NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE

ICLR(2015)

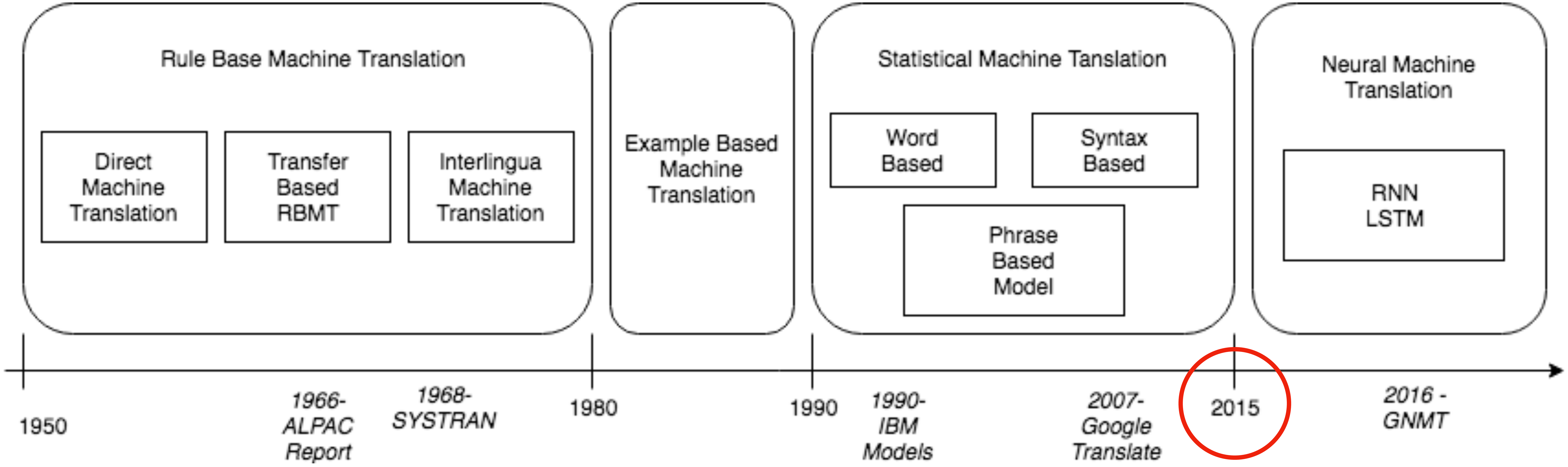
Dzmitry Bahdanau, KyungHyun Cho, Yoshua Bengio

