Sequence to Sequence Learning With Neural Networks

Hya Sutskever
Google

Oriol Vinyals
Google

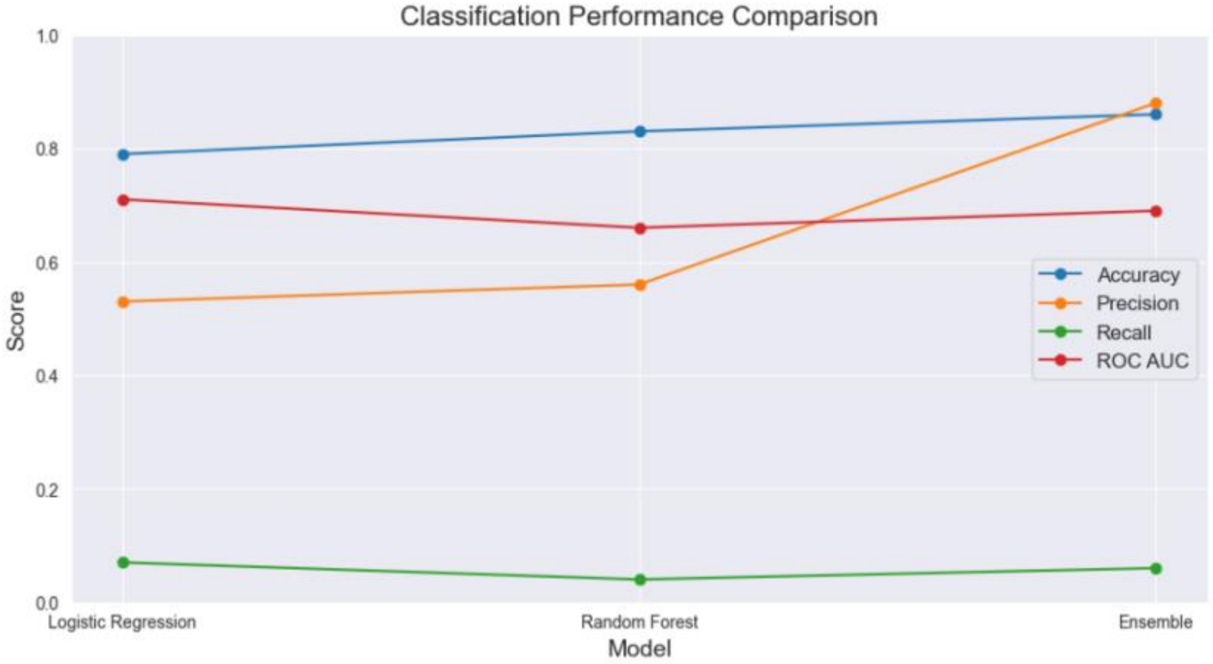
Quoe V. Le
Google

Department of Big Data in Software Engineering, Hallym University
Tae Hoon-Her

SMOTE

Factors that can reduce the accuracy of a dataset

- Overfitting
- Imbalanced Data
- Feature Selection
- Mislabelled Labels
- Noise
- Missing Data

