: Consensus-based Image Description Evaluation
METEOR	Meteor Universal: Language Specific Translation Evaluation for Any Target Language
Rouge-L	ROUGE: A Package for Automatic Evaluation of Summaries
BLEU	BLEU: a Method for Automatic Evaluation of Machine Translation

http://mscoco.org/dataset/#captions-leaderboard

Microsoft
Common Objects in Context

cocodataset@outlook.com

Home People Explore **Dataset**

Overview | Challenges | Download | Evaluate | Leaderboard

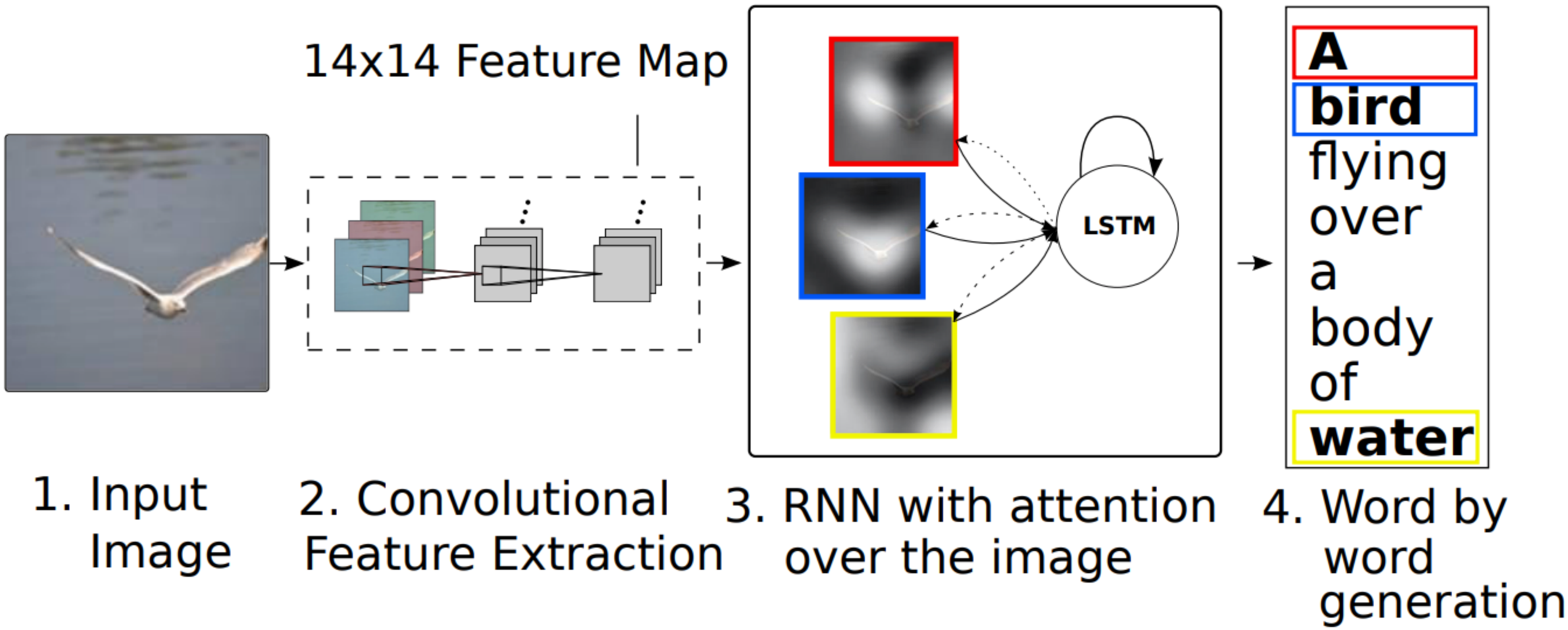
Table-C5 | Table-C40 | **2015 Captioning Challenge**

Last update: June 8, 2015. Visit CodaLab for the latest results.

	M1	M2	M3	M4	M5
Human[5]	0.638	0.675	4.836	3.428	0.352
Google[4]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
MSR[8]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Montreal	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
MSR Ca	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Berkeley	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
m-RNN[1	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Nearest Neighbor[11]	0.216	0.255	3.801	2.716	0.196
PicSOM[13]	0.202	0.250	3.965	2.552	0.182
Brno University[3]	0.194	0.213	3.079	3.482	0.154
m-RNN (Baidu/ UCLA)[16]	0.190	0.241	3.831	2.548	0.195
MIL[6]	0.168	0.197	3.349	2.915	0.159
MLBL[7]	0.167	0.196	3.659	2.420	0.156

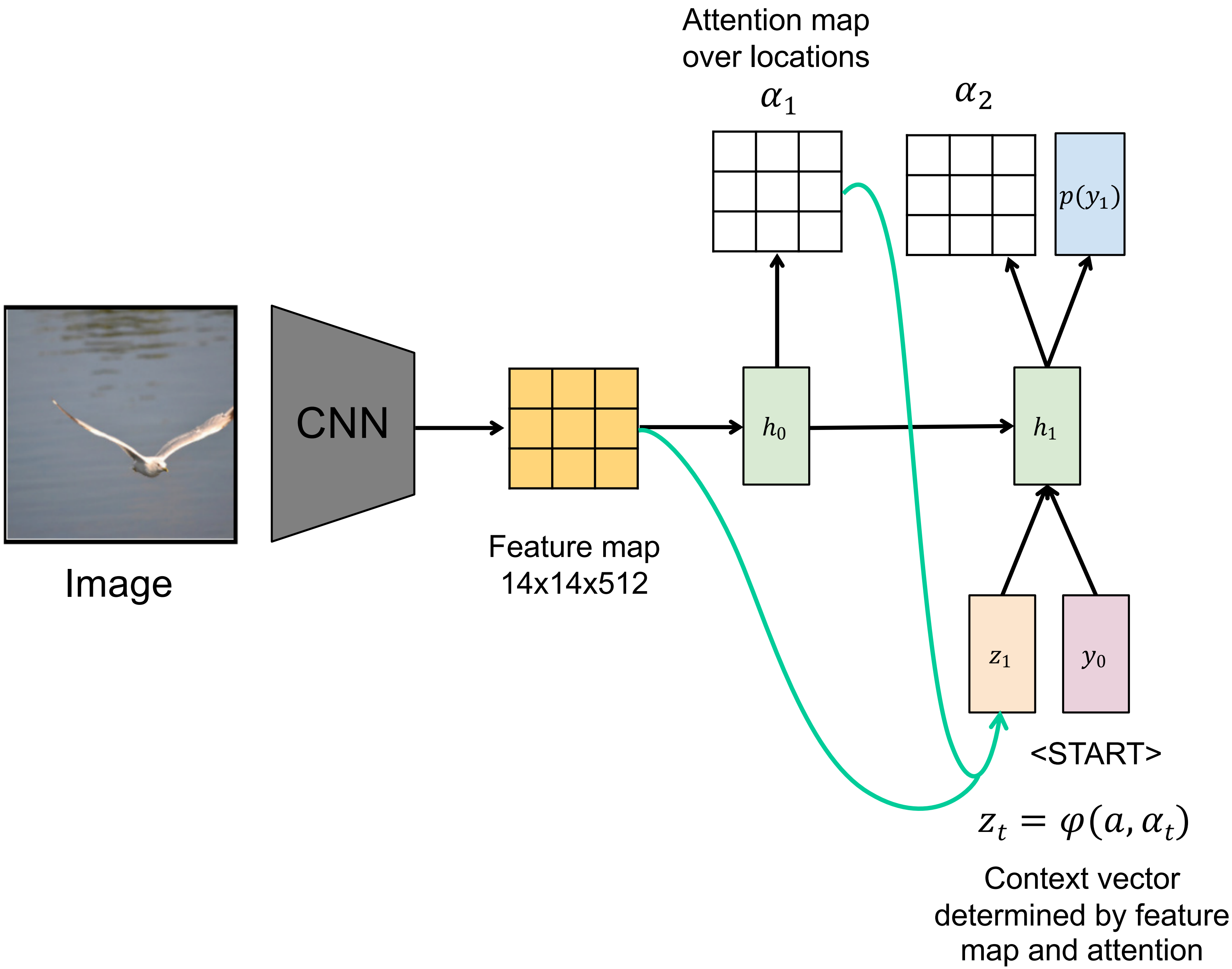
M1	Percentage of captions that are evaluated as better or equal to human caption.
M2	Percentage of captions that pass the Turing Test.
M3	Average correctness of the captions on a scale 1-5 (incorrect - correct).
M4	Average amount of detail of the captions on a scale 1-5 (lack of details - very detailed).
M5	Percentage of captions that are similar to human description.

Captioning with attention



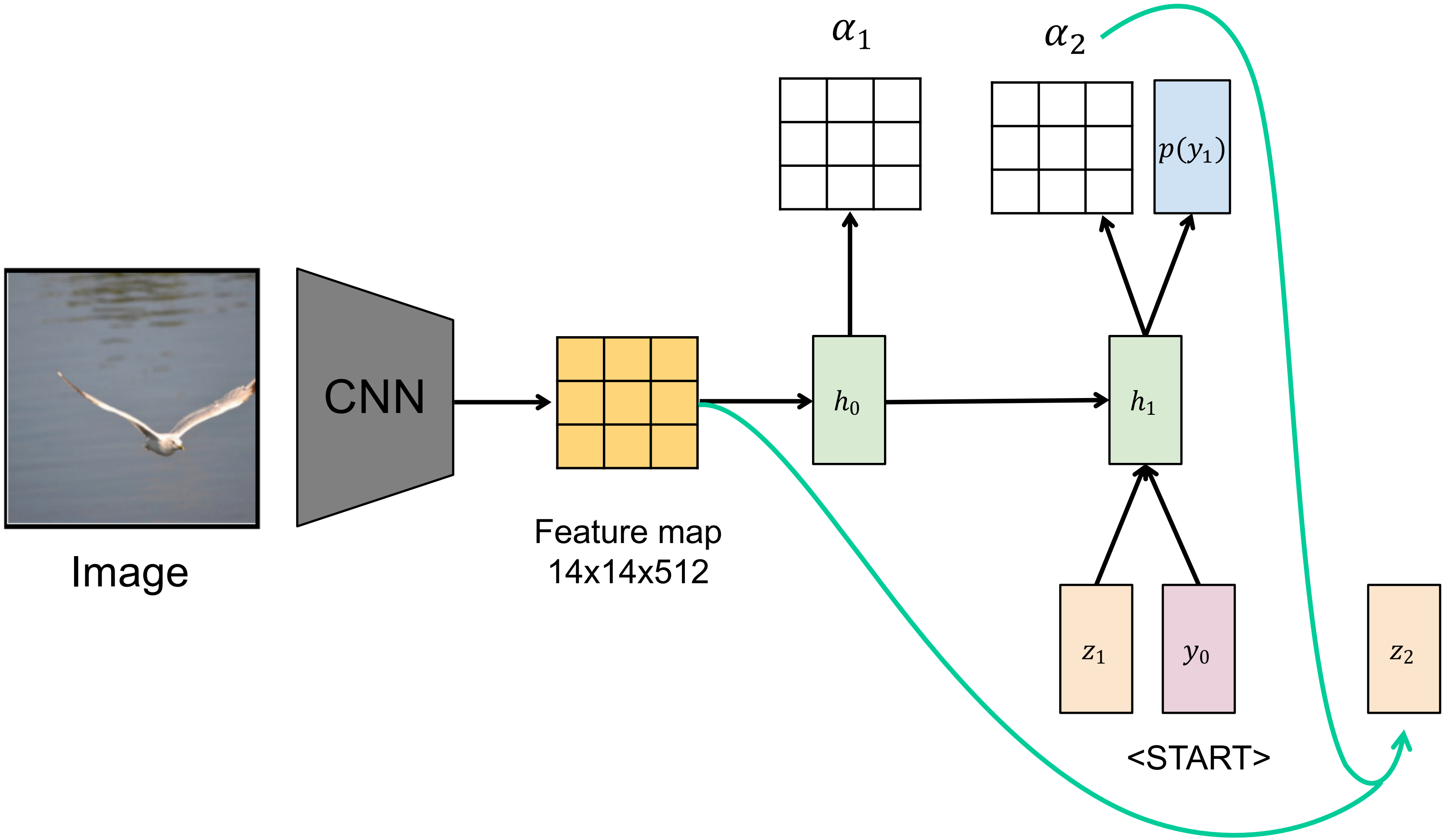
K. Xu et al., [Show, Attend and Tell: Neural Image Caption Generation with Visual Attention](), ICML 2015

Captioning with attention



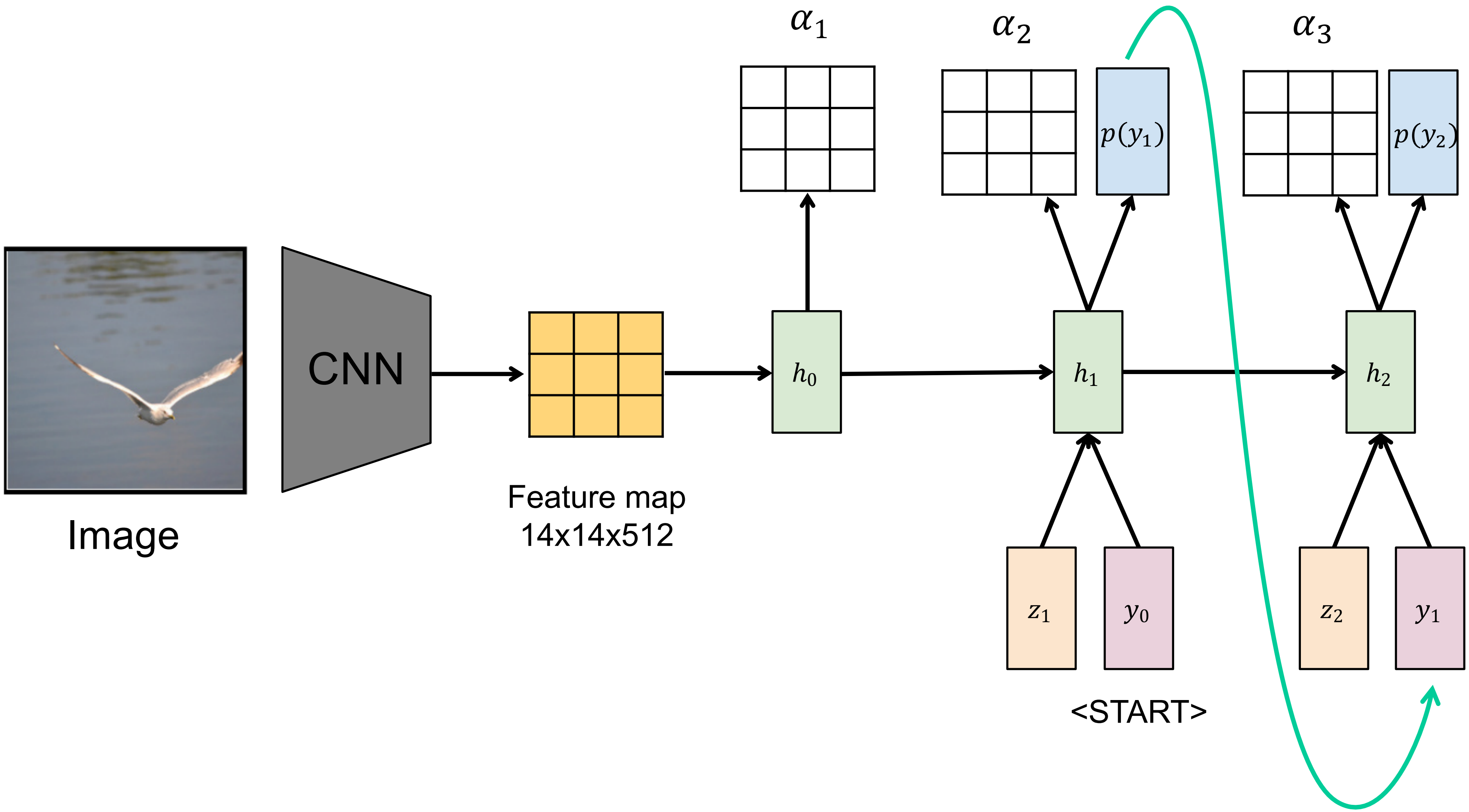
Adapted from [Stanford CS231n](), [Berkeley CS294]()

