NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE

By Dzmitry Bahdanau, KyungHyun Cho, Yoshua Bengio

Jieru Zhang

S1719048

University of Edinburgh

Overview

- Introduction
  - Why NMT
  - Why attention
- Body
  - Encoder-decoder with attention
  - How attention works
  - Experiments design
- Results
- Conclusion
- Discussion

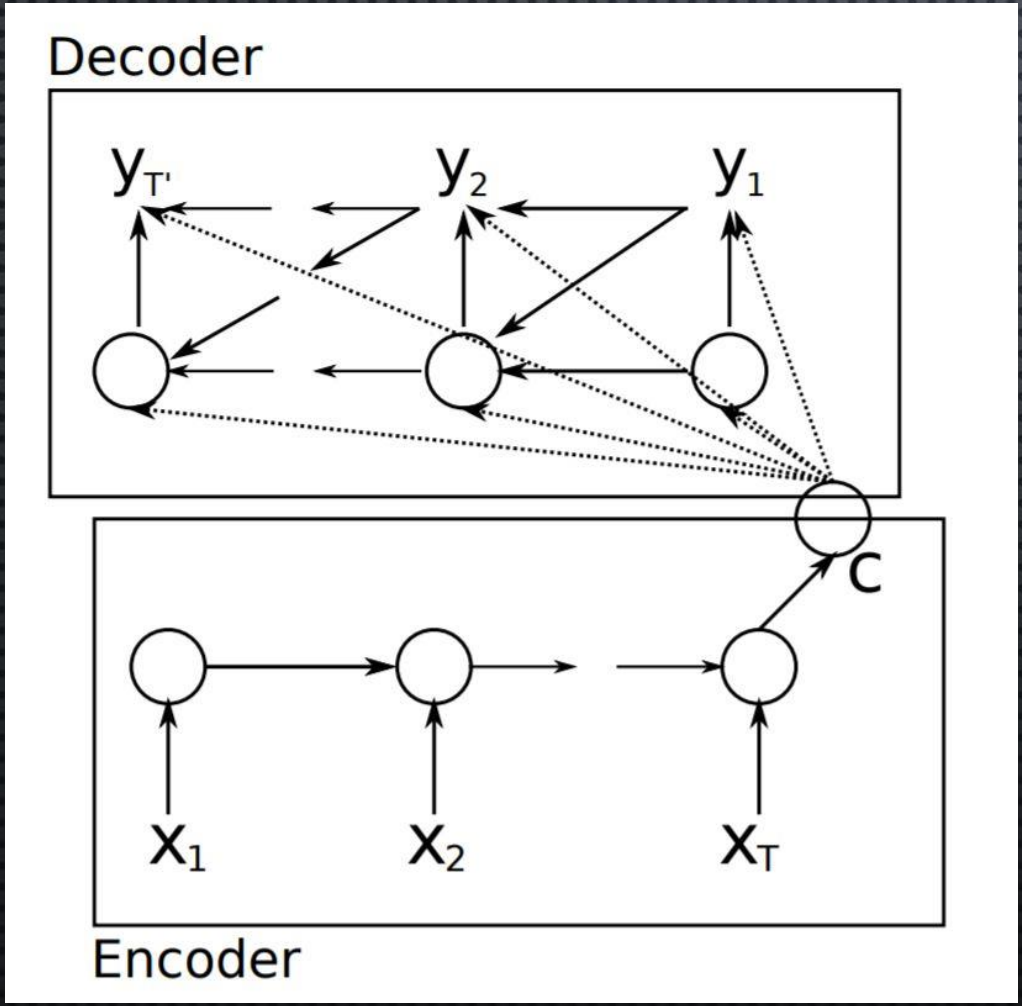
Introduction – why NMT?

Stage 1: Rule-based Machine Translation (RBMT)

Stage 2: Statistical machine translation (SMT)

Stage 3: Neural machine translation (NMT)

Introduction – vanilla encoder-decoder



(Cho et al., 2014)

$p(y_i \mid y_1, \ldots, y_{i-1}, \boldsymbol{x}) = \mathrm{g}(y_{i-1}, s_i, c)$

Target: variable length sentence

Context vector: fixed-length internal representation

Source: variable length sentence

Introduction – why attention?