Jun-Hyung Lee

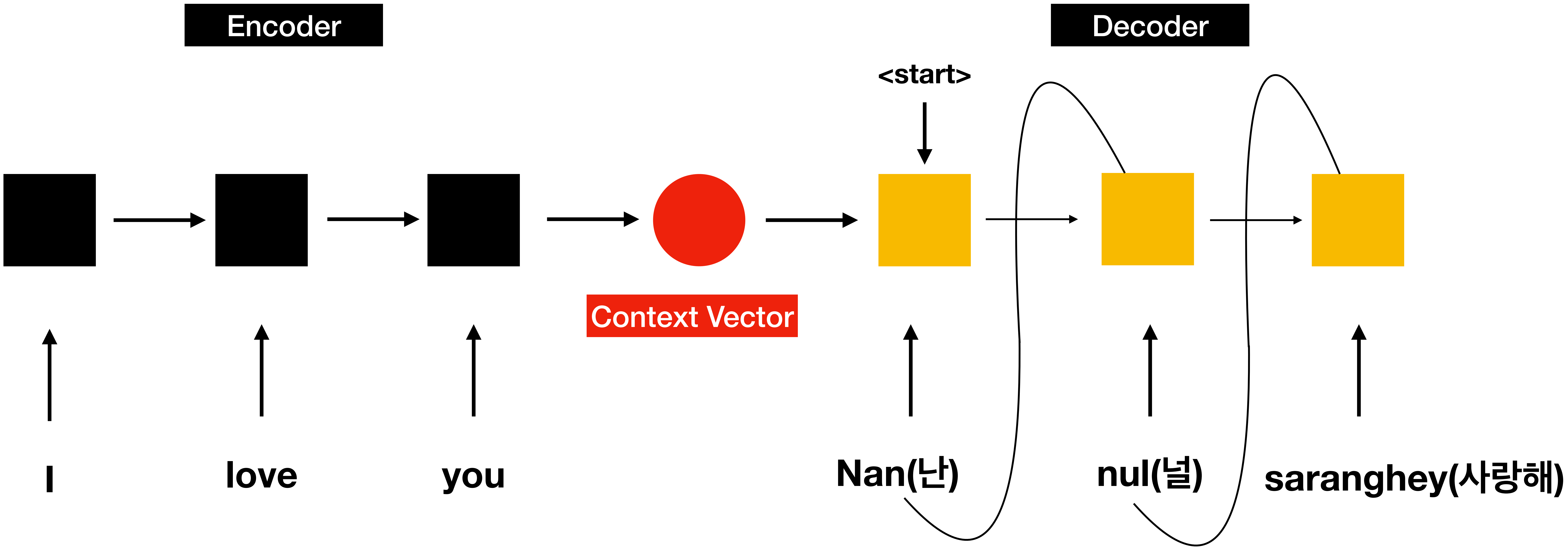
Overview

- History
- Overall
- Main Contributions
- Architecture
- Experiment
- Conclusion