Captioning with attention



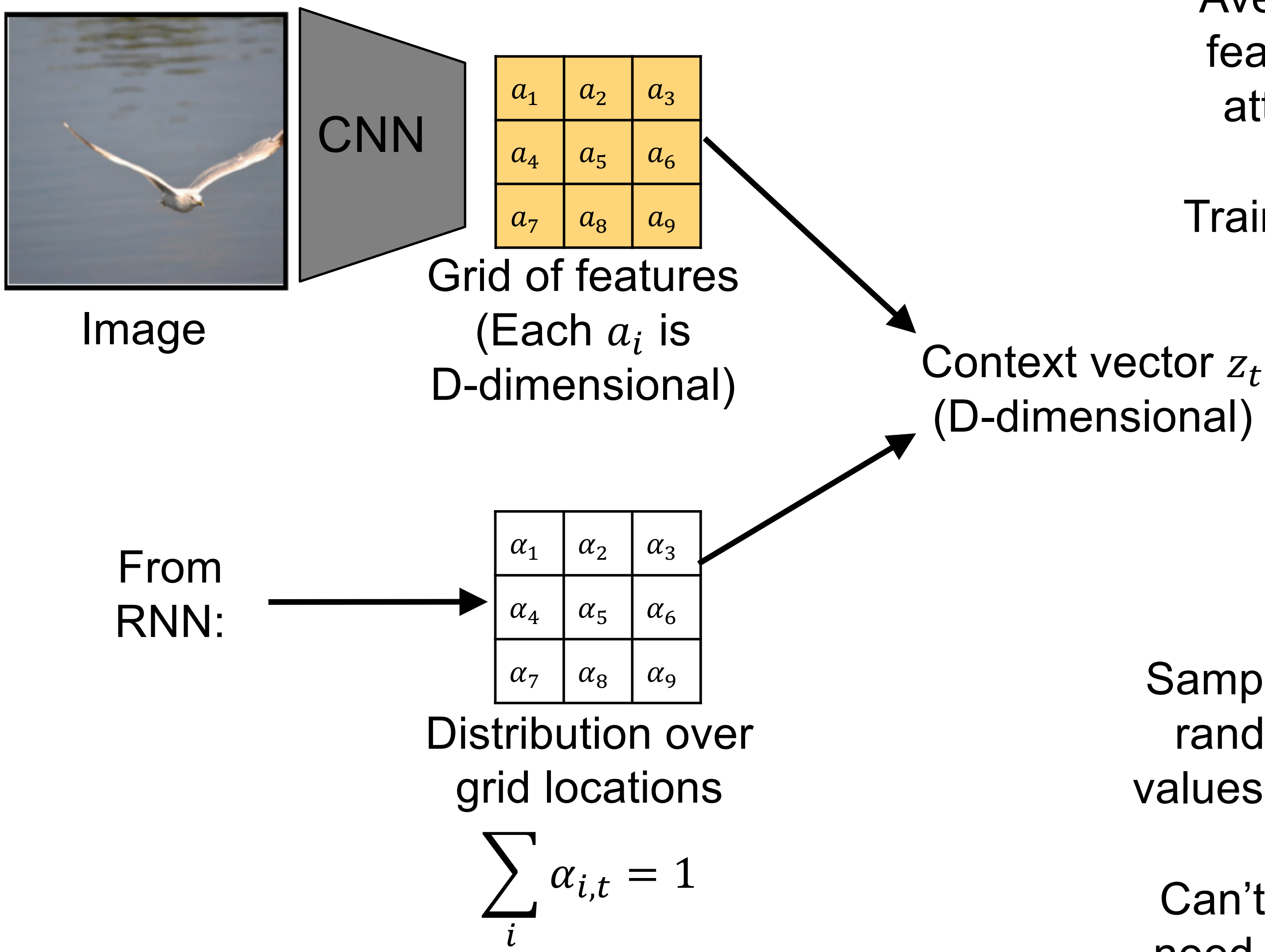
Adapted from Stanford CS231n, Berkeley CS294

Captioning with attention



Adapted from [Stanford CS231n](), [Berkeley CS294]()

“Soft” and “hard” attention



Soft attention:
Average over locations of feature map weighted by attention: $z_t = \sum_i \alpha_{i,t}\, a_i$

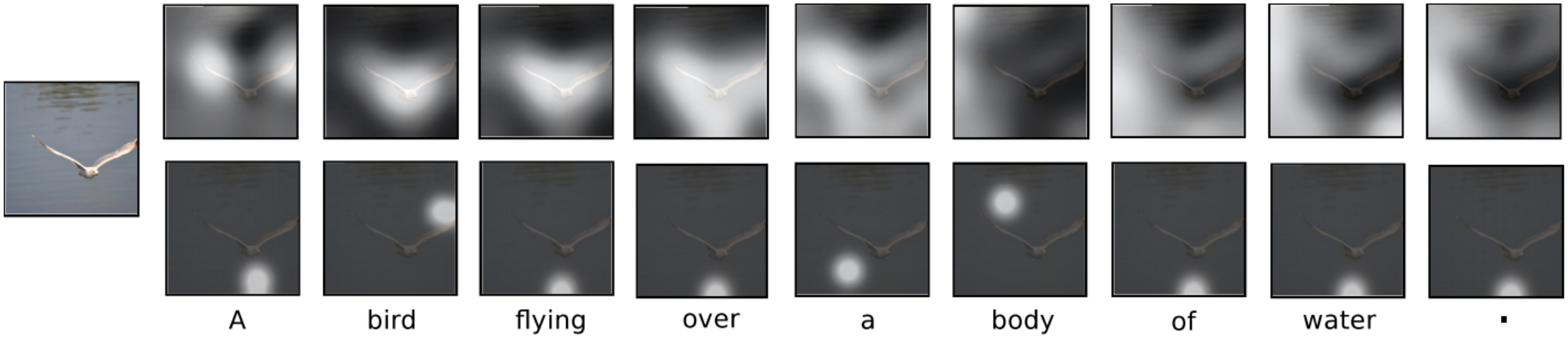
Train with gradient descent

Hard attention:
Sample ONE location: z_t is a random variable taking on values a_i with probabilities $\alpha_{i,t}$

Can’t use gradient descent; need reinforcement learning

“Soft” and “hard” attention

Soft attention:
Average over locations of feature map weighted by attention: $z_t = \sum_i \alpha_{i,t}\, a_i$



Hard attention:
Sample ONE location: z_t is a random variable taking on values a_i with probabilities $\alpha_{i,t}$

Results

Dataset	Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR
Flickr8k	Google NIC	63	41	27	-	-
	Soft-Attention	**67**	44.8	29.9	19.5	18.93
	Hard-Attention	**67**	**45.7**	**31.4**	**21.3**	**20.30**
Flickr30k	Google NIC	66.3	42.3	27.7	18.3	-
	Soft-Attention	66.7	43.4	28.8	19.1	**18.49**
	Hard-Attention	**66.9**	**43.9**	**29.6**	**19.9**	18.46
COCO	Google NIC	66.6	46.1	32.9	24.6	-
	Soft-Attention	70.7	49.2	34.4	24.3	**23.90**
	Hard-Attention	**71.8**	**50.4**	**35.7**	**25.0**	23.04

Source

Example Results

- Good captions

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.

Example Results

- Mistakes

A large white bird standing in a forest.

A woman holding a clock in her hand.

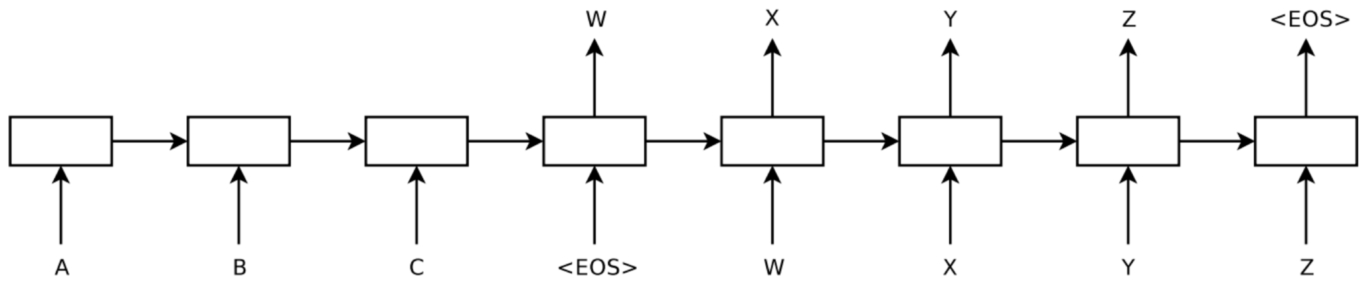
A man wearing a hat and
a hat on a skateboard.

A person is standing on a beach
with a surfboard.

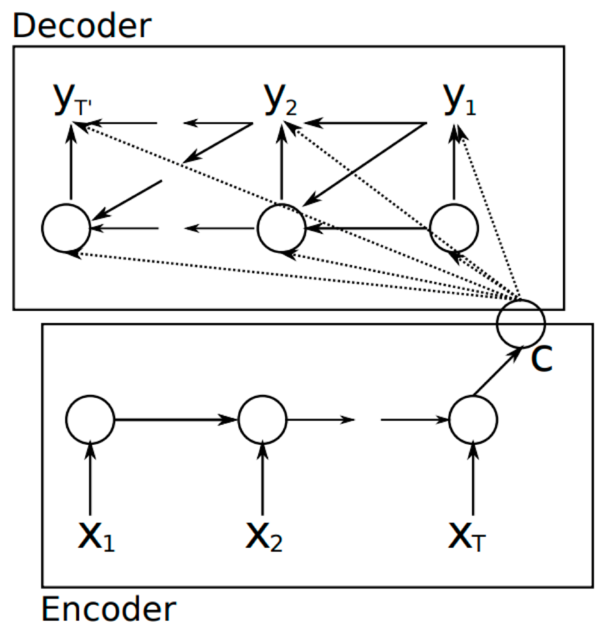
A woman is sitting at a table
with a large pizza.

A man is talking on his cell phone
while another man watches.

Machine translation: Vanilla Seq2Seq



I. Sutskever, O. Vinyals, Q. Le, [Sequence to Sequence Learning with Neural Networks](), NIPS 2014



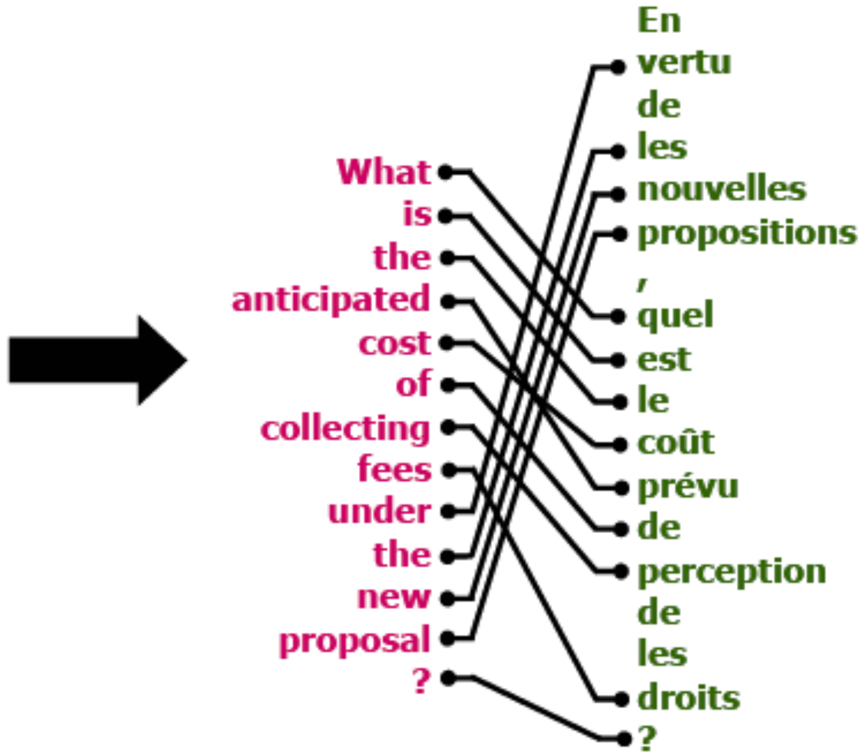
K. Cho, B. Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio, [Learning phrase representations using RNN encoder-decoder for statistical machine translation](), ACL 2014

Machine translation with attention

- Key idea: translation requires *alignment*

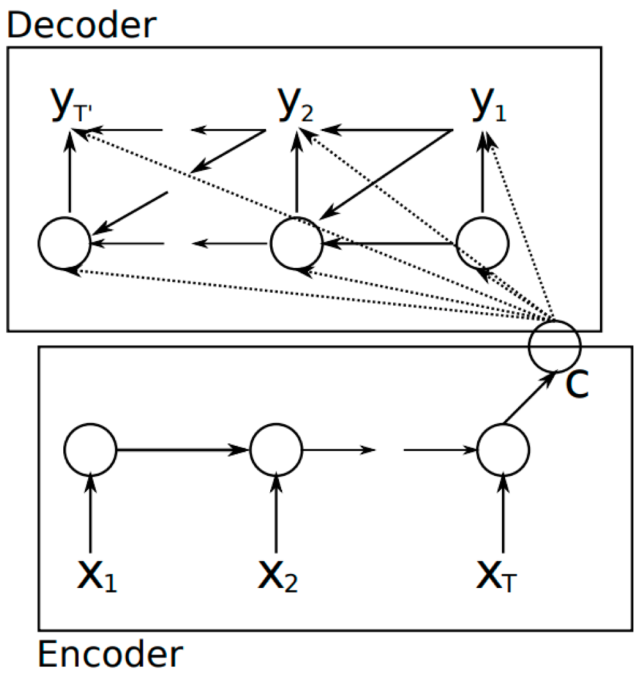
What is the anticipated cost of collecting fees under the new proposal?

En vertu des nouvelles propositions, quel est le coût prévu de perception des droits?



