Sequence to Sequence Learning with Neural Networks

Ilya Sutskever, Oriol Vinyals, Quoc V. Le (NIPS'14)

단국대학교 모바일시스템공학과 양윤성

Contents

1. Introduction
2. The Model
3. Experiments
4. Conclusion

1-(1). Prologue

Introduction to proposed architecture

- Translation methods using deep learning
- Encoder-decoder structure of multi-layer LSTM
- Sequence to Sequence
- Based technology on natural language processing
- Reversed order of input

1-(2). Statistical Machine Translation

Translation method before deep learning

$$P(is|My\ \ name) = P(My\ \ name\ \ is)/P(My\ \ name)$$

- Count-based approach
- Choose the most likely one that the word can be
- Required for all probability sentences → Need very large DB
- Difficulty to understand the context of sentences (such as word order changes)

1-(3). Traditional RNN Translation

Neural Machine Translation(NMT)

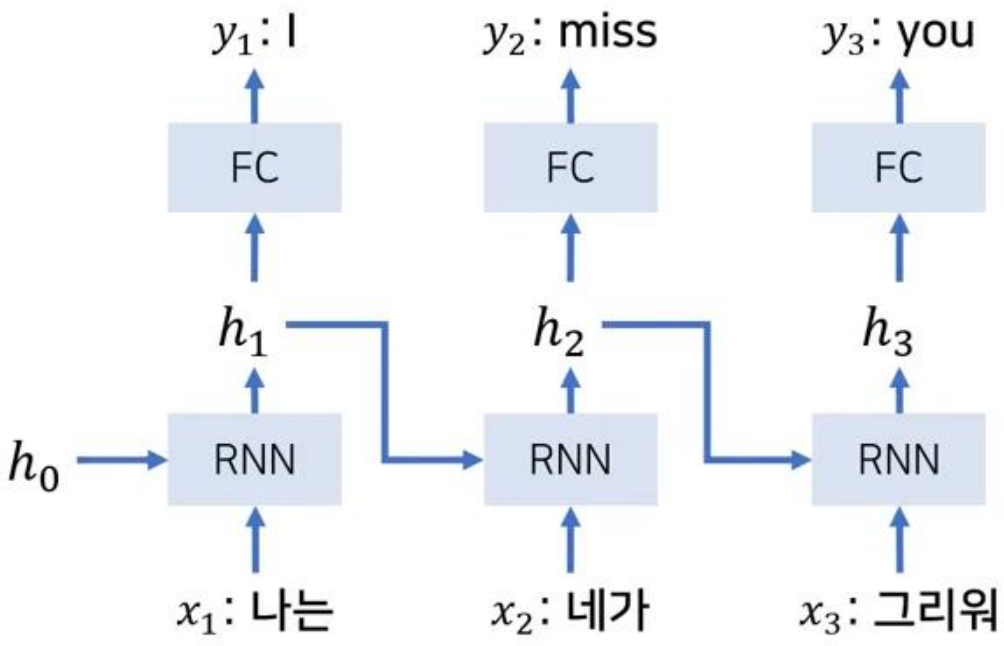
- Encoder-Decoder Framework

Input : (x1, x2, ⋯, xT)

Output : (y1, y2, ⋯, yT)

Input length = Output length

$$h_t = \mathrm{sigm}\left(W^{\mathrm{hx}} x_t + W^{\mathrm{hh}} h_{t-1}\right)$$
$$y_t = W^{\mathrm{yh}} h_t$$



1-(3). Traditional RNN Translation

What is the problem?

- The input and output must be the same size
  - 오늘 어때? → How are you?
  - Need to change input sequence size
- Long-term dependencies problem
  - Remember previous information only in a hidden state → no loss prevention
  - Therefore, it is difficult to predict and translate exact value
  - LSTM : Additional memory(cell state) and control gates