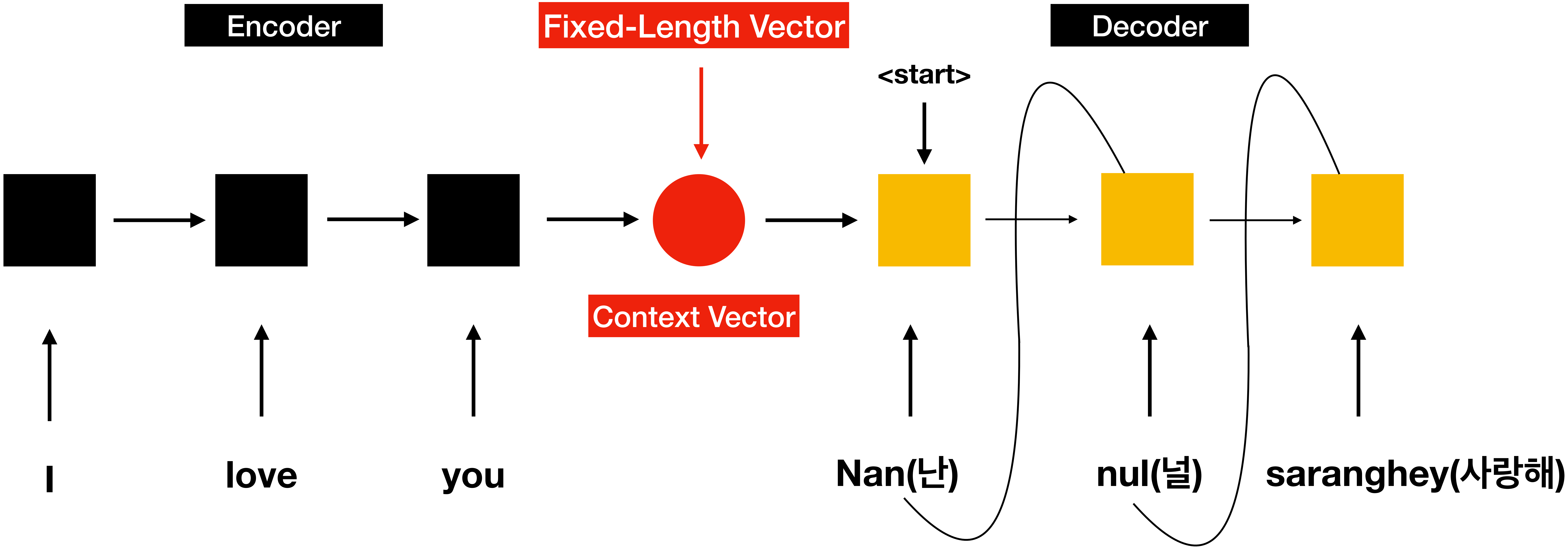
History


Rule Base Machine Translation
Direct Machine Translation
Transfer Based RBMT
Interlingua Machine Translation
Example Based Machine Translation
Statistical Machine Tanslation
Word Based
Syntax Based
Phrase Based Model
Neural Machine Translation
RNN LSTM
1950
1966- ALPAC Report
1968- SYSTRAN
1980
1990
1990- IBM Models
2007- Google Translate
2015
2016 - GNMT