Machine translation with attention

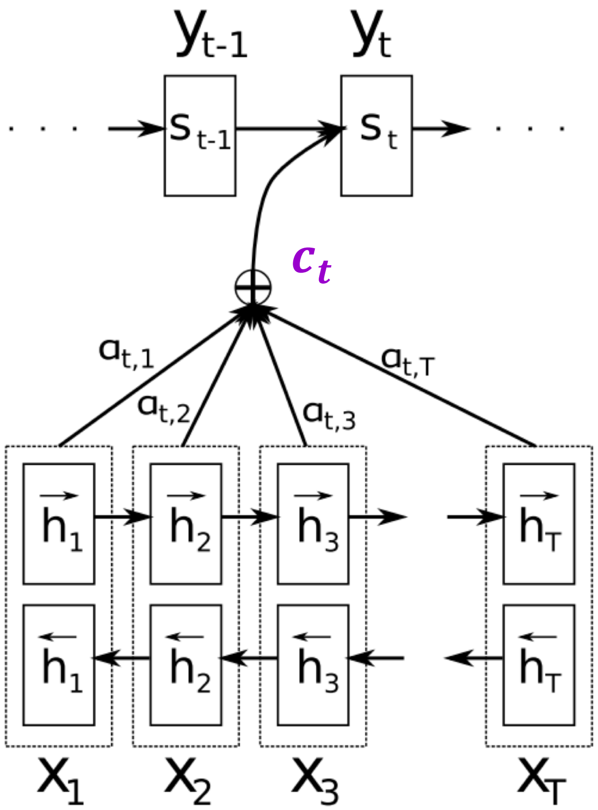
Standard encoder-decoder



A fixed context vector $c = h_T$ is used for decoding each word.

$$h_t = f(x_t, h_{t-1})$$

Attention-based model



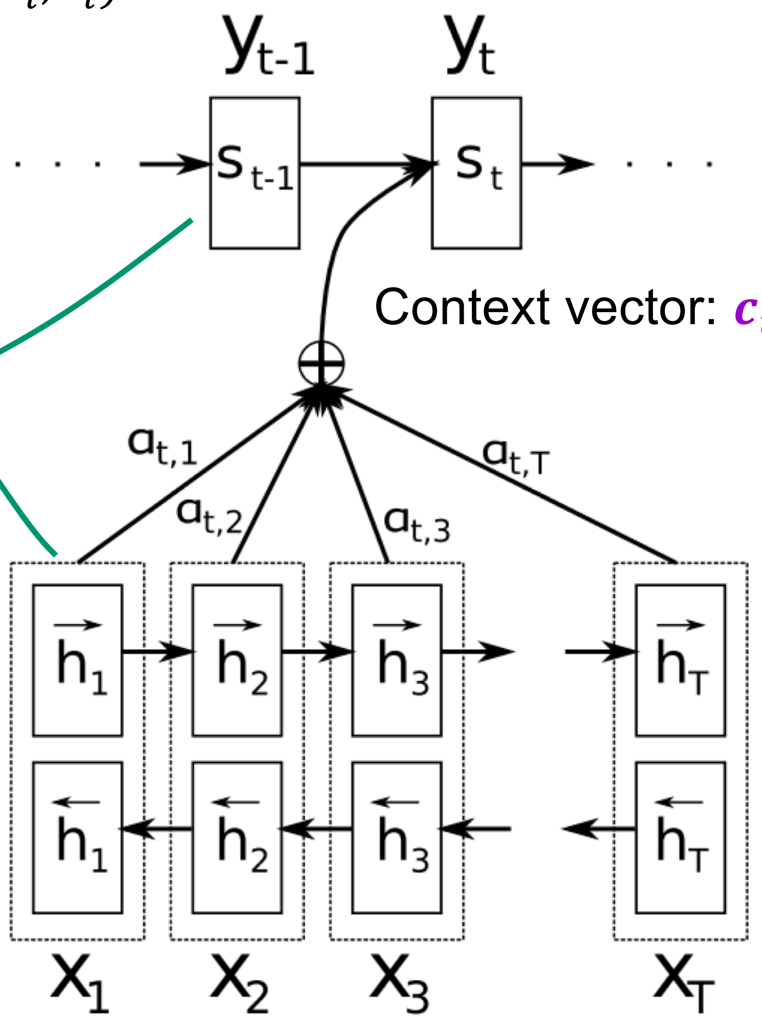
Context vector c_t pays attention to different phrases in the source when generating each word

D. Bahdanau, K. Cho, Y. Bengio, Neural Machine Translation by Jointly Learning to Align and Translate, ICLR 2015

Global attentional model

$$p(y_i|y_1, y_2, \dots, y_{i-1}, \boldsymbol{x}) = g(y_{i-1}, s_i, c_i)$$

$$s_i = f(s_{i-1}, y_i, c_i)$$



Context vector: $\boldsymbol{c_t} = \sum_{j=1}^{T} \alpha_{tj} h_j$

Alignment model:
$e_{tj} = a(s_{t-1}, h_j)$

$\alpha_{tj} = \text{softmax}(e_{tj})$

Bidirectional LSTM encoder: $h_j = [\overrightarrow{h}_j; \overleftarrow{h}_j]$

D. Bahdanau, K. Cho, Y. Bengio, [Neural Machine Translation by Jointly Learning to Align and Translate](), ICLR 2015

Attention model

α_{tj}

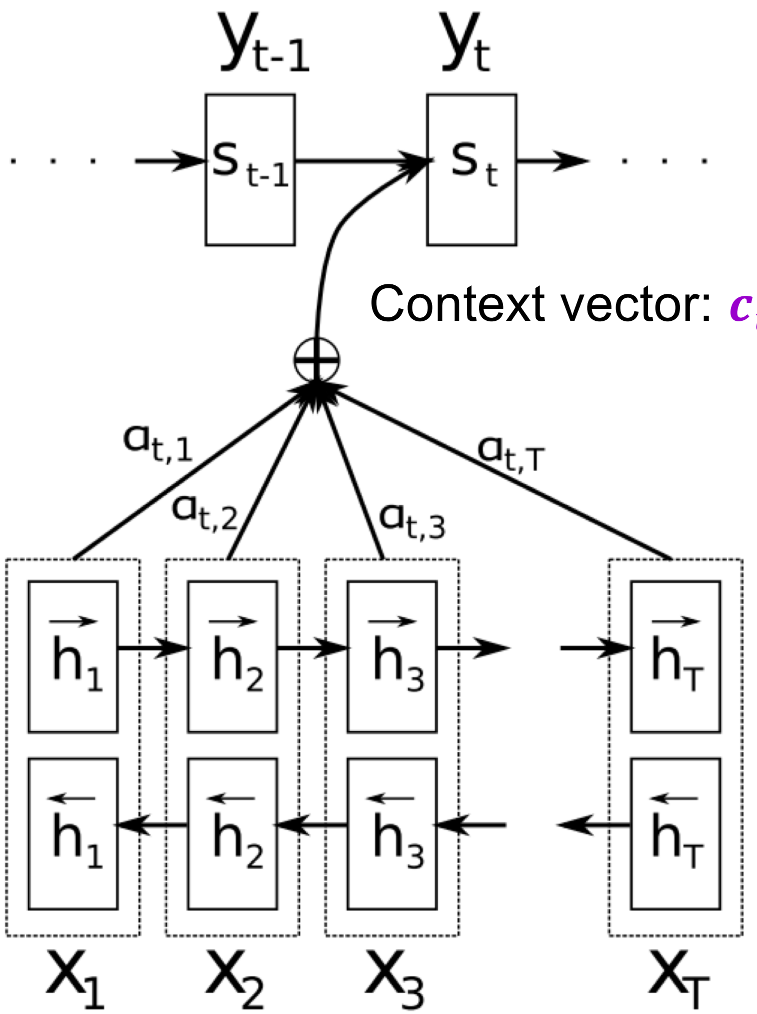
How much attention should output y_t pay to input x_j

$\alpha_{tj} = \text{softmax}(e_{tj})$

How to compute e_{tj}?

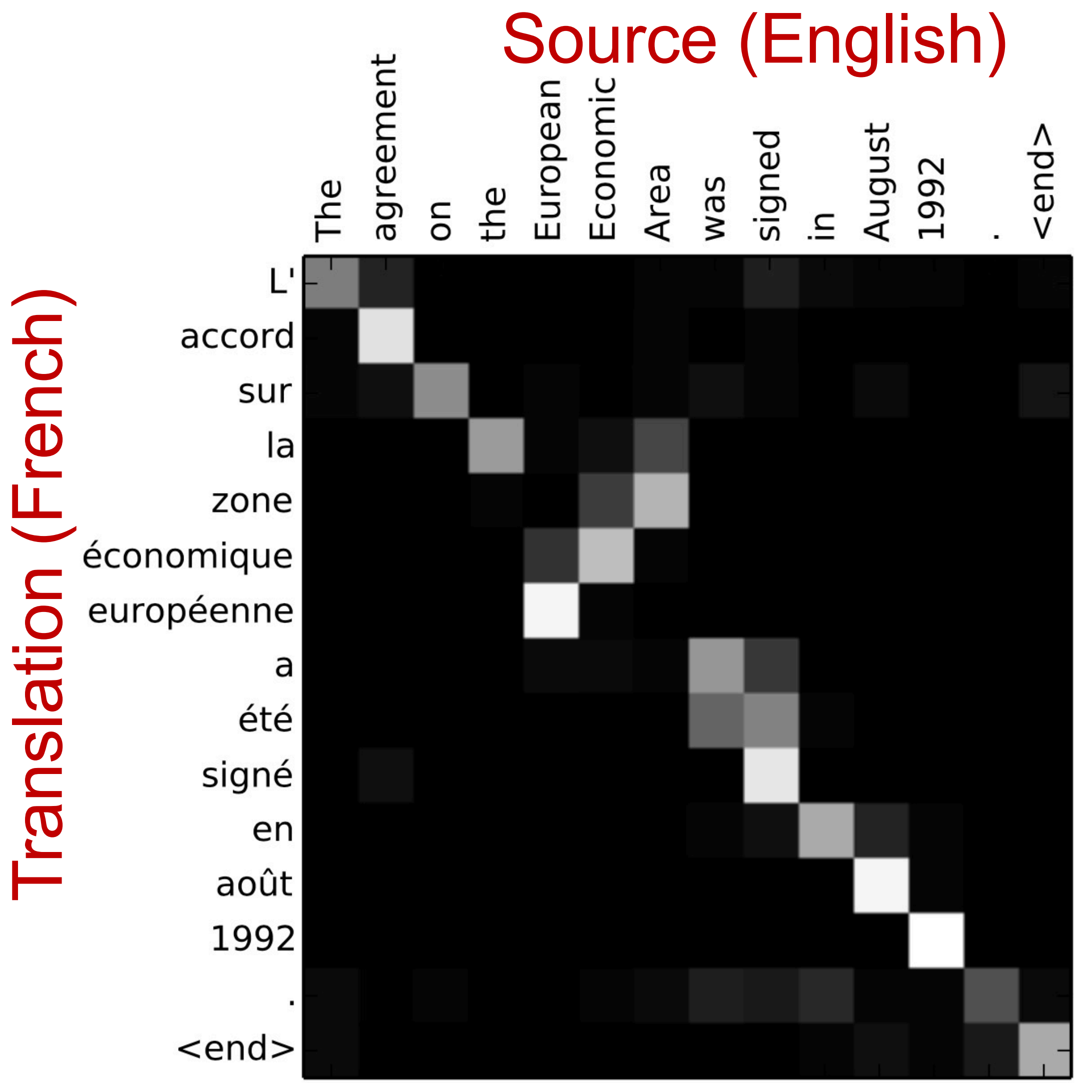
$e_{tj} = a(s_{t-1}, h_j)$

Train this using small NN
This model is effectively trying to align encoder hidden states with decoder hidden states

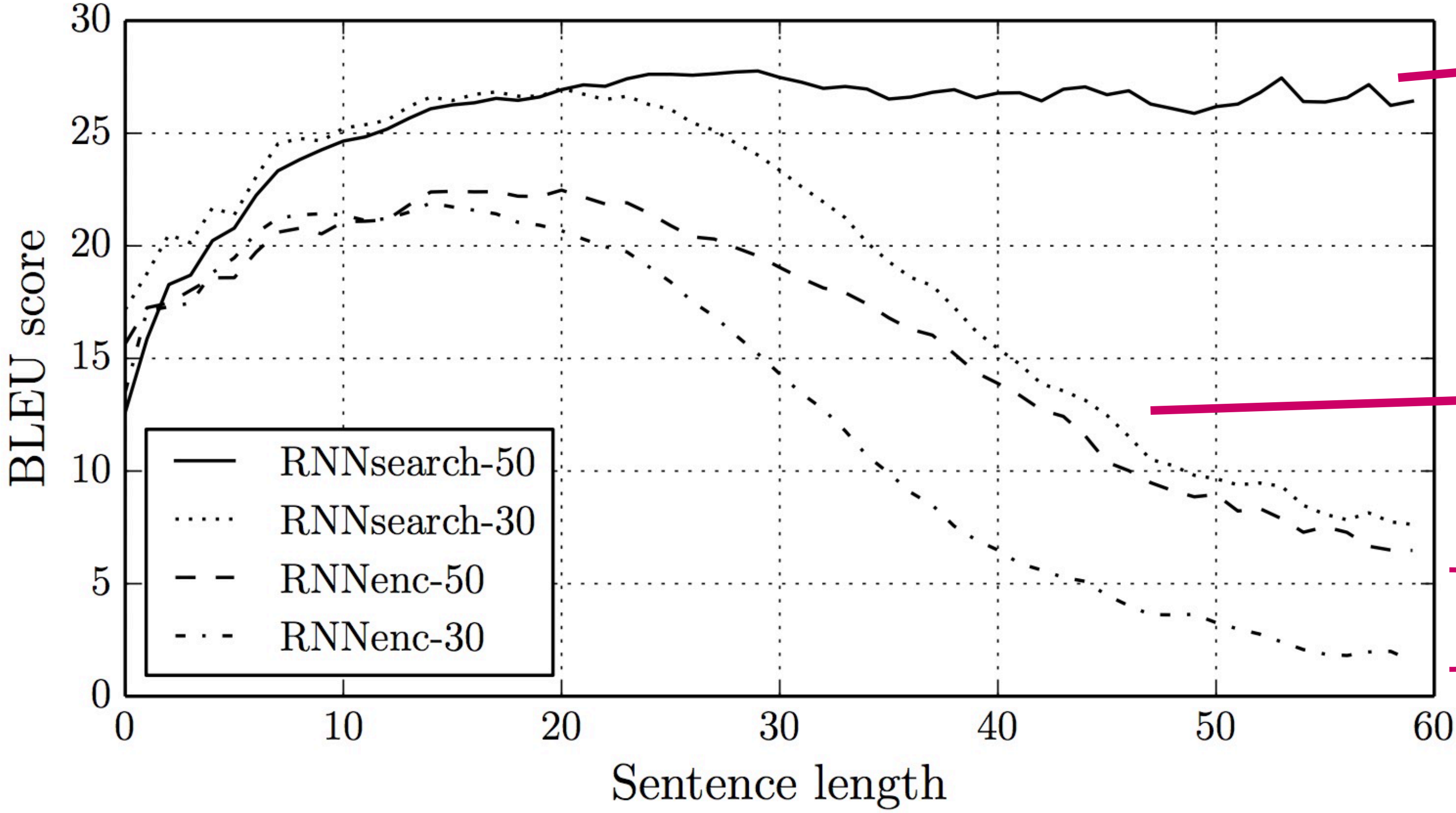


Context vector: $\boldsymbol{c_t} = \sum_{j=1}^{T} \alpha_{tj} h_j$

Example alignment



Quantitative evaluation



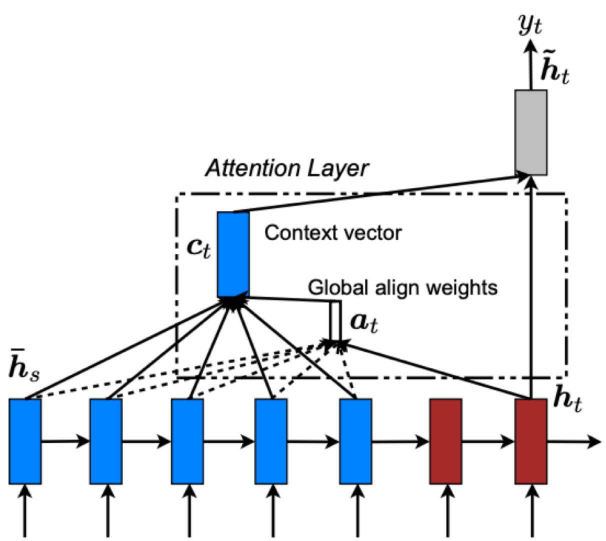
With attention (trained with sentence length <= 50)