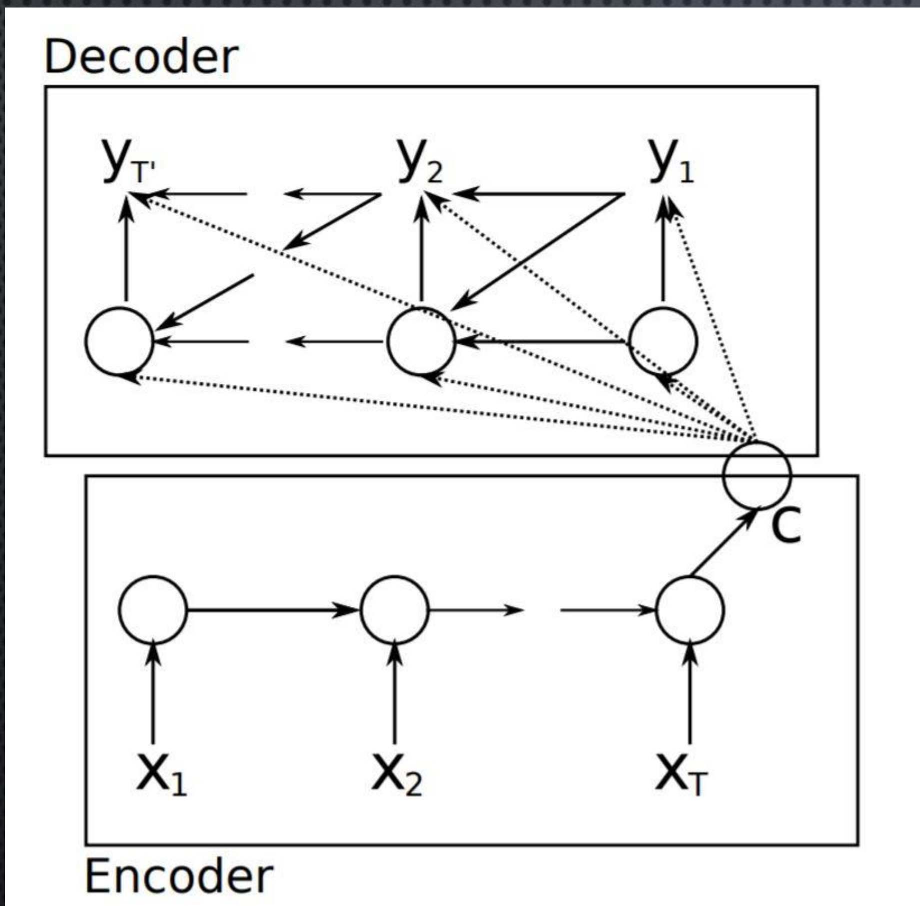
Problem

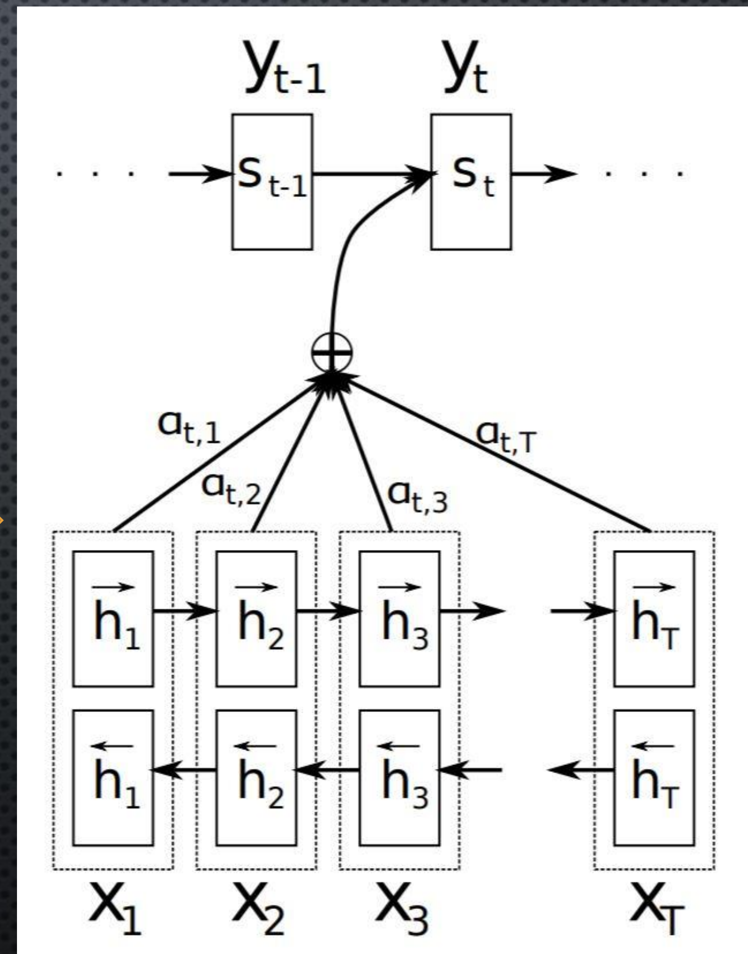
Fixed-length context vector might lose information about the source sentence.