Contents

1. Introduction
2. The Model
3. Experiments
4. Conclusion

2-(1). Basic Concepts

Language Model Features

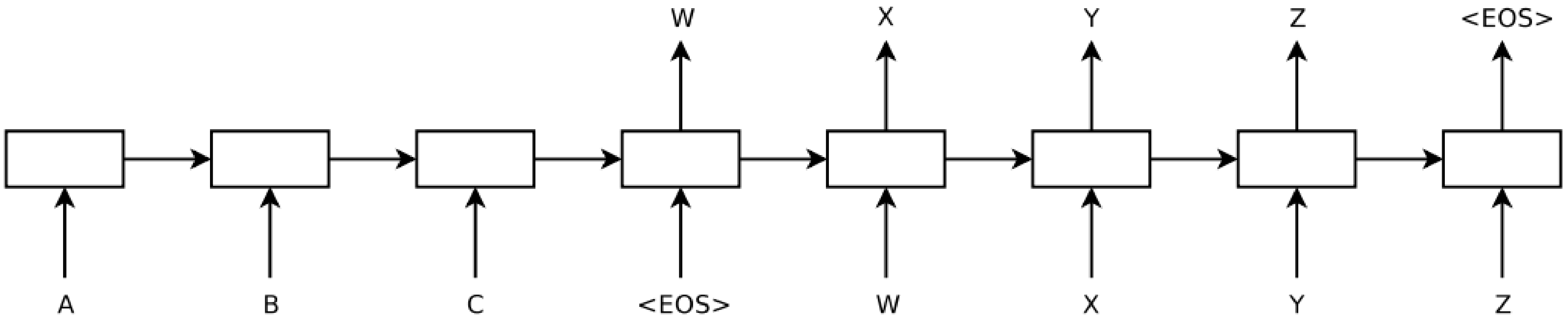
- A sentence(W) consists of several words(w1, w2, ⋯)
- Joint Probability : P(W) = P(w1, w2, ⋯, wn)
- Chain Rule : It can be divided by the product of the conditional probability.

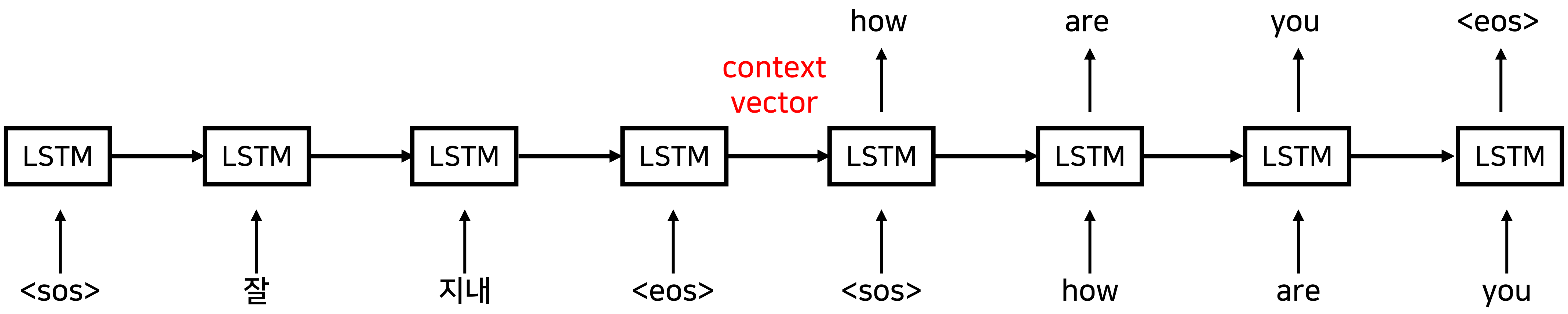
$$P(w_1, w_2, w_3, \dots, w_n) = P(w_1) * P(w_2|w_1) * P(w_3|w_1, w_2), \dots, P(w_n|w_1, w_2, \dots, w_{n-1})$$

$$= \prod_{i=1}^{n} P(w_i|w_1, \dots, w_{i-1})$$

$$P(I\ go\ to\ school) = P(I, go, to, school) =$$
$$P(I) * P(go|I) * P(to|I, go) * P(school|I, go, to)$$

2-(2). Seq2Seq Model





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

2-(3). Embedding

- Vectorize the meaning of a word (Input of encoder & decoder)
- So that computers can understand natural language
- It becomes possible to calculate the similarity of words