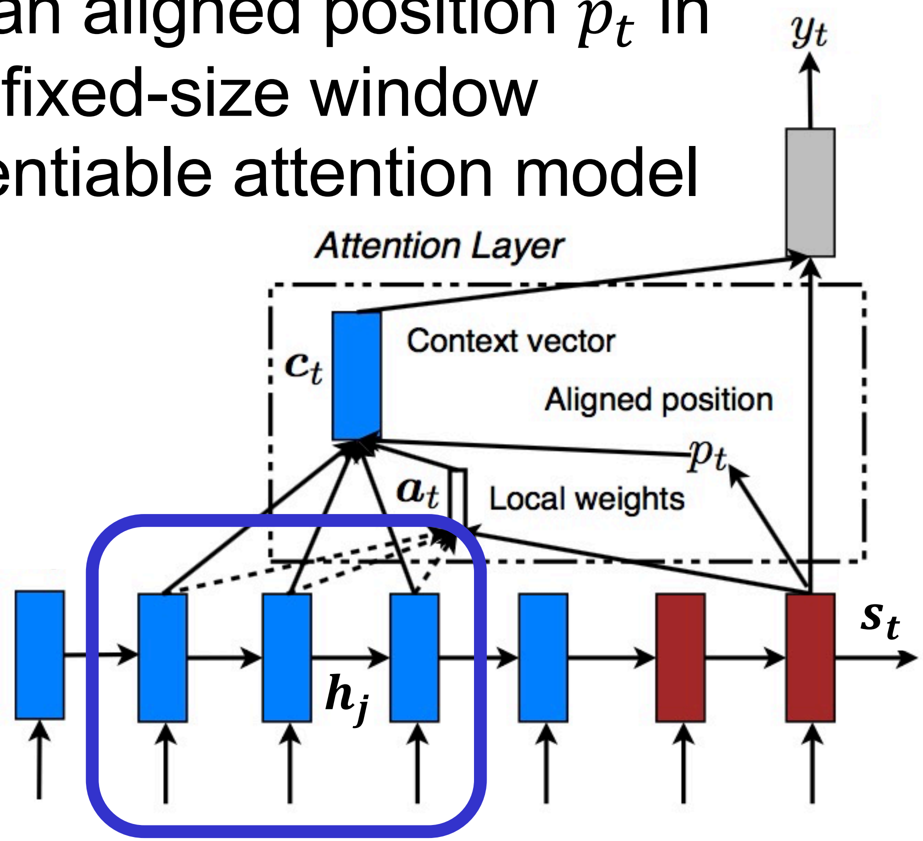
With attention (trained with sentence length <= 30)

No attention

Local attention

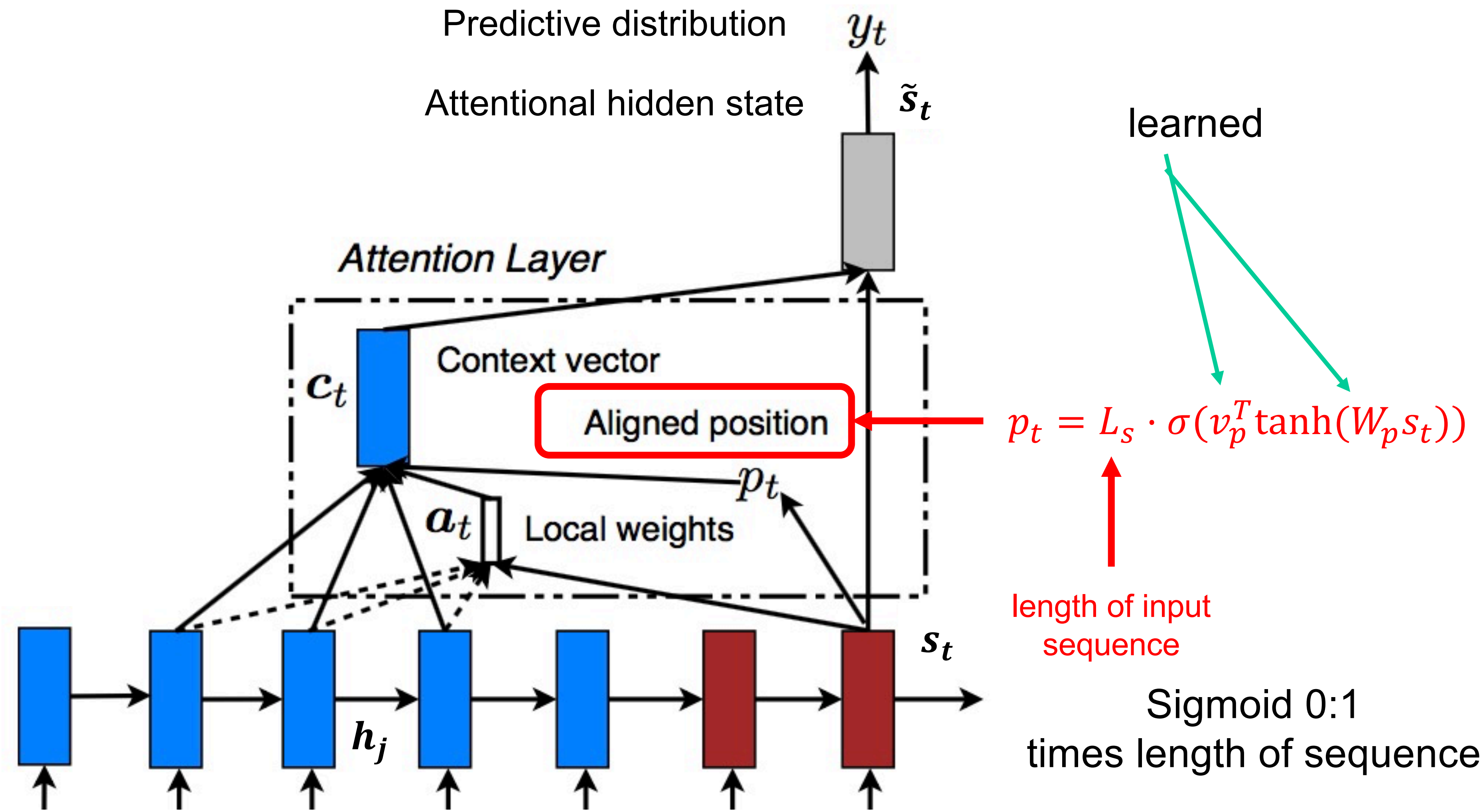
- Key idea: design mechanism similar to “hard attention” but differentiable
- Gloab attention - Attend over whole input
- For each target word, predict an aligned position p_t in the source; form context from fixed-size window around p_t - here simple differentiable attention model



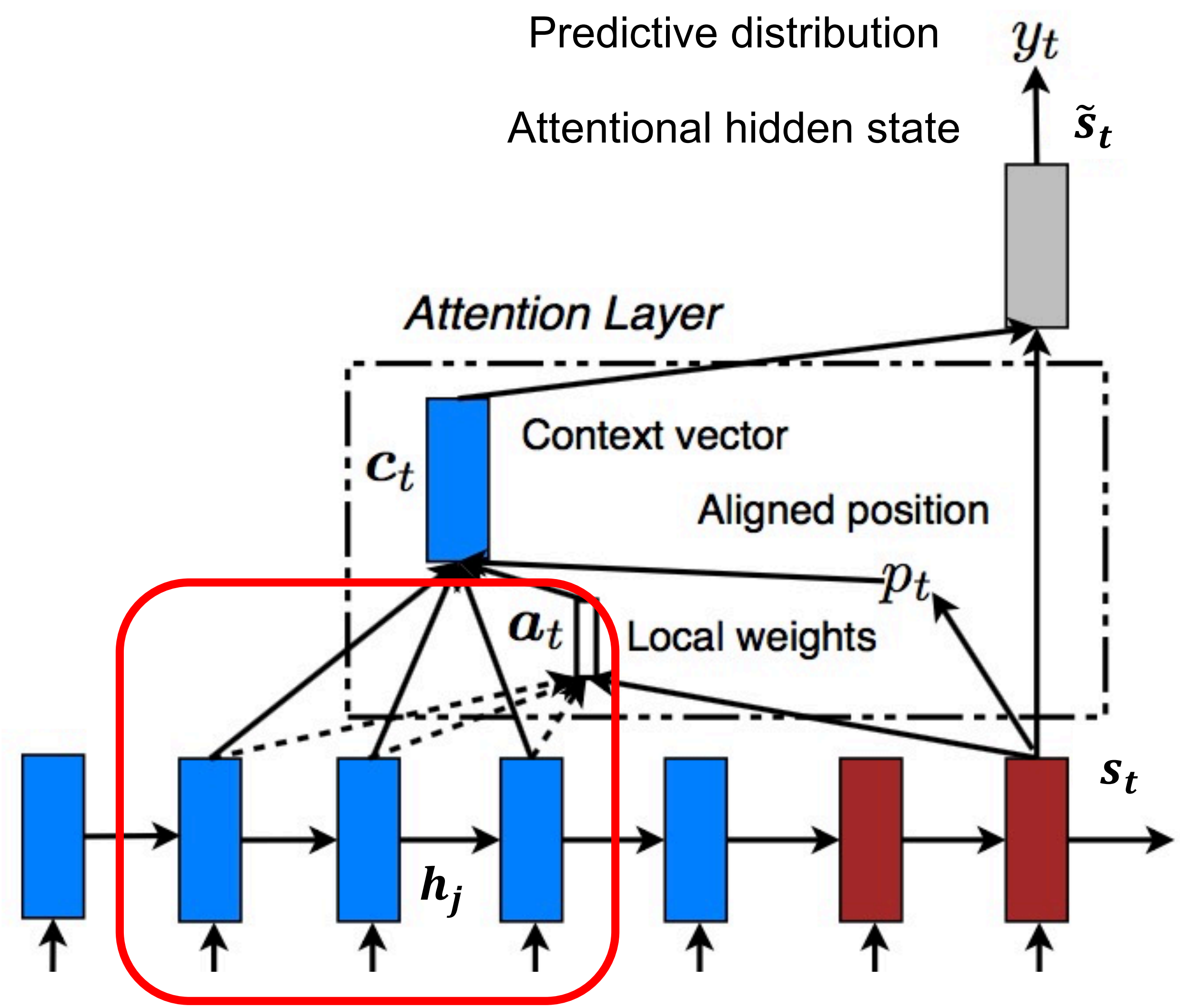


M.-T. Luong, H. Pham, and C. Manning, [Effective approaches to attention-based neural machine translation](), EMNLP 2015

Local attention



Local attention

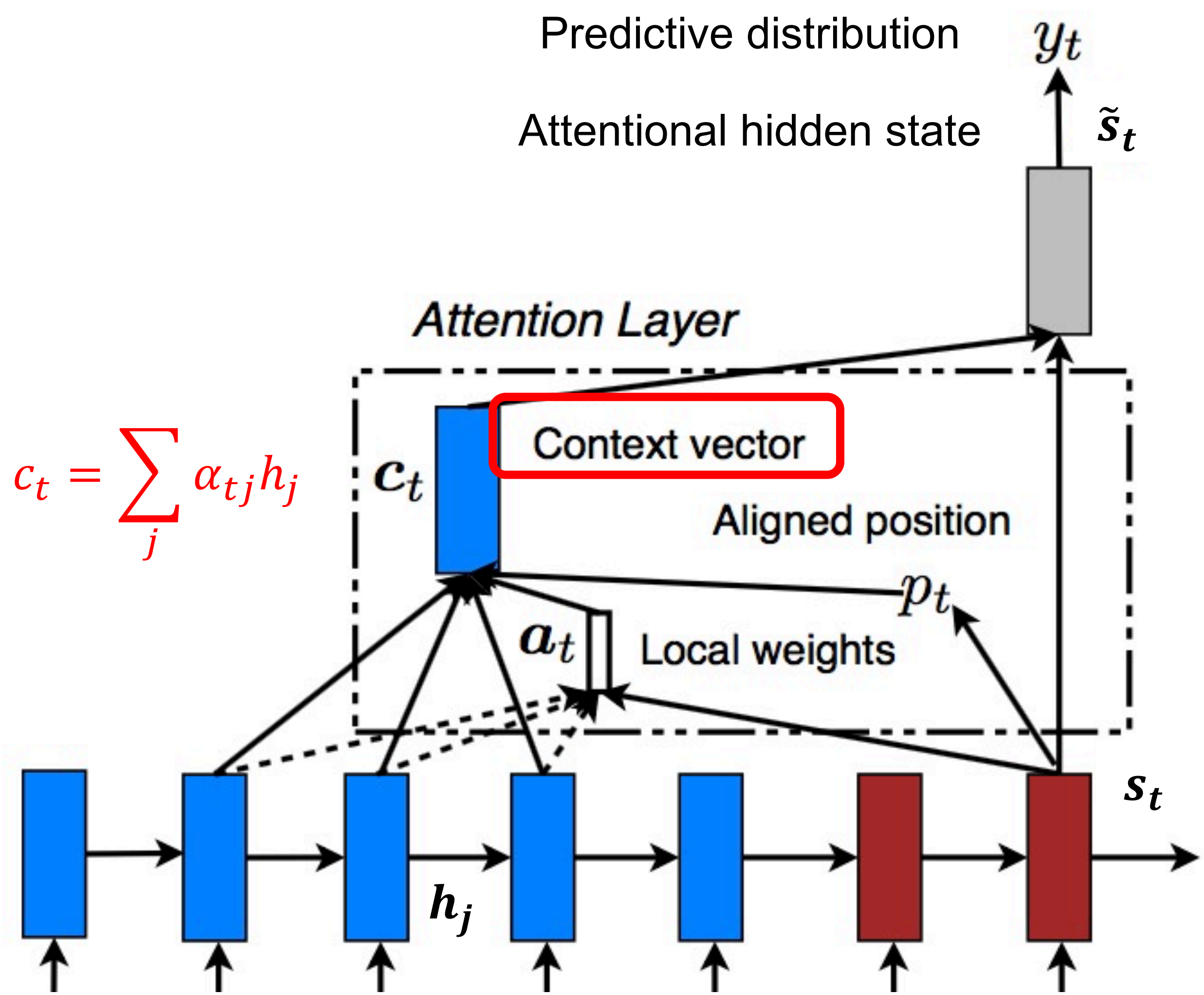


$p_t = L_s \cdot \sigma(v_p^T \tanh(W_p s_t))$

$\alpha_{tj} = \mathrm{softmax}(h_j^T W_a s_t) \exp(-\frac{(j - p_t)^2}{2\sigma^2})$