Solution

Attention mechanism: decide where to put attention on the source sentence when decoding each target word.

Encoder-decoder with attention

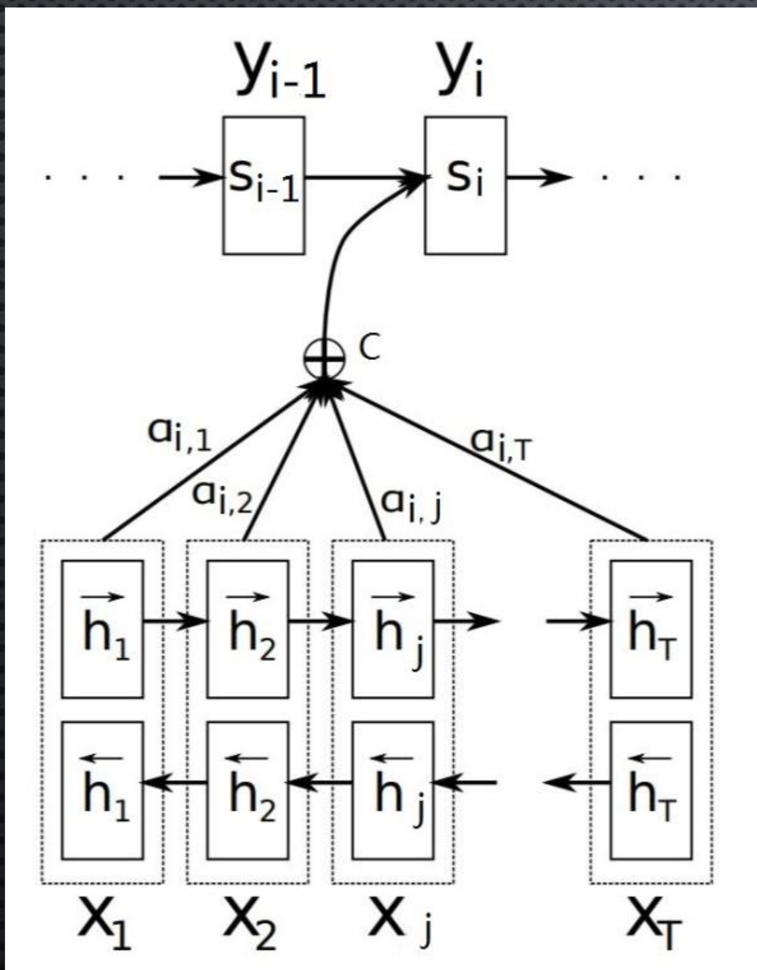




$p(y_i \mid y_1, \dots, y_{i-1}, \boldsymbol{x}) = \mathrm{g}(y_{i-1}, s_i, c)$

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

How attention works?



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

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

$$c_i = \sum_i^T \alpha_{ij}\, h_j$$

$$\alpha_{ij} = softmax\ (a(s_{i-1}, h_j))$$

h_j = concatenate$(\overrightarrow{h_j}, \overleftarrow{h_j})$

Model architecture

- Word embedding: 620 dimension.
- Encoder: BiRNN, each with 1000 gated hidden units.
- Decoder: RNN with 1000 gated hidden units.
- Output layer: multi-layered output function with maxout 500 units.
- Softmax, minibatch SGD, beam search.