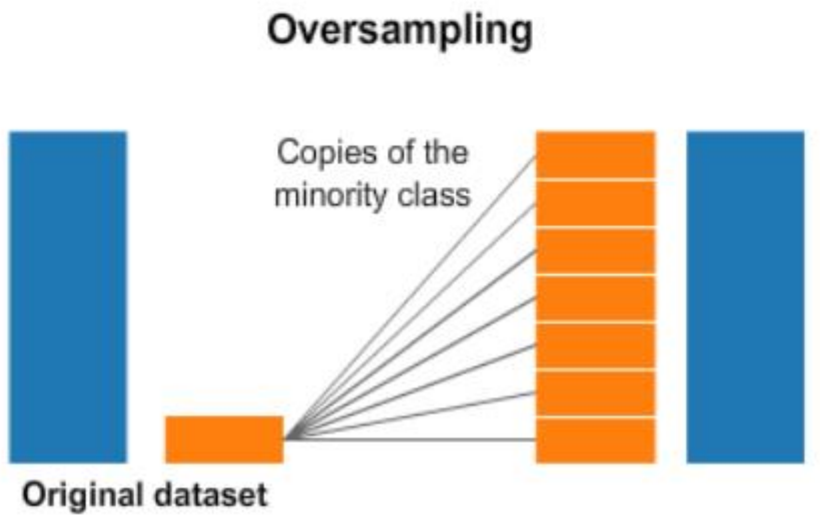


Before Start

Oversampling

Synthetic Minority Over-Sampling Technique

- Using K-NN Algorithm



SMOTE

Synthetic Minority Over-Sampling Technique

- Using K-NN Algorithm

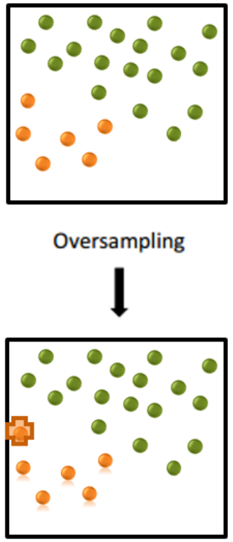
Advantage

- Less risk of overfitting compared to simple random oversampling

Why?

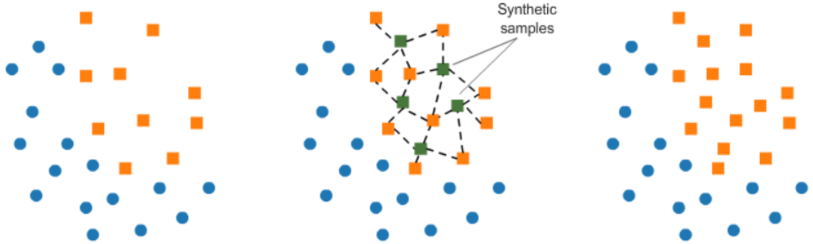
Oversampling

- Randomly increases the number of data samples in the minority class
- If the generated samples are dissimilar to the original samples, it may cause overfitting