Local attention

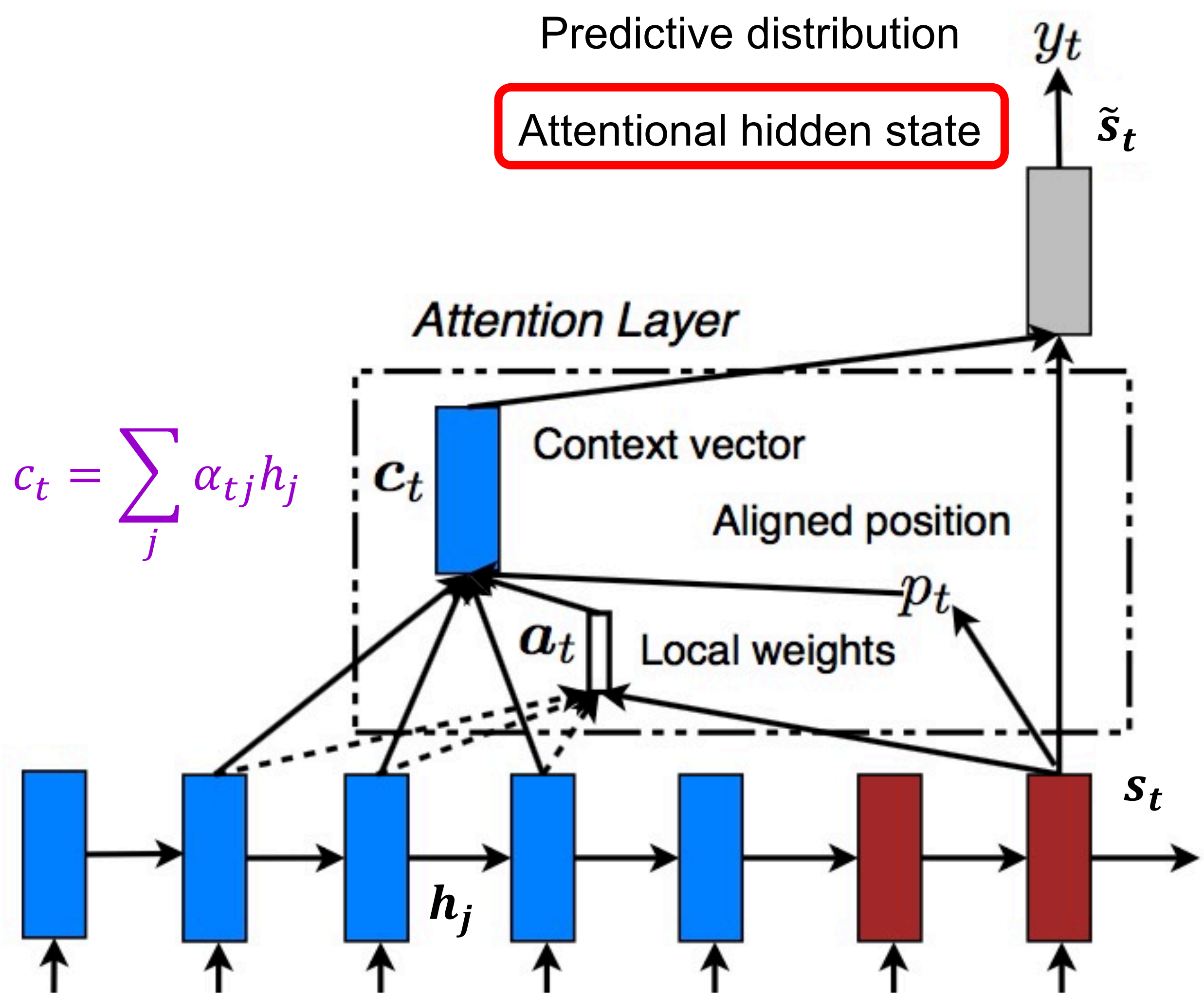
$$c_t = \sum_j \alpha_{tj} h_j$$



$$p_t = L_s \cdot \sigma(v_p^T \tanh(W_p s_t))$$

$$\alpha_{tj} = \text{softmax}(h_j^T W_a s_t) \exp\left(-\frac{(j - p_t)^2}{2\sigma^2}\right)$$

Local attention



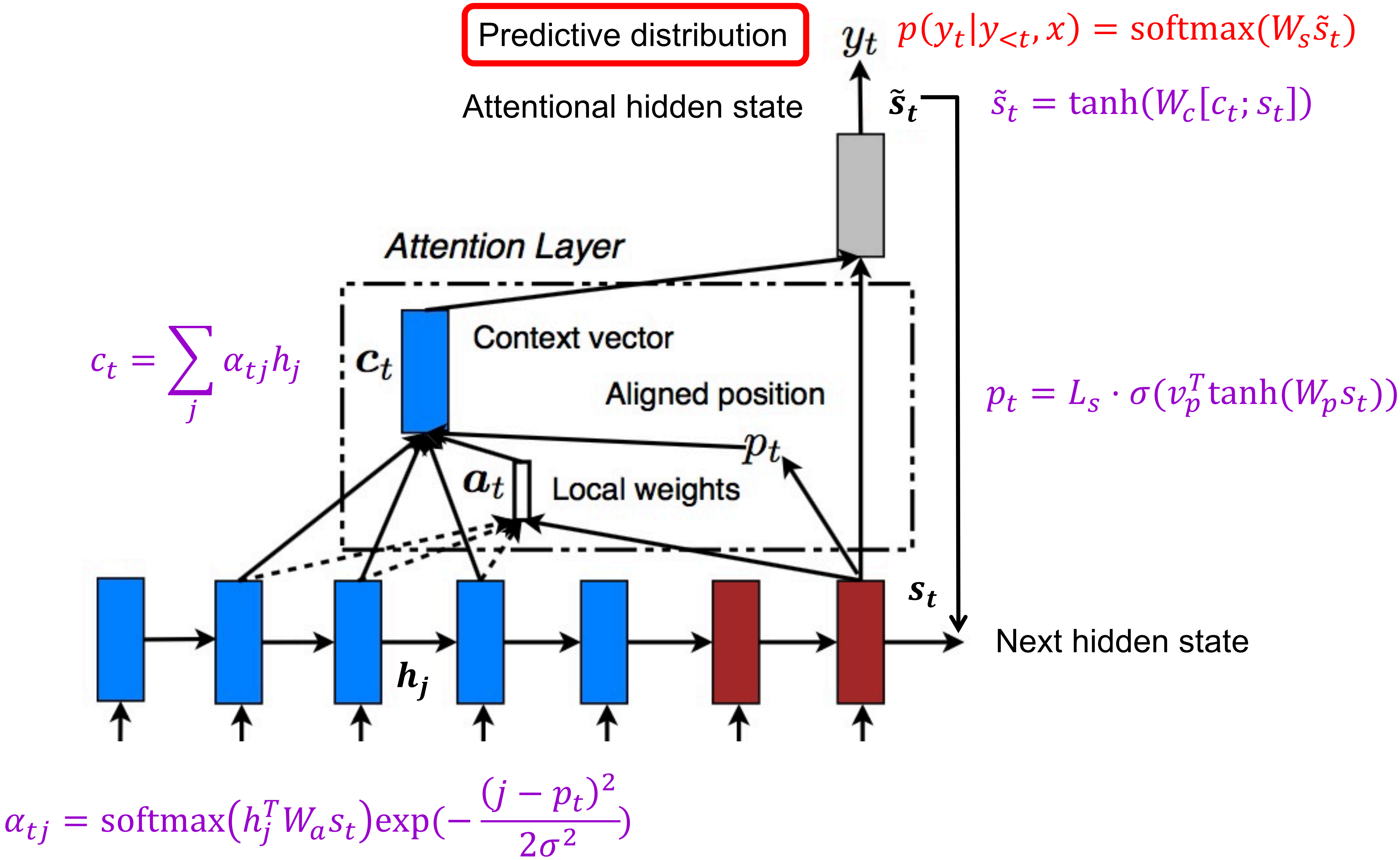
$$\tilde{s}_t = \tanh(W_c[c_t; s_t])$$

$$c_t = \sum_j \alpha_{tj} h_j$$

$$p_t = L_s \cdot \sigma(v_p^T \tanh(W_p s_t))$$

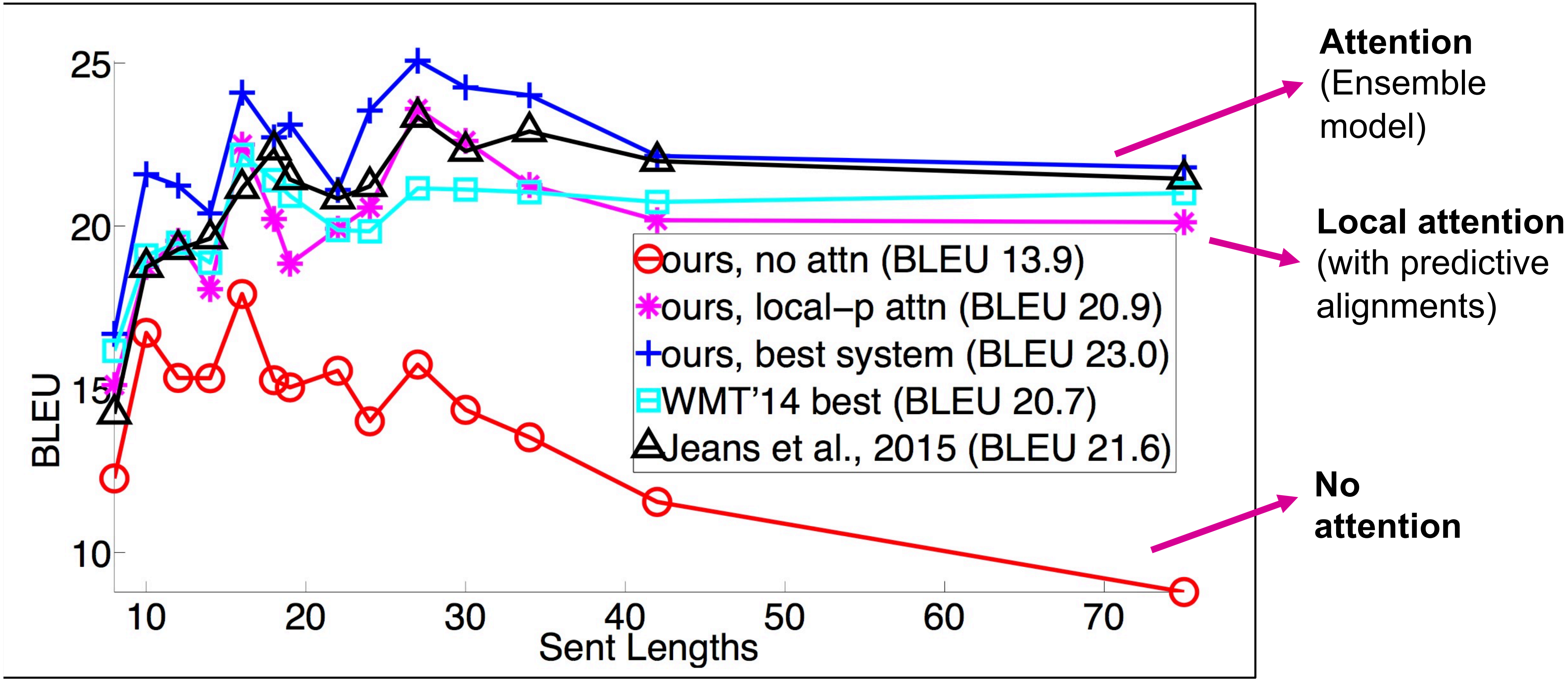
$$\alpha_{tj} = \mathrm{softmax}\big(h_j^T W_a s_t\big)\exp\Big(-\frac{(j - p_t)^2}{2\sigma^2}\Big)$$

Local attention



Results

- English-German translation



Google Neural Machine Translation (GNMT)

Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation

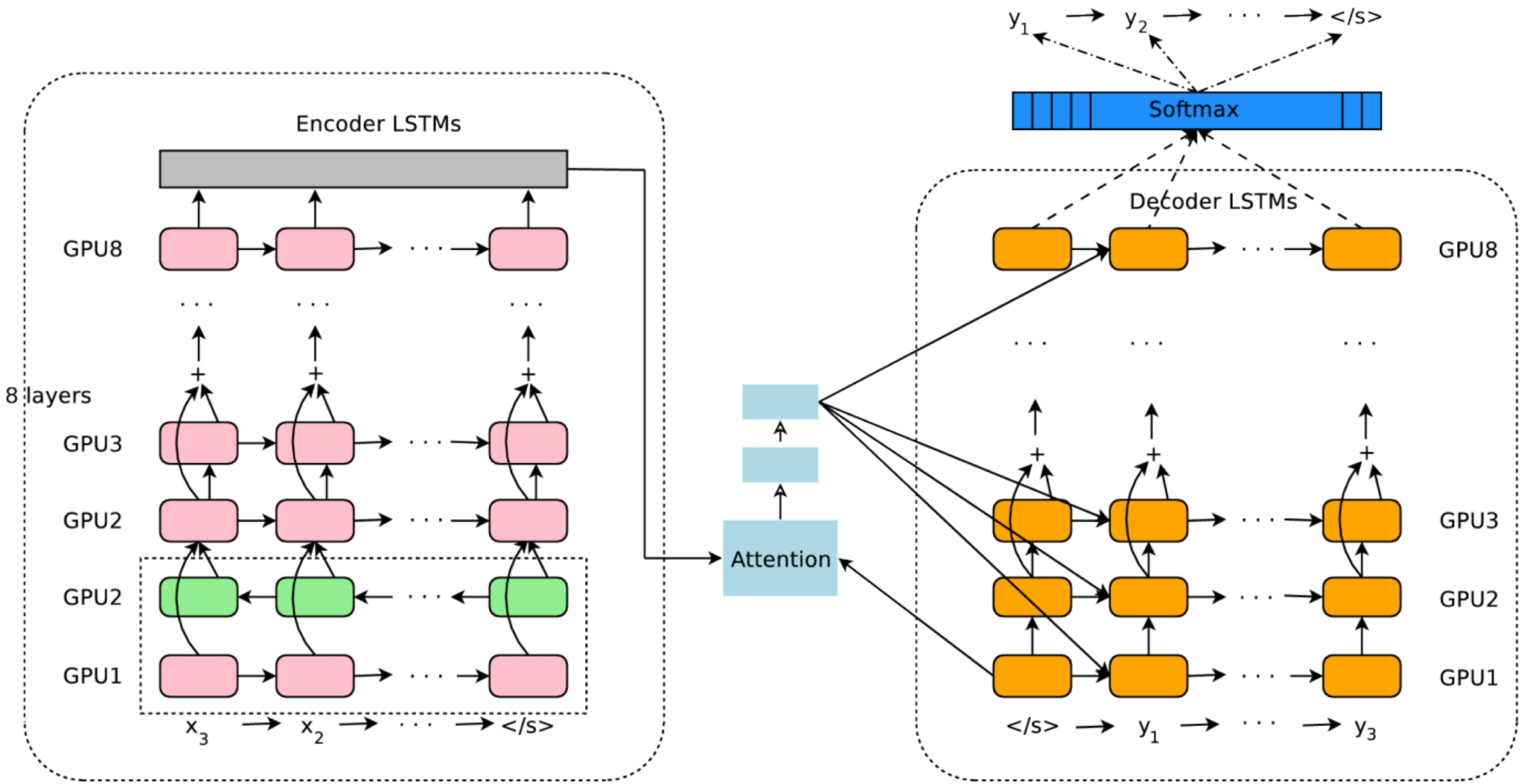
Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi
yonghui,schuster,zhifengc,qvl,mnorouzi@google.com

Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, Jeffrey Dean

Y. Wu et al., Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, arXiv 2016

https://www.nytimes.com/2016/12/14/magazine/the-great-ai-awakening.html

Google Neural Machine Translation (GNMT)



Y. Wu et al., [Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation](), arXiv 2016

Google Neural Machine Translation (GNMT)

- **Standard training objective:** maximize log-likelihood of ground truth output given input:

$$\sum_i \log P_W(Y_i^* | X_i)$$

  - Not related to task-specific reward function (e.g., BLEU score)
  - Does not encourage "better" predicted sentences to get better likelihood
- **GMNT objective:** expectation of rewards over possible predicted sentences Y:

$$\sum_i \sum_Y P_W(Y | X_i)\, r(Y, Y_i^*)$$

  - Use variant of BLEU score to compute reward
  - Reward is not differentiable -- need reinforcement learning to train (initialize with ML-trained model)

Google Neural Machine Translation (GNMT)

- Results on production data (500 randomly sampled sentences from Wikipedia and news websites)

Table 10: Mean of side-by-side scores on production data

	PBMT	GNMT	Human	Relative Improvement
English → Spanish	4.885	5.428	5.550	87%
English → French	4.932	5.295	5.496	64%
English → Chinese	4.035	4.594	4.987	58%
Spanish → English	4.872	5.187	5.372	63%
French → English	5.046	5.343	5.404	83%
Chinese → English	3.694	4.263	4.636	60%

Side-by-side scores: range from 0 (“completely nonsense translation”) to 6 (“perfect translation”), produced by human raters fluent in both languages

PBMT: Translation by phrase-based statistical translation system used by Google
GNMT: Translation by our GNMT system
Human: Translation by humans fluent in both languages