Previous Encoder-Decoder



Previous Encoder-Decoder



Motivation

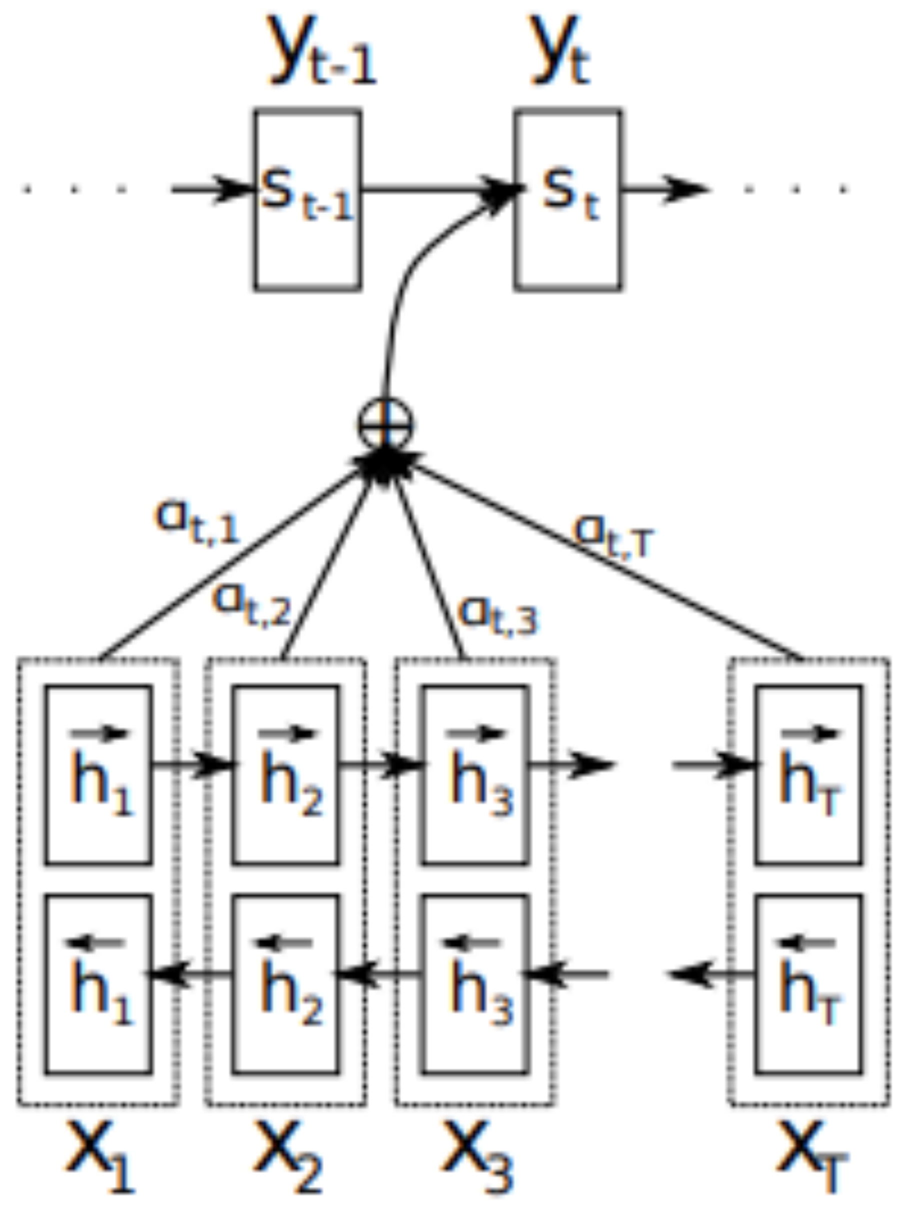


Figure 1: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence $(x_1, x_2, \ldots, x_T)$.

Decoder

Attention

Encoder

- [our model] does not attempt to encode a whole input sentence into a single fixed-length. Instead, it encodes the input sentence into a sequence of vectors and chooses a subset of these vectors adaptively while decoding the translation

- Each time the proposal model generates a word in a translation, it (soft-)searches for a set of positions in a source sentence where the most relevant information is concentrated

- Interpreted in another way, the **attention mechanism** is simply giving the network access to its internal memory, which is the hidden state of the encoder

- With this new approach the information can be spread throughout the sequence of annotations, which can be selectively retrieved by the decode accordingly.

Main Contributions

- **The encoder: a bidirectional RNN**
  - the hidden state should encode information from both the previous and following words
- **The decoder: (attention) model**
  - attention mechanism: a weighted sum of the input hidden states

Abbreviations

f = LSTM function

g = non-linear

q = forward RNN

alpha = feedforward NN

c = sequence of hidden-states (context vector)