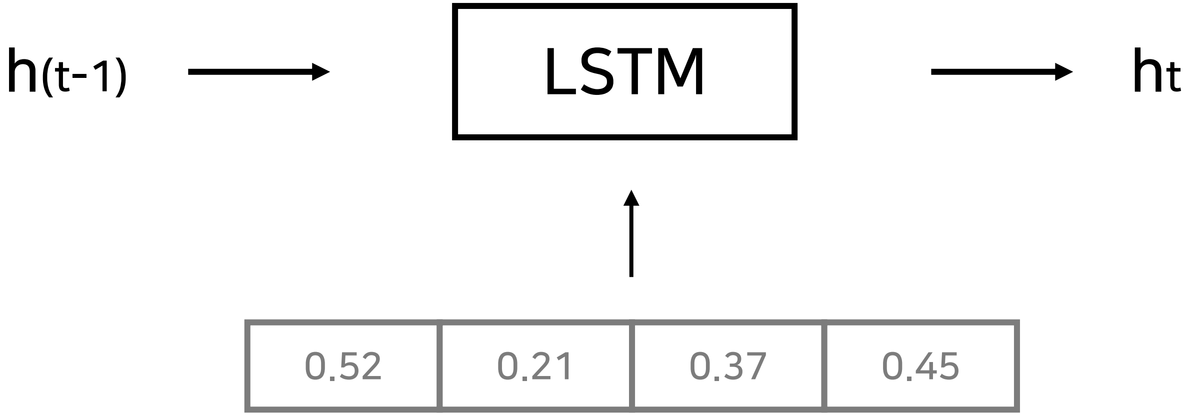
n-demension

python	0.52	0.21	0.37	...
ruby	0.48	0.21	0.33	...
word	0.05	0.23	0.06	...

Process first-time words to use similarity
and reduce translation time

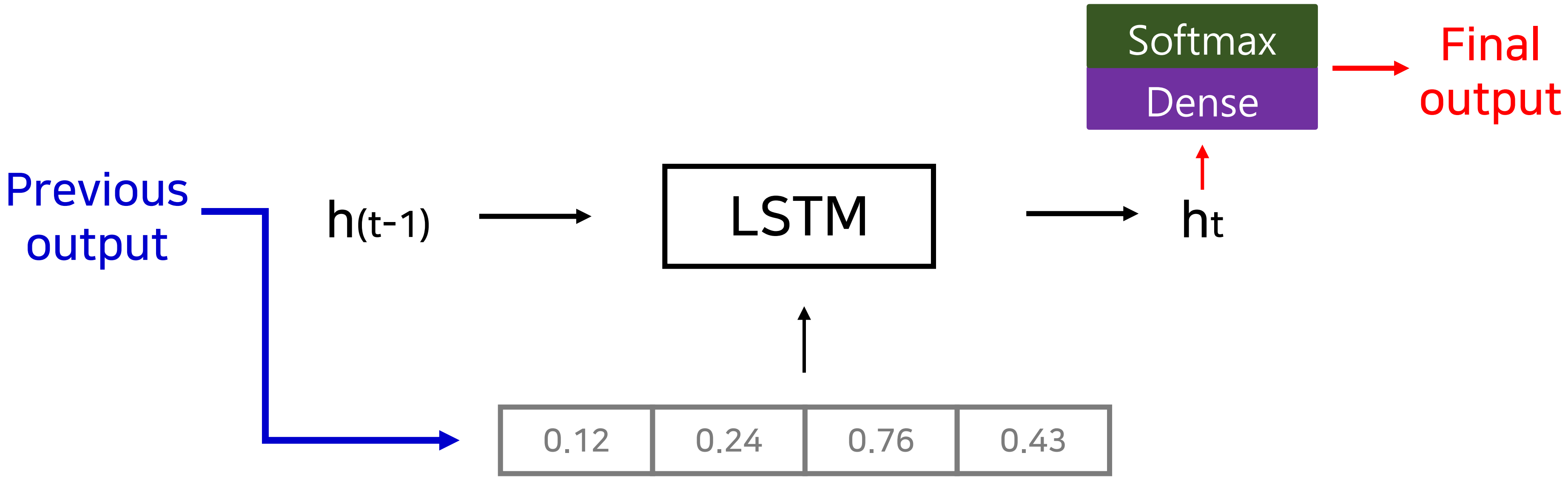
2-(4). Encoder

- Enter embedded word sequentially
- Operate word and previous hidden state into LSTM → Update hidden state
- The last hidden state becomes the context vector