Convolutional sequence models

- Instead of recurrent networks, use 1D convolutional networks

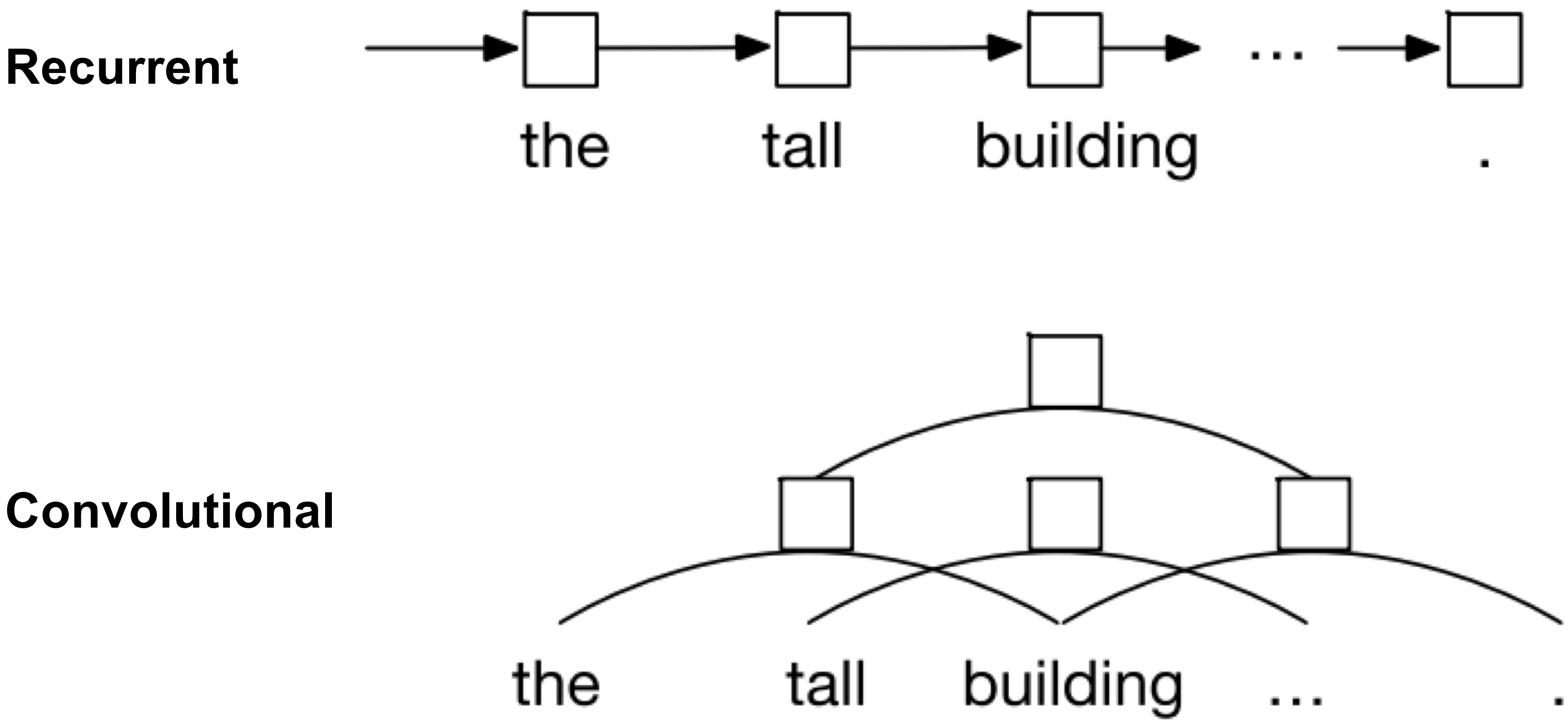


Image source

Convolutional sequence to sequence learning

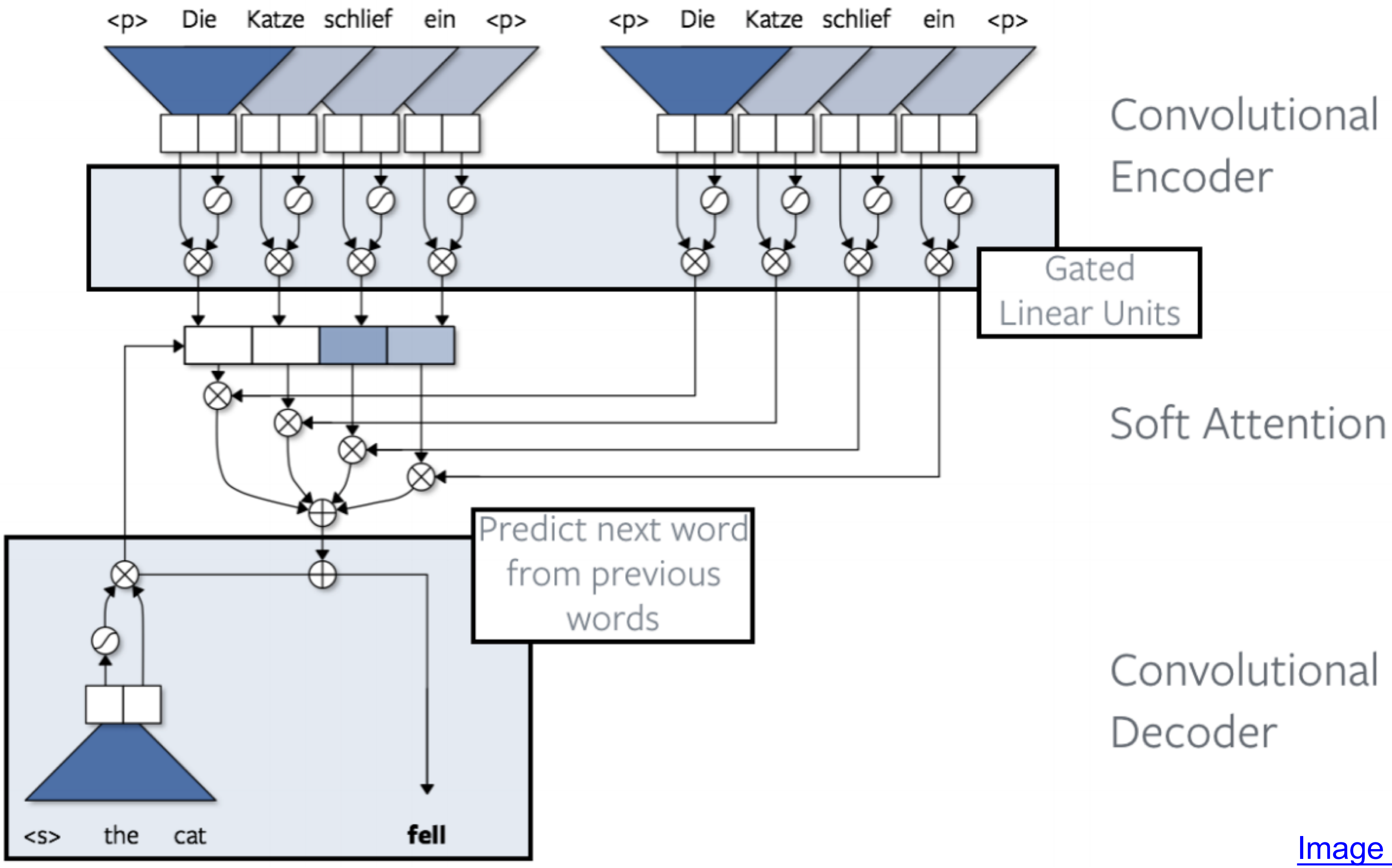


Image source

J. Gehring, M. Auli, D. Grangier, D. Yarats, Y. Dauphin, Convolutional sequence to sequence learning, ICML 2017

Convolutional sequence to sequence learning

- Results

WMT'14 English-German	BLEU
Luong et al. (2015) LSTM (Word 50K)	20.9
Kalchbrenner et al. (2016) ByteNet (Char)	23.75
Wu et al. (2016) GNMT (Word 80K)	23.12
Wu et al. (2016) GNMT (Word pieces)	24.61
ConvS2S (BPE 40K)	25.16

WMT'14 English-French	BLEU
Wu et al. (2016) GNMT (Word 80K)	37.90
Wu et al. (2016) GNMT (Word pieces)	38.95
Wu et al. (2016) GNMT (Word pieces) + RL	39.92
ConvS2S (BPE 40K)	40.51

Convolutional sequence models

- From the conclusion:

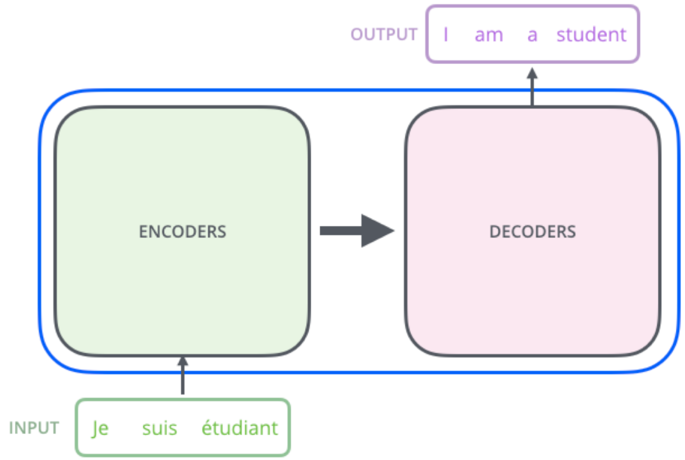
The preeminence enjoyed by recurrent networks in sequence modeling may be largely a vestige of history. Until recently, before the introduction of architectural elements such as dilated convolutions and residual connections, convolutional architectures were indeed weaker. Our results indicate that with these elements, a simple convolutional architecture is more effective across diverse sequence modeling tasks than recurrent architectures such as LSTMs. Due to the comparable clarity and simplicity of TCNs, we conclude that convolutional networks should be regarded as a natural starting point and a powerful toolkit for sequence modeling.

S. Bai, J. Kolter, and V. Koltun, An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling, arXiv 2018

Attention is all you need

- NMT architecture using only FC layers and attention
  - More efficient and parallelizable than recurrent or convolutional architectures, faster to train, better accuracy

Transformer



A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, I. Polosukhin, [Attention is all you need](), NIPS 2017

Attention is all you need

- NMT architecture using only FC layers and attention

Encoder: receives entire input sequence and outputs encoded sequence of the same length

Decoder: predicts one word at at a time, conditioned on encoder output and previously predicted words