j = encoder’s time-step

i = decoder’s time-step

Align = attention

Annotation = hidden-state

Seq2Seq with Attention - Encoder

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

$$a(s_{i-1}, h_j) = v_a^\top \tanh\left(W_a s_{i-1} + U_a h_j\right),$$

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

Weighted sum 이용

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

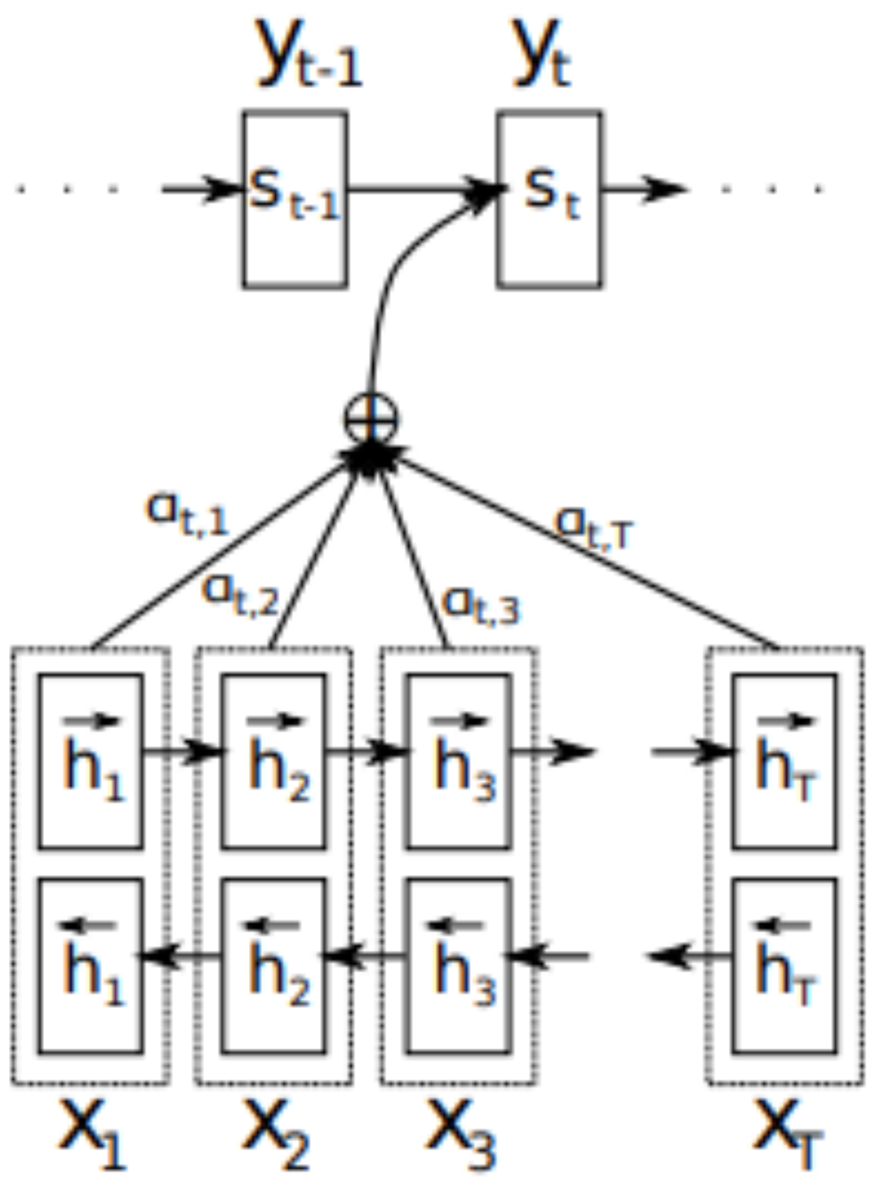


Figure 1: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence $(x_1, x_2, \ldots, x_T)$.

Decoder

Attention

Encoder

Seq2Seq with Attention - Decoder

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

$$s_i = f(s_{i-1}, y_{i-1}, c_i) = (1 - z_i) \circ s_{i-1} + z_i \circ \tilde{s}_i,$$

$$\tilde{s}_i = \tanh\left(We(y_{i-1}) + U\left[r_i \circ s_{i-1}\right] + Cc_i\right),$$

$$z_i = \sigma\left(W_z e(y_{i-1}) + U_z s_{i-1} + C_z c_i\right),$$
$$r_i = \sigma\left(W_r e(y_{i-1}) + U_r s_{i-1} + C_r c_i\right),$$

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

$$p(y_i | s_i, y_{i-1}, c_i) \propto \exp\left(y_i^\top W_o t_i\right),$$

$$t_i = \left[\max\left\{\tilde{t}_{i,2j-1}, \tilde{t}_{i,2j}\right\}\right]_{j=1,\ldots,l}^\top$$