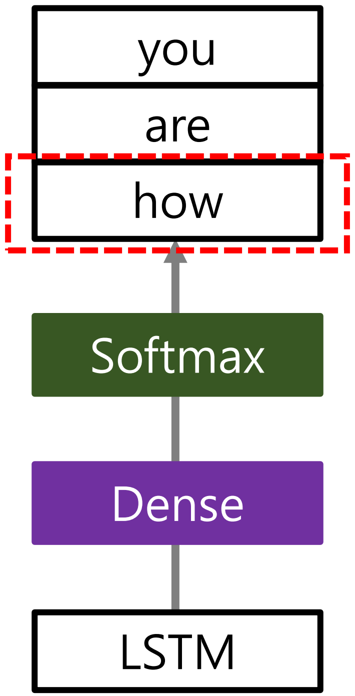
2-(5). Decoder

- Enter embedded word sequentially
- Operate word and previous hidden state into LSTM → Update hidden state
- Dense, Softmax are need

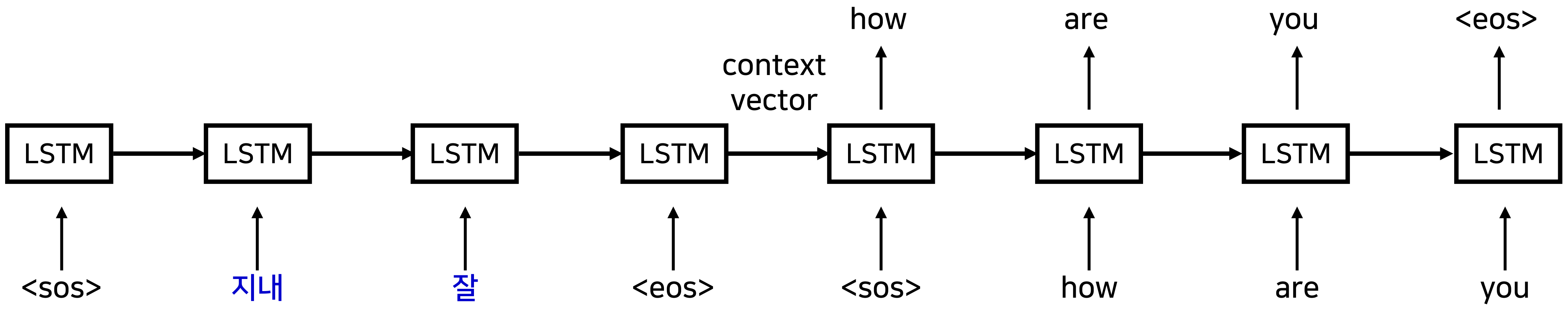


2-(6). Dense & Softmax Layer

- Only need decoder
- Dense Layer : Linear transformation for input data
- Softmax Layer : Calculate the final probability of values
  - Normalize between 0 and 1
  - The highest probability word will be final choice



2-(7). Reverse Input Sequence



- Close distance between w1 in the input W and w1 in the output W
  → Increased association
- Decrease the efficiency of the last word?
  → But it's still more efficient

- Learning difficulty ▼ learning efficiency and translation accuracy ▲

Contents

1. Introduction
2. The Model
3. Experiments
4. Conclusion

3-(1). Decoding and Rescoring

$\mathcal{S}$: Training Set

S : Sequence

T : Translation Result

$$1/|\mathcal{S}| \sum_{(T,S)\in\mathcal{S}} \log p(T|S)$$

- Learning so that one input sequence S can produce one-on-one matching output sequence T
- Use logs to maximize probability
- Multiply by 1/|S| for normalization