Experiments design

- Data
  - English-French parallel corpora from WMT'14
- Experiments details
  - Four models:
    - RNNencdec-30
    - RNNencdec-50

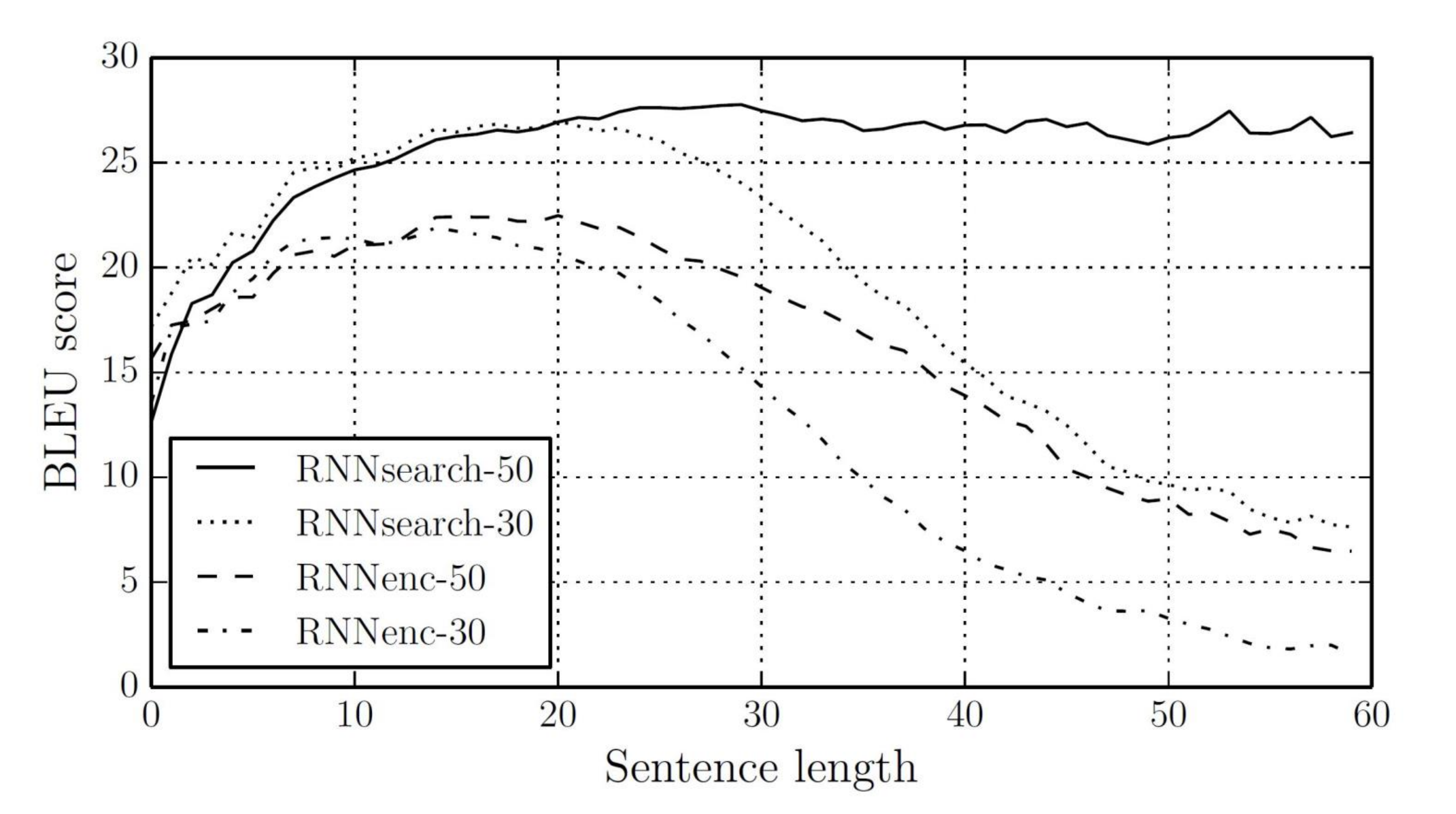
      RNN encoder-decoder (Cho et al., 2014)
    - RNNsearch-30
    - RNNsearch-50