$$\tilde{t}_i = U_o s_{i-1} + V_o E y_{i-1} + C_o c_i.$$

Beam Search

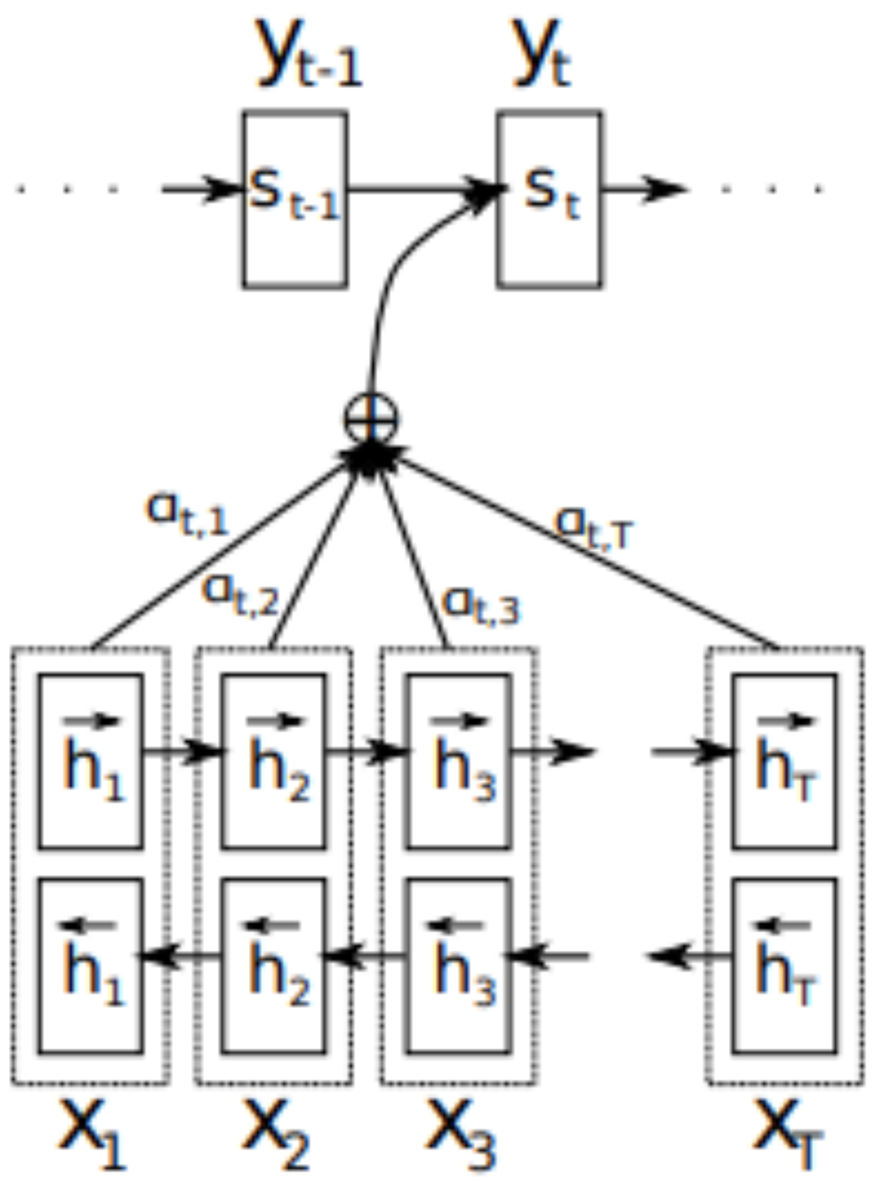


Figure 1: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence $(x_1, x_2, \ldots, x_T)$.