3-(1). Decoding and Rescoring

S : Sequence

T : Translation Result

$$\hat{T} = \arg\max_{T} p(T|S)$$

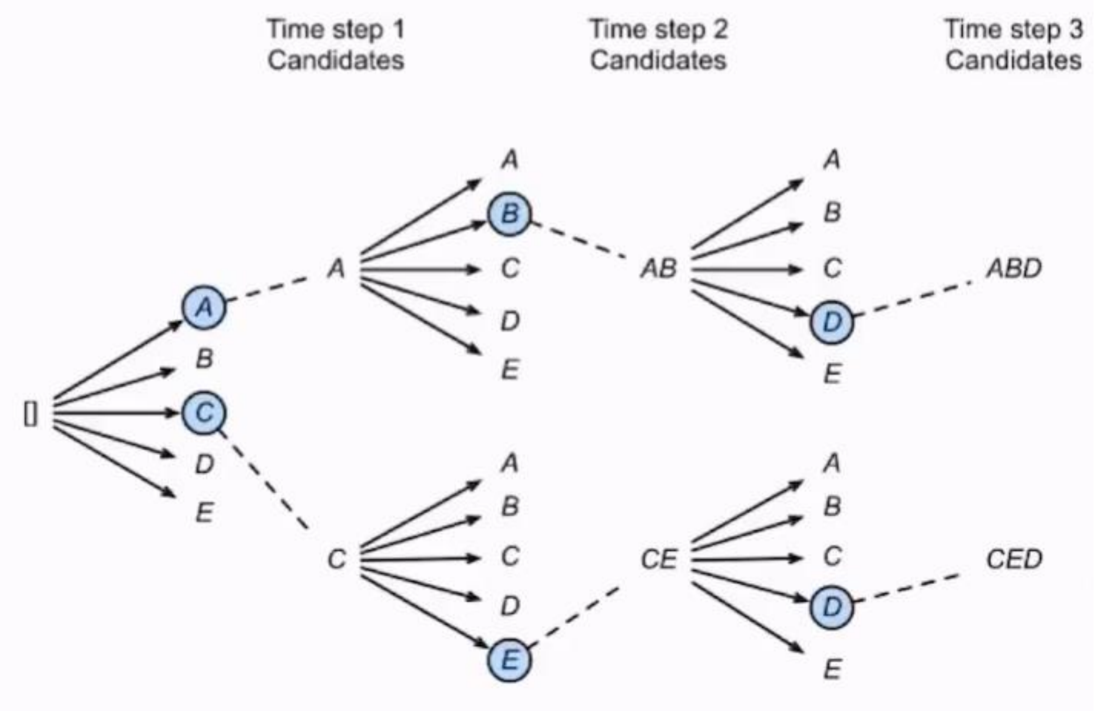
- Once learning is complete, look for the most likely translation
- Using left-to-right beam search method

3-(2). Beam Search

- Exhaustive Search : Explore the number of all cases
- Greedy Search : Unable to restore one's choice

Beam size : 5

- Explore the most probable translation results at each stage
- Normalization → To correctly compare sentences of different lengths
- When <EOS>, remove it from beam and confirm the set



3-(3). BLEU Score

- Indicator of machine translation quality
- How consistent are the results compared to the actual translation

$$BLEU = \min(1, \frac{length\_of\_prediction}{length\_of\_reference})(\prod_{i=1}^{4} precision_i)^{\frac{1}{4}}$$

- Precision
- Clipping
- Brevity Penalty

3-(3). BLEU Score

Precision & clipping $$\left(\prod_{i=1}^{4} precision_i\right)^{\frac{1}{4}}$$

EX) 1-gram

Prediction s : The more decomposition the more flavor the food has

Answer s : The more the merrier I always say

- 1-gram precision : 5/9
- 1-gram precision + clipping (Deduplication) : 3/9
- Multiply 1-gram result to 4-gram result and square ¼

3-(3). BLEU Score

Brevity Penalty

$$\min(1, \frac{length\_of\_prediction}{length\_of\_reference})$$

Prediction s : The more decomposition

Answer s : The more the merrier I always say

Brevity Penalty : min(1, 3/7) = 3/7

- To prevent increase/decrease of translation due to length
- If it's shorter than the answer, multiply len(prediction)/len(reference)
  - A kind of correction factor

3-(4). Results

Method	test BLEU score (ntst14)
Bahdanau et al. [2]	28.45
Baseline System [29]	33.30
Single forward LSTM, beam size 12	26.17
Single reversed LSTM, beam size 12	30.59
Ensemble of 5 reversed LSTMs, beam size 1	33.00
Ensemble of 2 reversed LSTMs, beam size 12	33.27
Ensemble of 5 reversed LSTMs, beam size 2	34.50
Ensemble of 5 reversed LSTMs, beam size 12	**34.81**