      Bidirectional encoder, and decoder with attention

Results – BLEU score

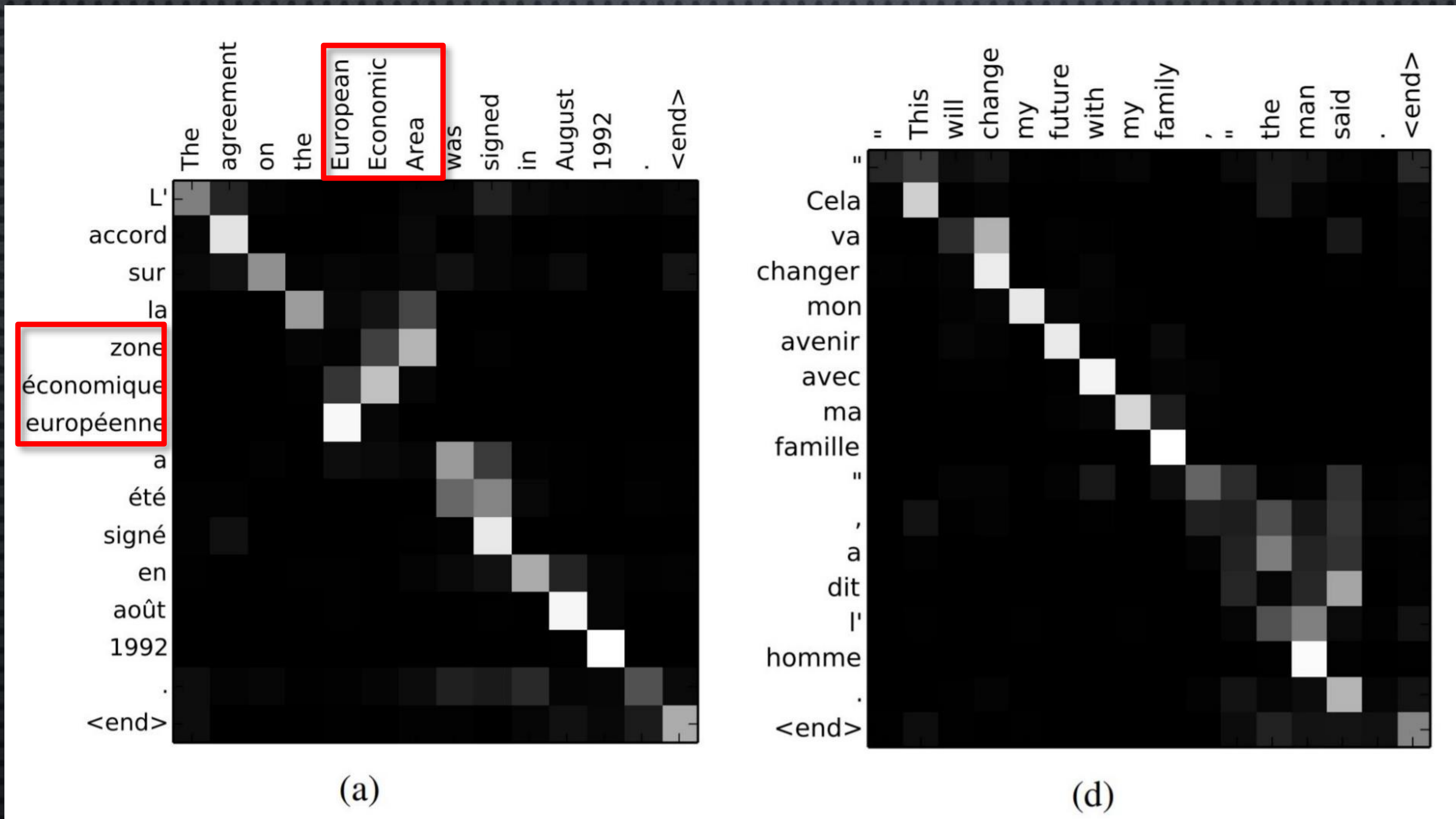
Model	All	No UNK°
RNNencdec-30	13.93	24.19
RNNsearch-30	21.50	31.44
RNNencdec-50	17.82	26.71
RNNsearch-50	26.75	34.16
RNNsearch-50*	28.45	36.15
Moses	33.30	35.63

Results



30
25
20
15
10
5
0
BLEU score
0
10
20
30
40
50
60
Sentence length
RNNsearch-50
RNNsearch-30
RNNenc-50
RNNenc-30

Results – RNNsearch-50



Conclusion

- RNN Encoder-decoder with attention
  - Free model from encoding source sentence to a fixed-length vector, performs better on longer sentences.
  - The alignment could align target word with relevant source word.
  - Comparable to the phrase-based statistical MT.
  - The whole system could be jointly trained, including the alignment model.

Reference

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104-3112).

THANKS

Discussion

- Are there some problems caused by attention?
- What's the differences between the effects of gated hidden units and attention?
- After alignment, the context vector is not "fixed length", does this mean the actual length of context vector is variable?
- Deep RNN helps here? LSTM units help here?
  - "Sequence to Sequence Learning with Neural Networks"