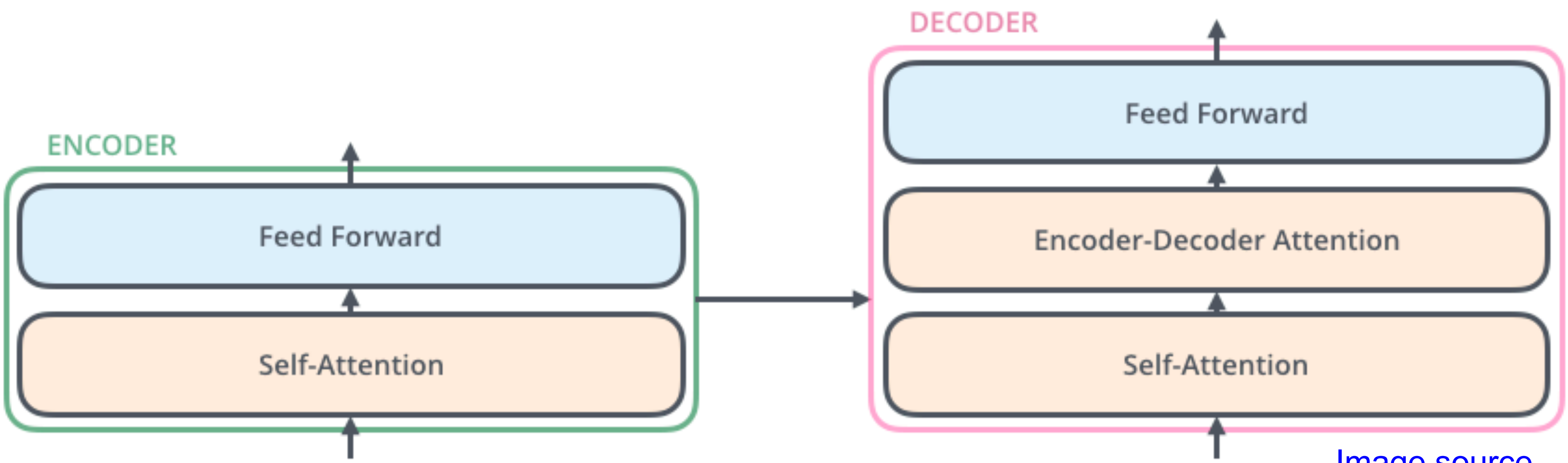


Image source

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, NIPS 2017

Self-attention

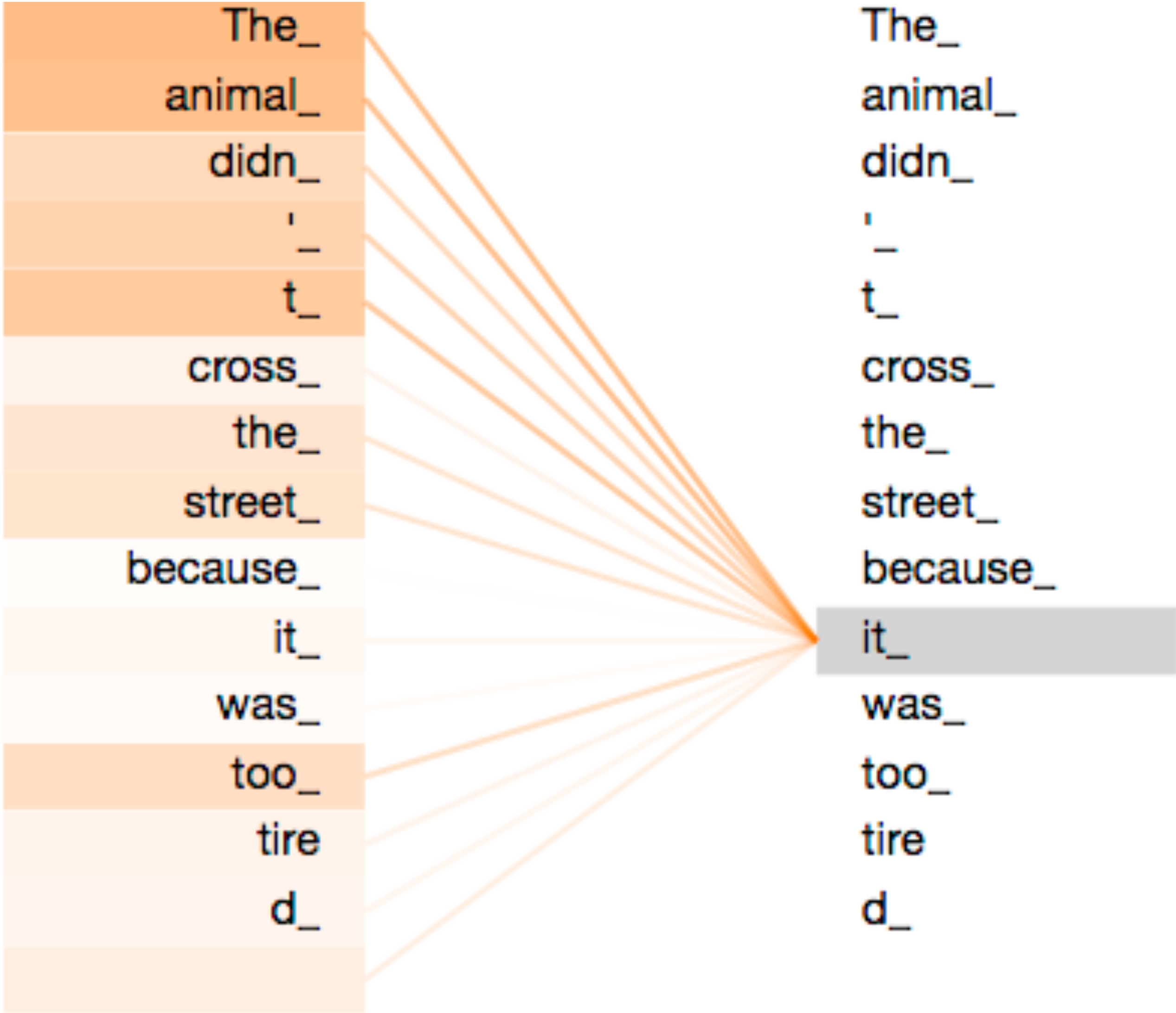
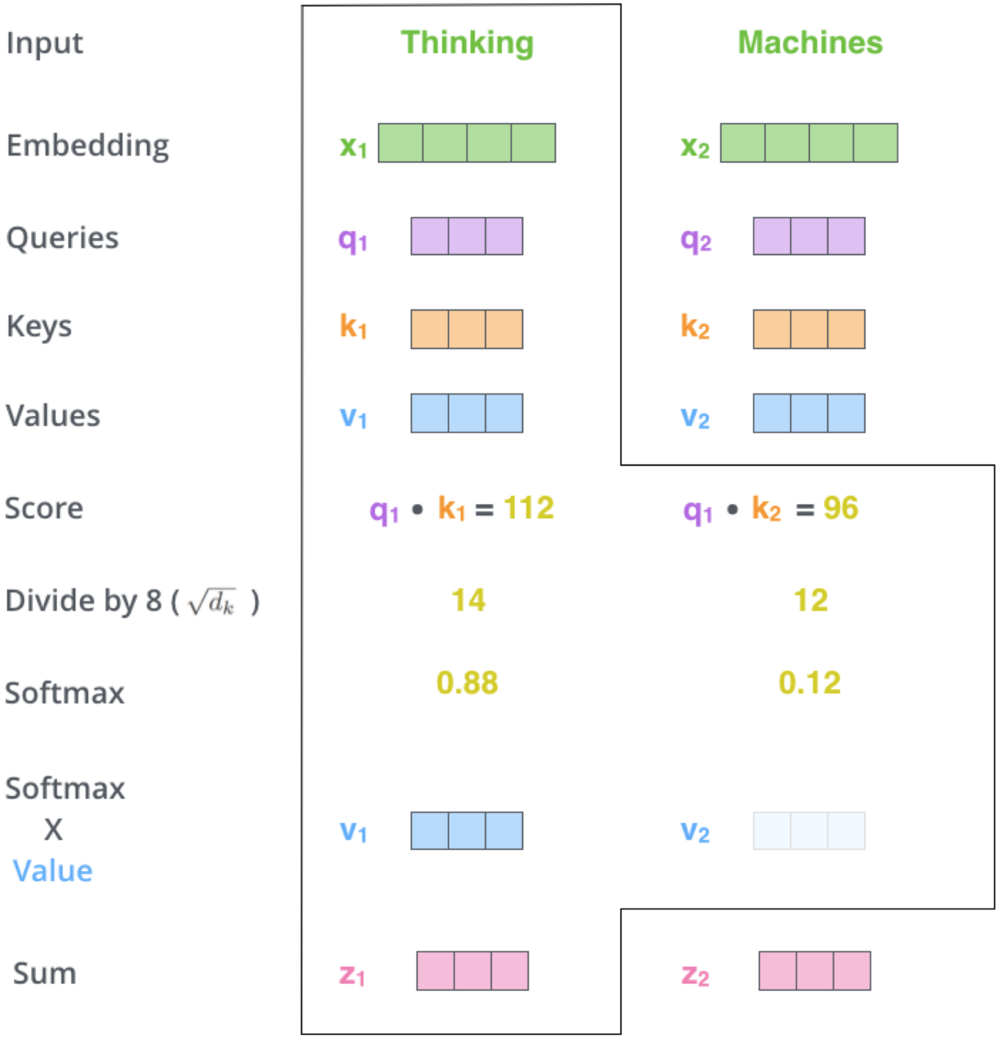
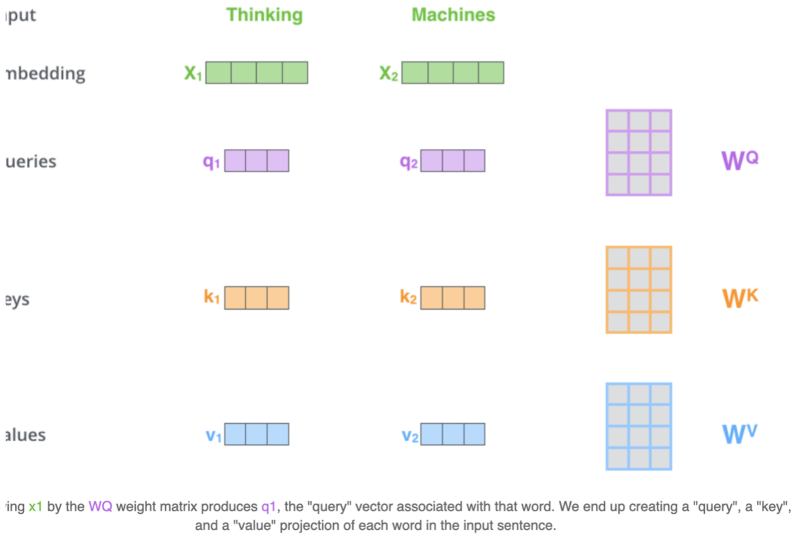


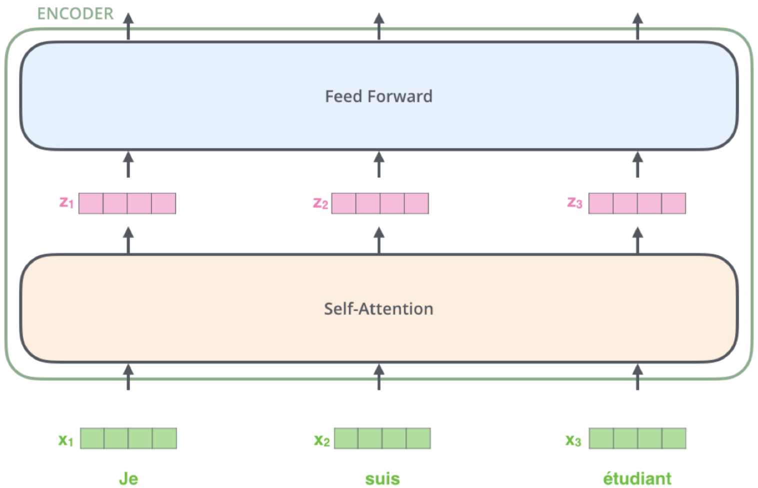
Image source

Self-attention details





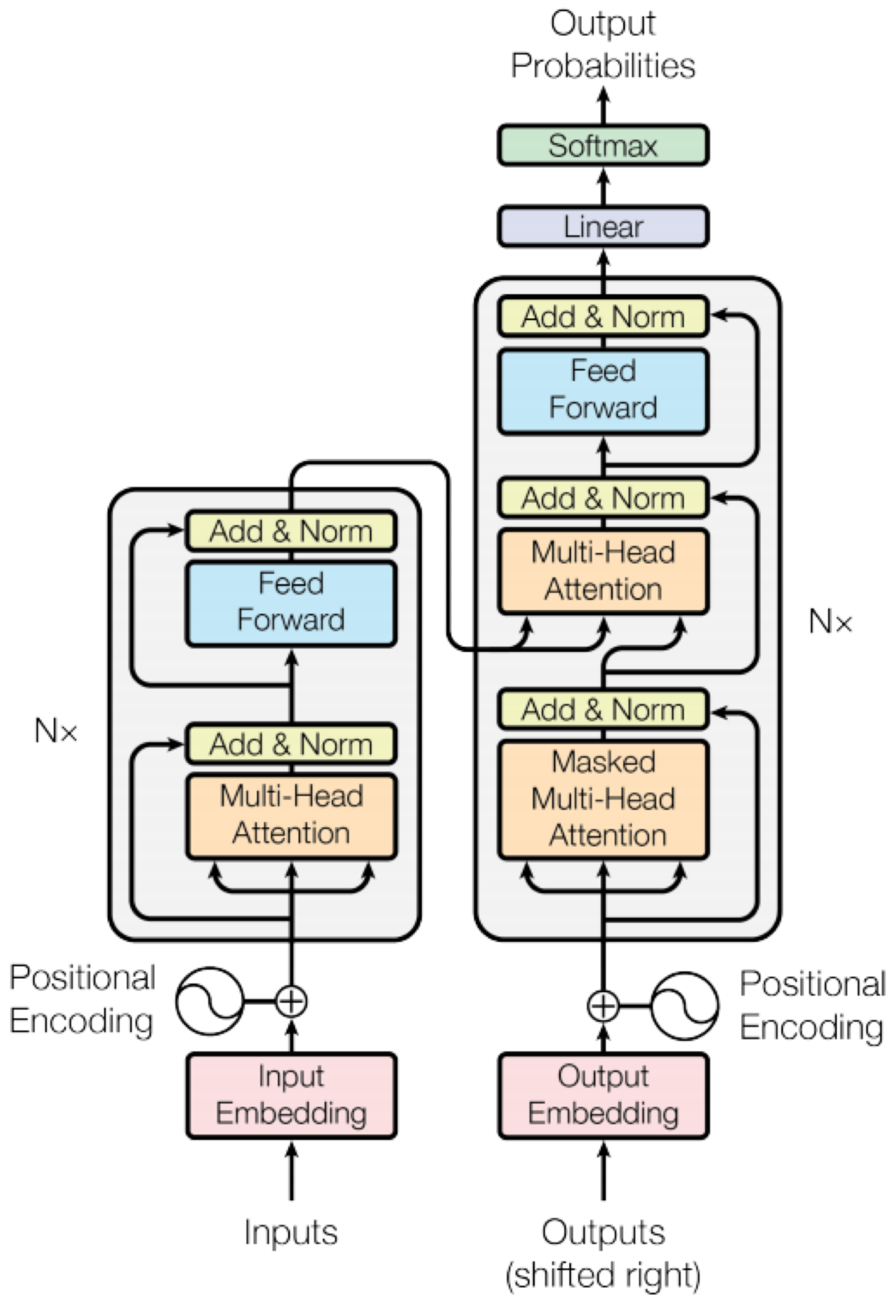
Generate better word embeddings depending on context



Transformer architecture in detail

Additional bells and whistles

Multiple attention heads



Positional encoding

- Hand-crafted encoding (using sines and cosines) is added to every input vector

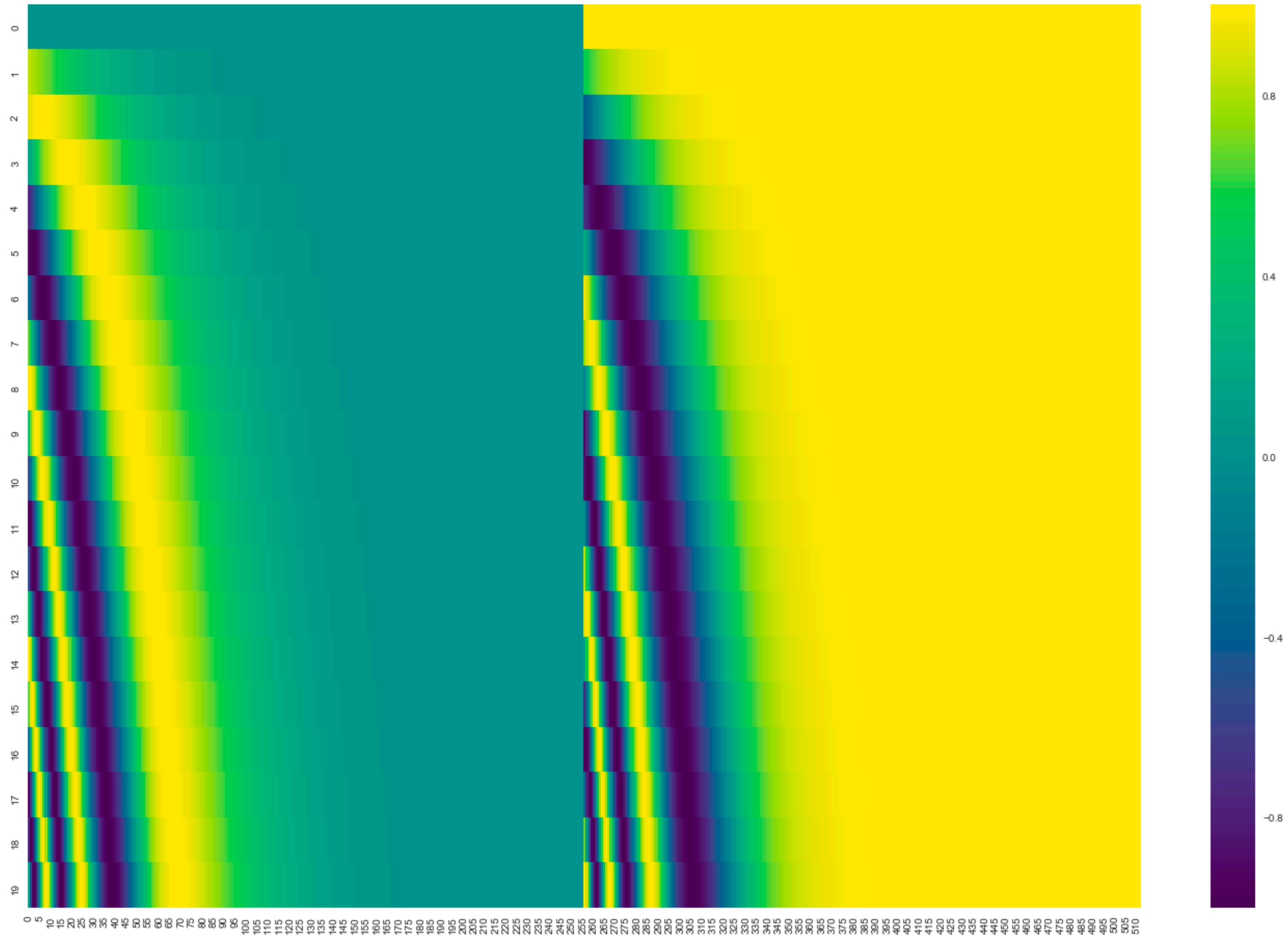


Image source

Attention mechanism

- *Scaled dot product attention*:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

  - Q, K, V are matrices with rows corresponding to queries, keys, and values, d_k is the dim. of the keys

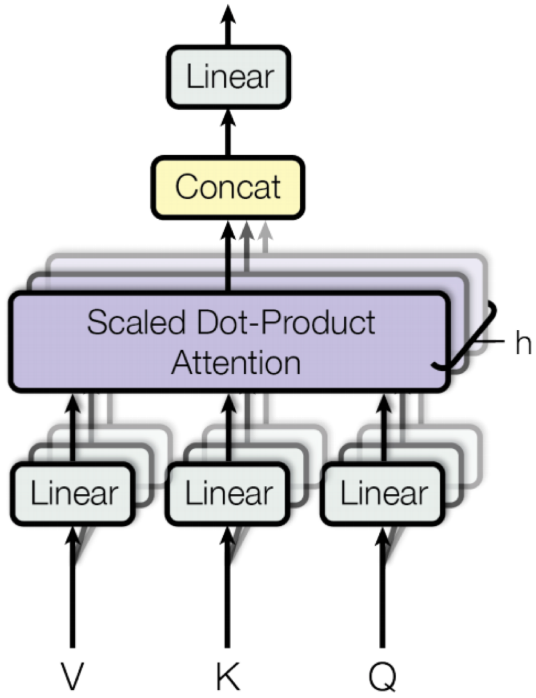
Attention mechanism

- *Scaled dot product attention*:

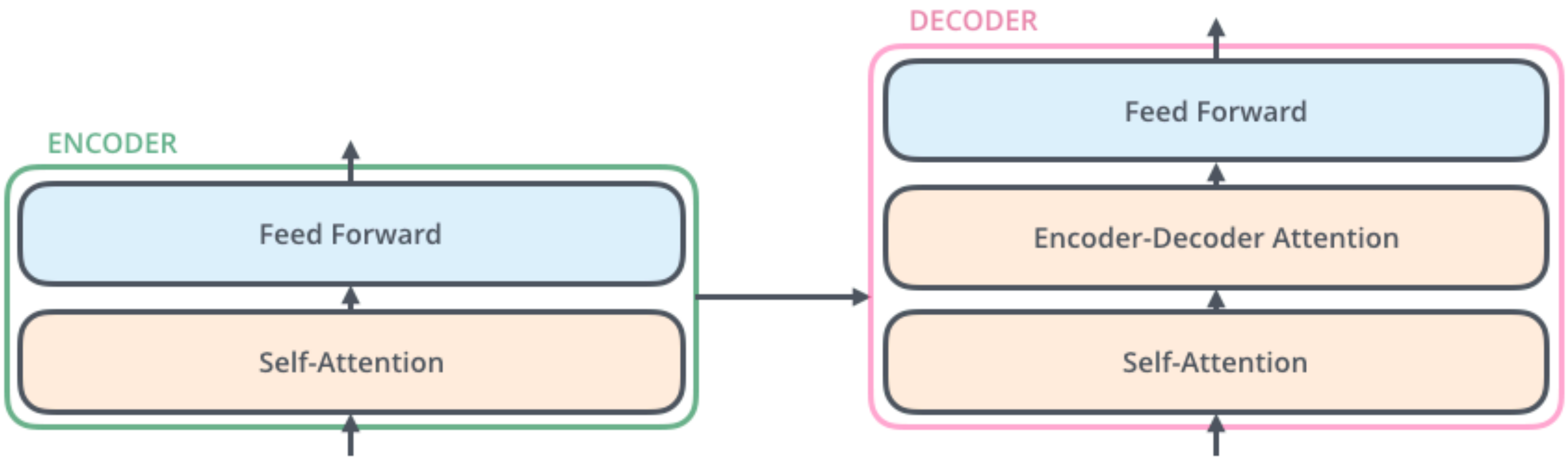
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

  - Q, K, V are matrices with rows corresponding to queries, keys, and values, d_k is the dim. of the keys

- *Multi-head attention*: run h attention models in parallel on top of different linearly projected versions of Q, K, V; concatenate and linearly project the results



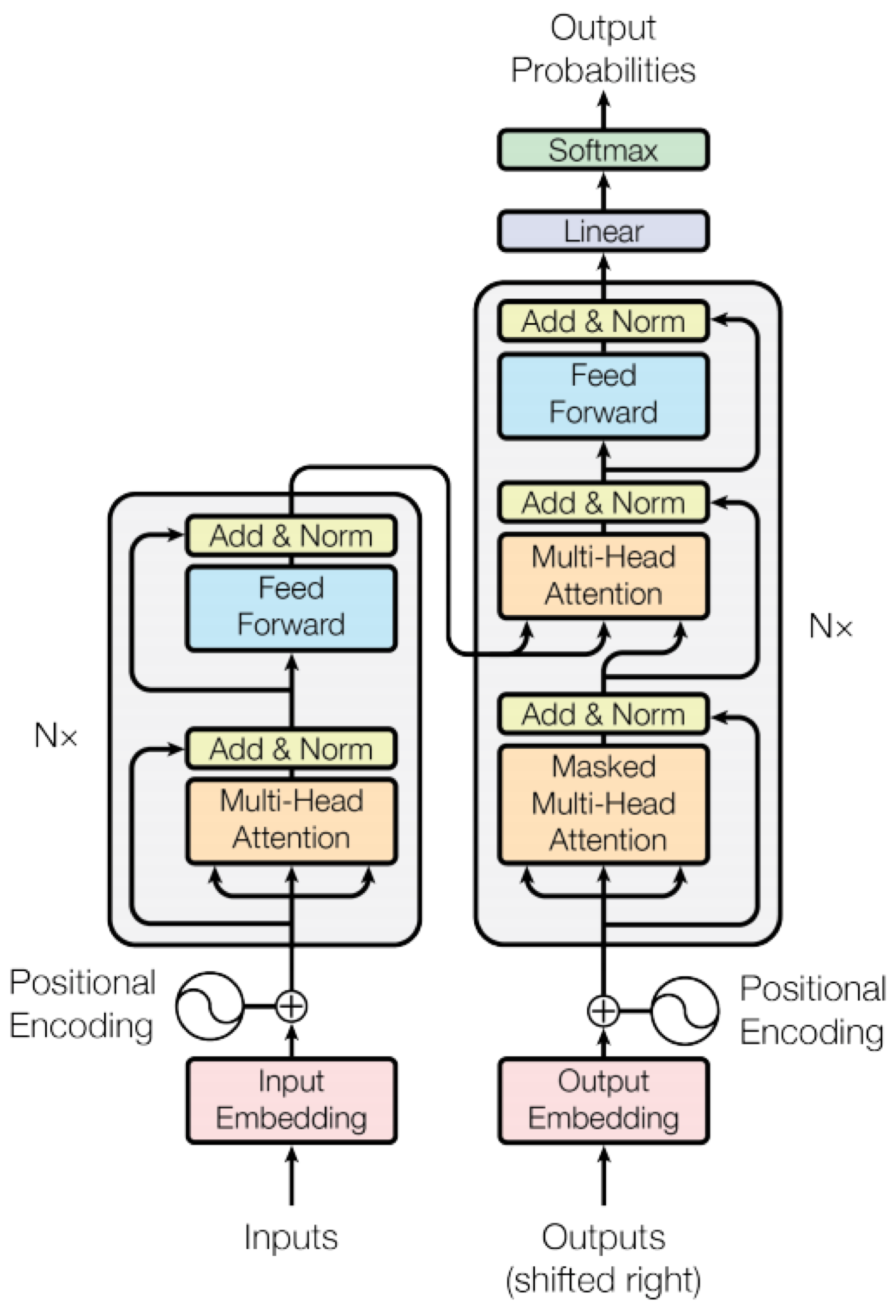
Attention mechanism



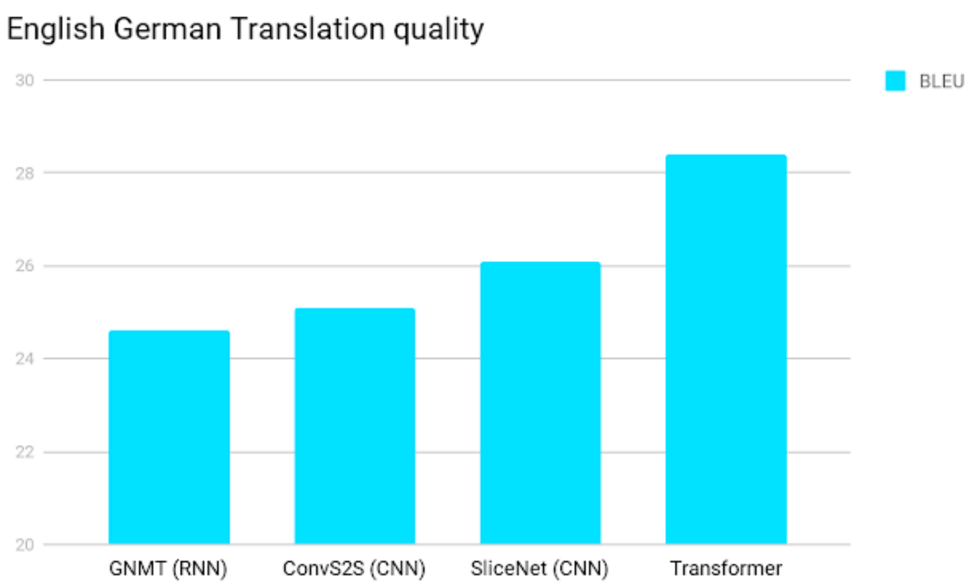
- *Encoder-decoder attention*: queries come from previous decoder layer, keys and values come from output of encoder
- *Encoder self-attention*: queries, keys, and values come from previous layer of encoder
- *Decoder self-attention*: values corresponding to future outputs are masked out

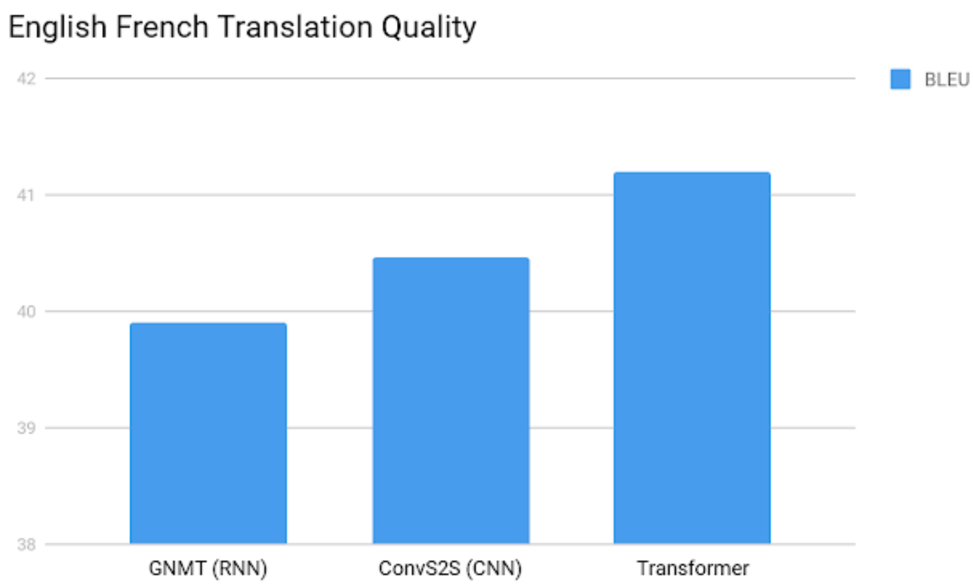
http://jalammar.github.io/

Transformer architecture in detail



Results





https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html

Other ideas

Training NLP – requires aligned datasets (of ImageNet variety)

Universal Language Model Fine-tuning for Text Classification *Jeremy Howard ,Sebastian Ruder∗*

How to finetune the language layers and classifier layers (architectures, loss functions, dropout)

Better word embeddings trained directly from language models

Deep contextualized word representations *Matthew E. Peters† , Mark Neumann† , Mohit Iyyer† , Matt Gardner*

Better ways to represent vectors ELMo vector assigned to a token or word is actually a function of the entire sentence containing that word. Therefore, the same word can have different word vectors under different contexts.

ELMo word top of a two-layer bidirectional language model (biLM). This biLM model has two layers stacked together.

Trained in unsupervised way

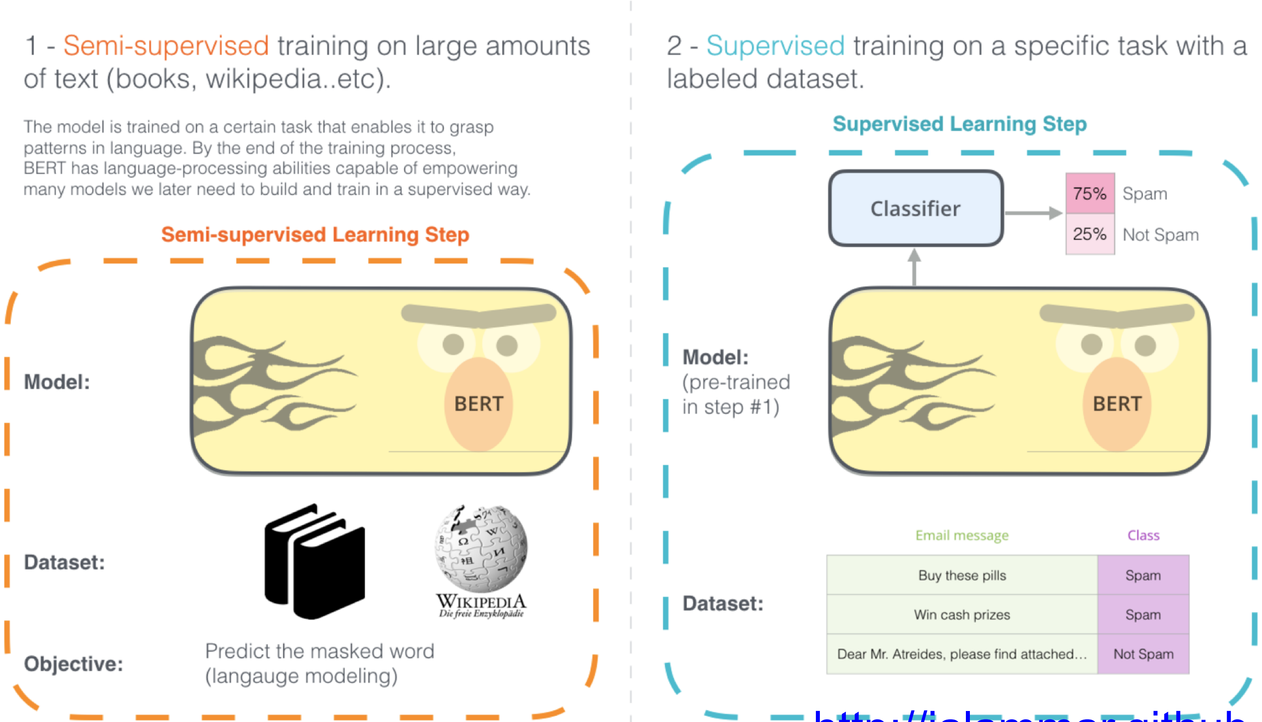
ELMO


Hey ELMo, what's the embedding of the word "stick"?
There are multiple possible embeddings! Use it in a sentence.
Oh, okay. Here: "Let's stick to improvisation in this skit"
Oh in that case, the embedding is: -0.02, -0.16, 0.12, -0.1 ….etc

BERT

Bidirectional **E**ncoder **R**epresentations from **T**ransformers

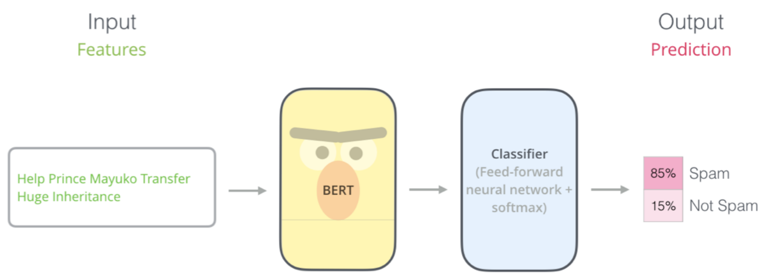
- Pretrain NLP representations
- Universal language models which can be adapted to many language tasks
- Seq2Seq models + attention – good to machine translation
- What about other language tasks ?



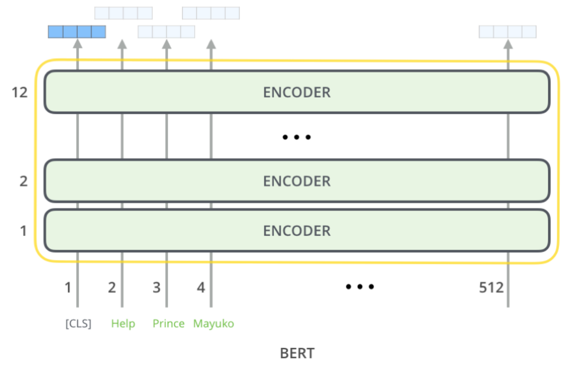
http://jalammar.github.io/illustrated-bert/

It can be used for different tasks

- Spam/not spam
- Fact checking fact/no fact
- Sentiment analysis positive/not
- Visual question answering



- Bert pretrained encoder of the transformer
- For classification – focus only on output in the first token – feed to feedforward NN



Parting thoughts

- Methodology for text generation problems
  - Evaluation is tricky
  - Maximum likelihood training is not the most appropriate (but alternatives involve optimizing non-differentiable objectives)
- Attention appears to be a game-changer for NMT (for image captioning, not as much)
  - But there is much more to MT than attention (dealing with unknown words, etc.)
- Recurrent architectures are not the only option for sequence modeling
  - Convolutional and feedforward alternatives should be considered