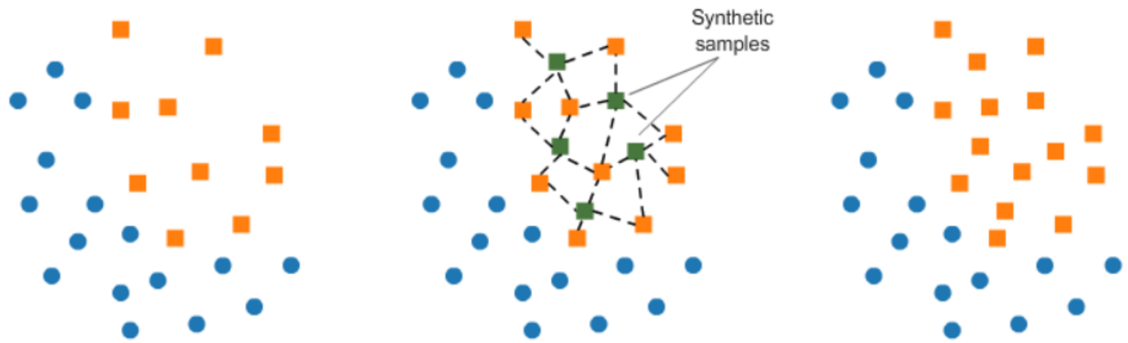


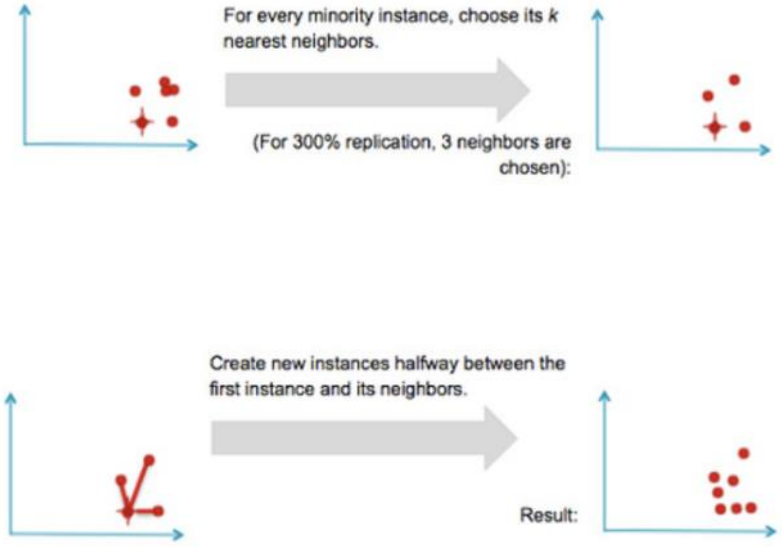
SMOTE

- Calculating the difference between samples in the minority and majority class
- Creating new samples that have intermediate values between minority and majority class samples
- As a result, the new samples are likely to have similar characteristics to the original data
- Maintaining data diversity



SMOTE working principle




For every minority instance, choose its *k* nearest neighbors.

(For 300% replication, 3 neighbors are chosen):

Create new instances halfway between the first instance and its neighbors.

Result:

- Randomly select one sample from the minority class data.
- Find k nearest neighbors of the selected sample. Euclidean distance is typically used to measure the distance.
- Randomly choose one of the neighboring samples.

Sequence to Sequence Learning With Neural Networks

Hya Sutskever
Google

Oriol Vinyals
Google

Quoe V. Le
Google

Department of Big Data in Software Engineering, Hallym University
Tae Hoon-Her

Before Seq-to-Seq

SMT

- When the structure of the input and output is relatively simple, effective English translation can be achieved
- Disadvantage
  - Cannot consider sentence structure
  - Difficult to judge whether the translation result is natural or not.

Seq-to-Seq

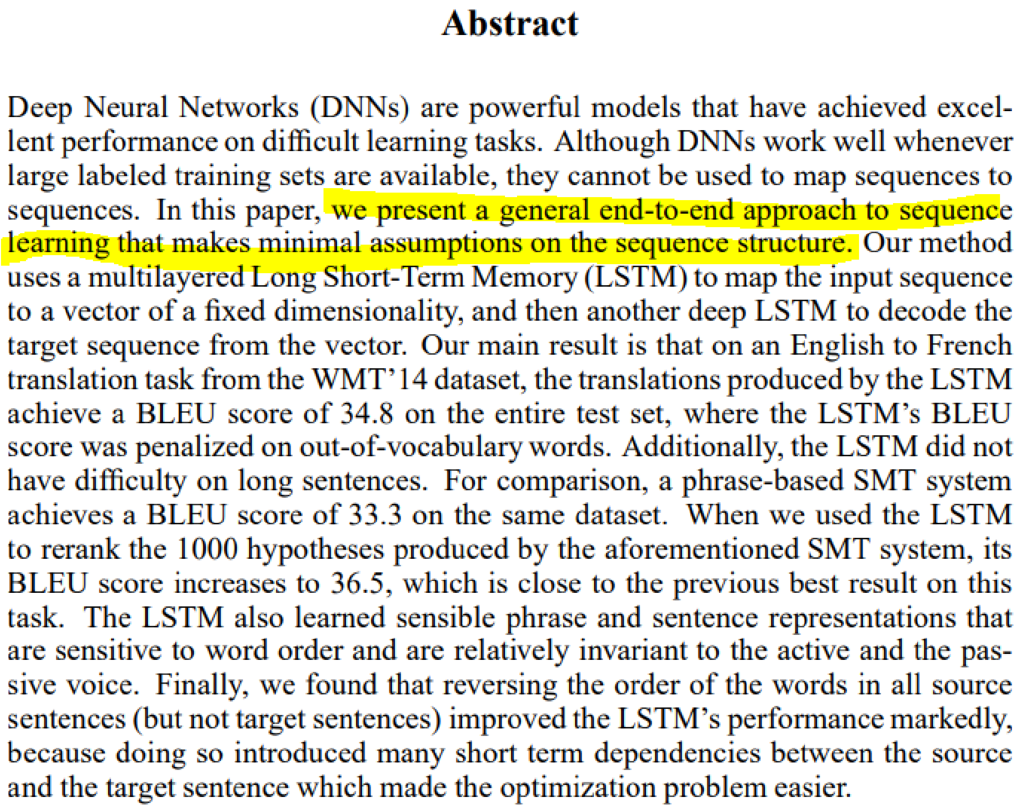
- "Neural Machine Translation by Jointly Learning to Align and Translate" first introduced the concept.
- In machine translation tasks, the attention mechanism was introduced into the RNN-based encoder-decoder model
- Generating an output sequence that takes into account all information of the input sequence.