Decoder

Attention

Encoder

Experiment

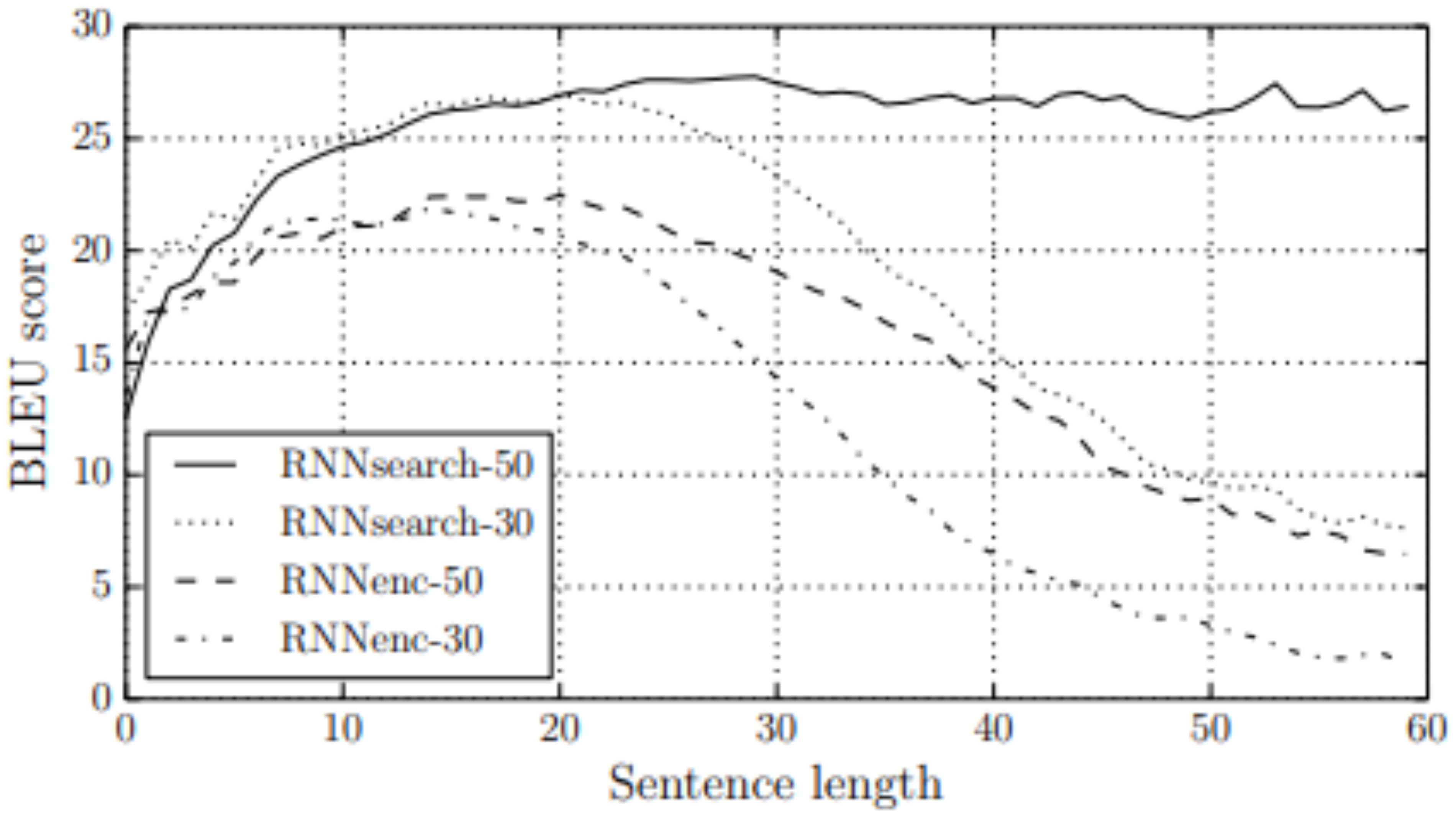


Figure 2: The BLEU scores of the generated translations on the test set with respect to the lengths of the sentences. The results are on the full test set which includes sentences having unknown words to the models.

- English-to-French translation.
- Bilingual,parallet corpora provided by ACL WMT'14
- Europal(61M words), news commentary(5.5M),UN(421M) and two crawled corpora of 90M and 272.5M words respectively, totaling 850M words.
- Reduced the size of the combined corpus to have 348M words.
- 1000 hidden units(GRU), minibatch SGD, Parameter update using Adadelta, SGD update with 80 minibatch