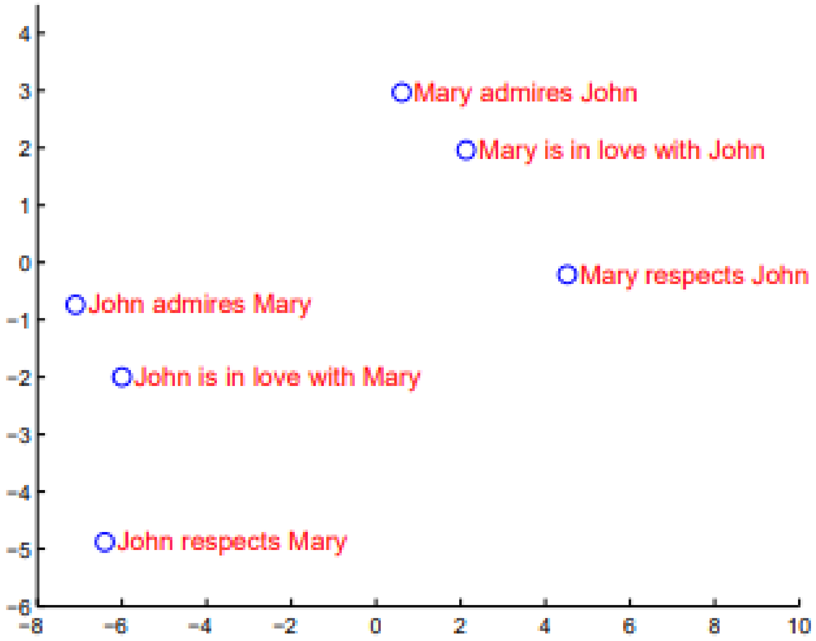
- Forward LSTM BLEU 26.17 → Reversed LSTM BLEU 30.59
- Add Ensemble and more beam Size → Increasing BLEU

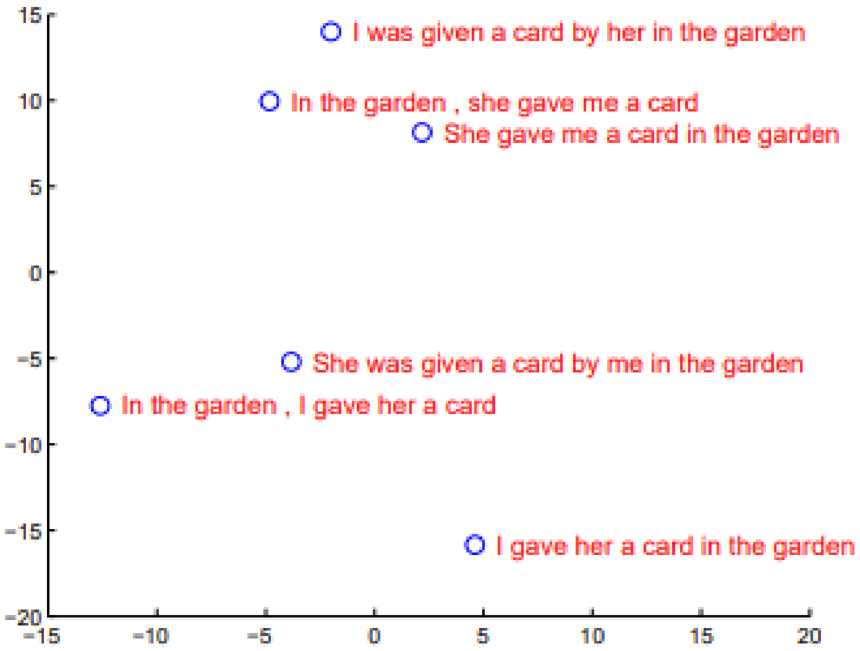
3-(4). Results

Method	test BLEU score (ntst14)
Baseline System [29]	33.30
Cho et al. [5]	34.54
Best WMT'14 result [9]	**37.0**
Rescoring the baseline 1000-best with a single forward LSTM	35.61
Rescoring the baseline 1000-best with a single reversed LSTM	35.85
Rescoring the baseline 1000-best with an ensemble of 5 reversed LSTMs	**36.5**
Oracle Rescoring of the Baseline 1000-best lists	$\sim$45

- To combine LSTM with STM using WMT'14 dataset
- Best result is 36.5 (Reversed + SMT + Ensemble)