Abstract

Important sentence from the Abstract

We present a general end-to-end approach to sequence learning that makes minimal assumptions on the sequence structure

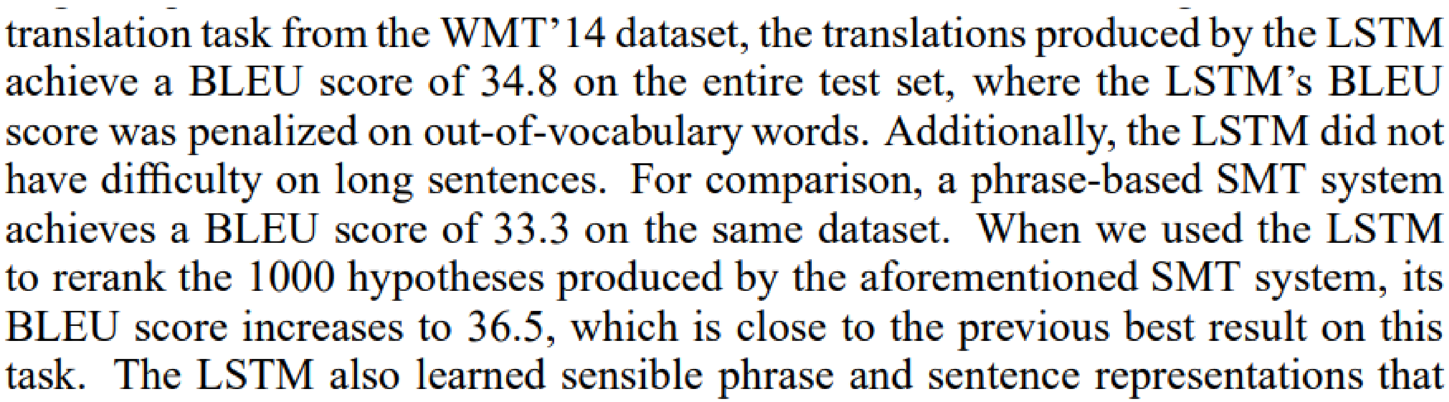
In tasks where the structure or length of the input and output sequences are different, or the relationship between the input and output is irregular, it is difficult to use



Abstract

Deep Neural Networks (DNNs) are powerful models that have achieved excellent performance on difficult learning tasks. Although DNNs work well whenever large labeled training sets are available, they cannot be used to map sequences to sequences. In this paper, we present a general end-to-end approach to sequence learning that makes minimal assumptions on the sequence structure. Our method uses a multilayered Long Short-Term Memory (LSTM) to map the input sequence to a vector of a fixed dimensionality, and then another deep LSTM to decode the target sequence from the vector. Our main result is that on an English to French translation task from the WMT'14 dataset, the translations produced by the LSTM achieve a BLEU score of 34.8 on the entire test set, where the LSTM's BLEU score was penalized on out-of-vocabulary words. Additionally, the LSTM did not have difficulty on long sentences. For comparison, a phrase-based SMT system achieves a BLEU score of 33.3 on the same dataset. When we used the LSTM to rerank the 1000 hypotheses produced by the aforementioned SMT system, its BLEU score increases to 36.5, which is close to the previous best result on this task. The LSTM also learned sensible phrase and sentence representations that are sensitive to word order and are relatively invariant to the active and the passive voice. Finally, we found that reversing the order of the words in all source sentences (but not target sentences) improved the LSTM's performance markedly, because doing so introduced many short term dependencies between the source and the target sentence which made the optimization problem easier.