Qualitative Analysis

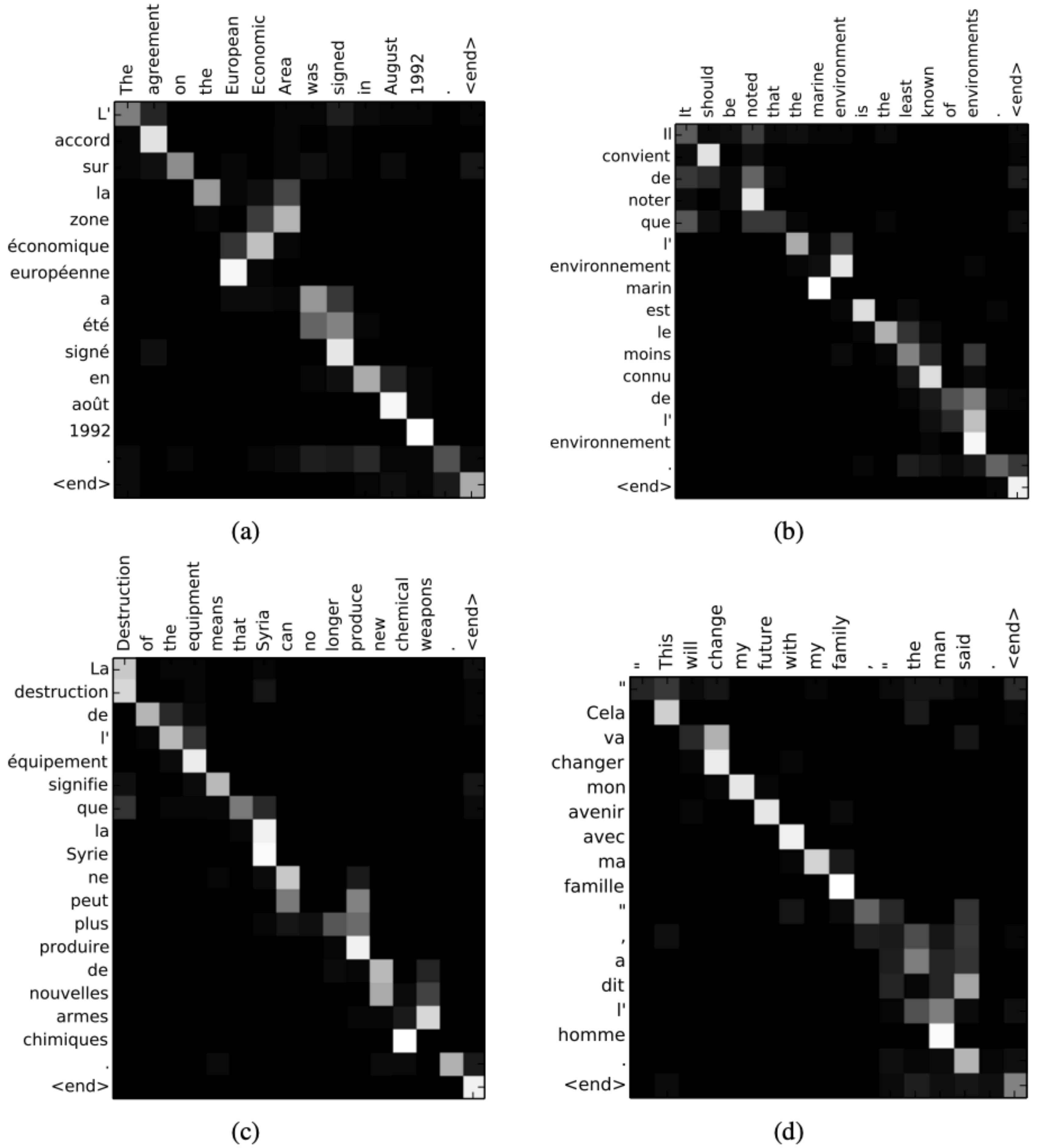


Figure 3: Four sample alignments found by RNNsearch-50. The x-axis and y-axis of each plot correspond to the words in the source sentence (English) and the generated translation (French), respectively. Each pixel shows the weight α_{ij} of the annotation of the j-th source word for the i-th target word (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.

- [the man] -> [l’homme]
- hard alignment [the] -> [l] , [man] -> [homme]
- the -> [l] , [le], [la], [les] (?)

Conclusion

- seq2seq by letting a model soft-search for a set of input words
- Annotations computed by an encoder, when generating each target word

  - free from source sentence fixed length

  - focus on information relevant to the generation of the text target word

Discussions

- How much improvement comes from attention? How much comes from bidirectional-ness?

  - Luong et al. shows that unidirectional LSTMs could perform just as well

- Why is softmax needed? Is it always desirable?

References

- https://arxiv.org/pdf/1409.0473.pdf
- https://heiwais25.github.io/nlp/2019/06/18/neural-machine-translation-by-jointly-learning-to-align-and-translate/
- https://www.youtube.com/watch?v=WsQLdu2JMgI&t=578s&ab_channel=MinsukHeo%ED%97%88%EB%AF%BC%EC%84%9D