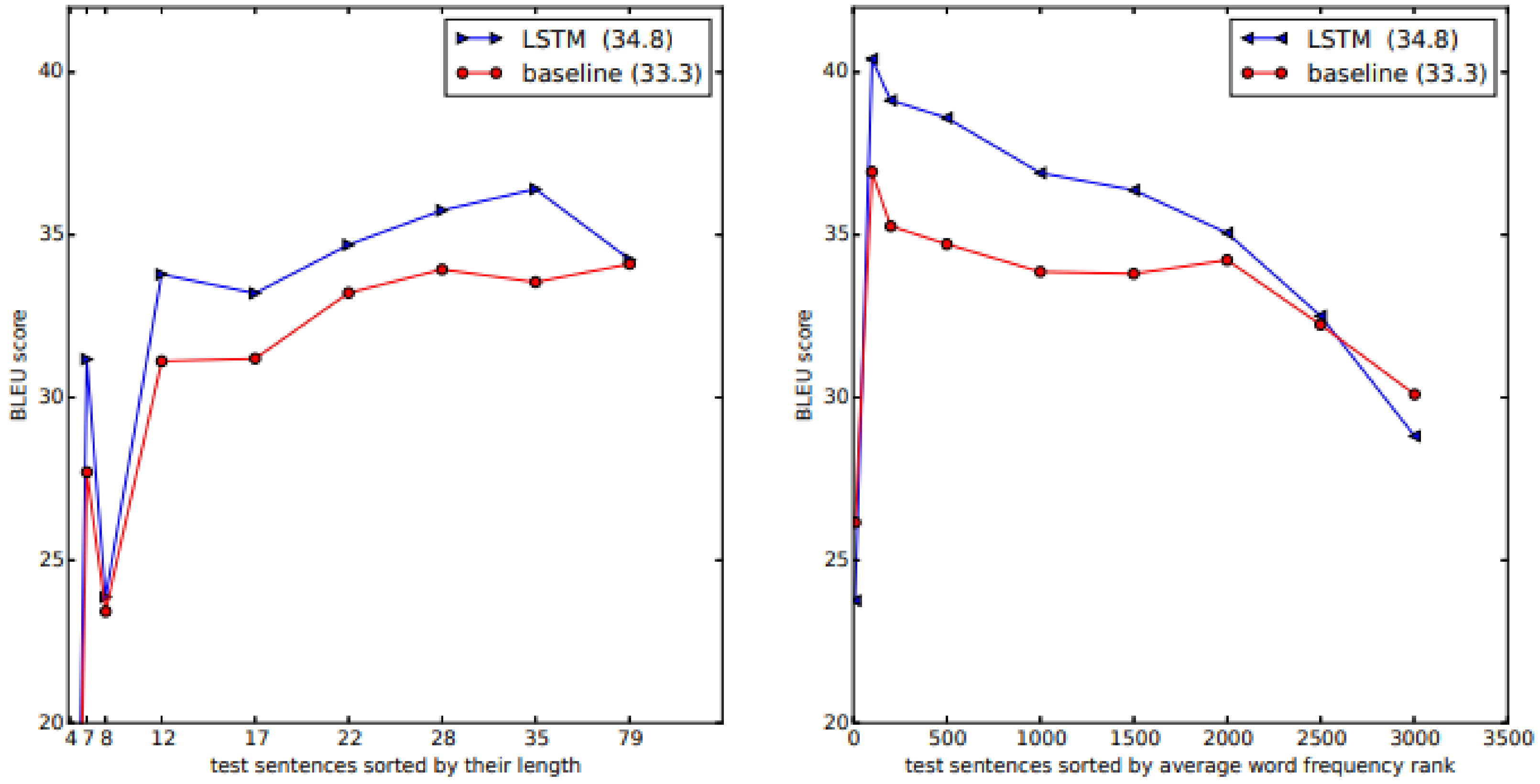
3-(4). Results





- Dimension and visualization of embedded results with encoder
- Confirmation of good classification by meaning

3-(4). Results



BLEU is more stable than baseline even if sentence length is longer

Contents

1. Introduction
2. The Model
3. Experiments
4. Conclusion

4. Conclusion

- Better translation performance than traditional SMT
- Some ideas to improve performance
  - LSTM
  - Reversing the words in the source sentences
  - Beam-search approach
- Disadvantages of fixed vector size → Curious about the improved structure

Thank You!

Sequence to Sequence Learning with Neural Networks