Abstract

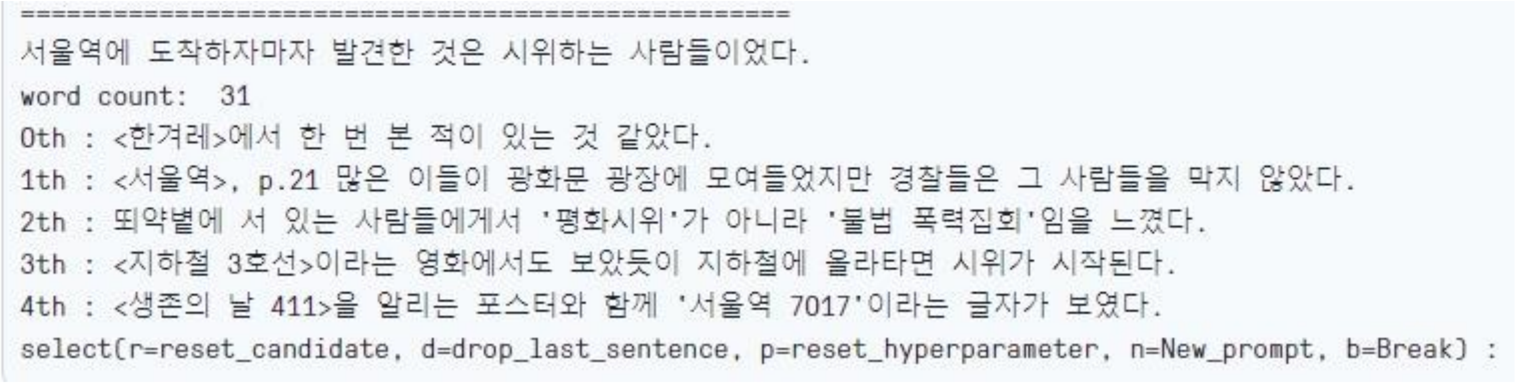


translation task from the WMT’14 dataset, the translations produced by the LSTM achieve a BLEU score of 34.8 on the entire test set, where the LSTM’s BLEU score was penalized on out-of-vocabulary words. Additionally, the LSTM did not have difficulty on long sentences. For comparison, a phrase-based SMT system achieves a BLEU score of 33.3 on the same dataset. When we used the LSTM to rerank the 1000 hypotheses produced by the aforementioned SMT system, its BLEU score increases to 36.5, which is close to the previous best result on this task. The LSTM also learned sensible phrase and sentence representations that

Seq2Seq : 34.8

SMT : 33.3

Seq2Seq + SMT = 36.5



```
==========================================================
서울역에 도착하자마자 발견한 것은 시위하는 사람들이었다.
word count:  31
0th : <한겨레>에서 한 번 본 적이 있는 것 같았다.
1th : <서울역>, p.21 많은 이들이 광화문 광장에 모여들었지만 경찰들은 그 사람들을 막지 않았다.
2th : 뙤약볕에 서 있는 사람들에게서 '평화시위'가 아니라 '불법 폭력집회'임을 느꼈다.
3th : <지하철 3호선>이라는 영화에서도 보았듯이 지하철에 올라타면 시위가 시작된다.
4th : <생존의 날 411>을 알리는 포스터와 함께 '서울역 7017'이라는 글자가 보였다.
select(r=reset_candidate, d=drop_last_sentence, p=reset_hyperparameter, n=New_prompt, b=Break) :
```

Example

I love you

- 모델이 학습해야하는 연관성이 줄어든다 → 학습이 더 수월하다
- I love you 같은 문장을 번역할 때, 일반적으로는 I 가 먼저 입력되고 love, you가 입력된다 하지만, 입력 순서를 바꿔서

You love I

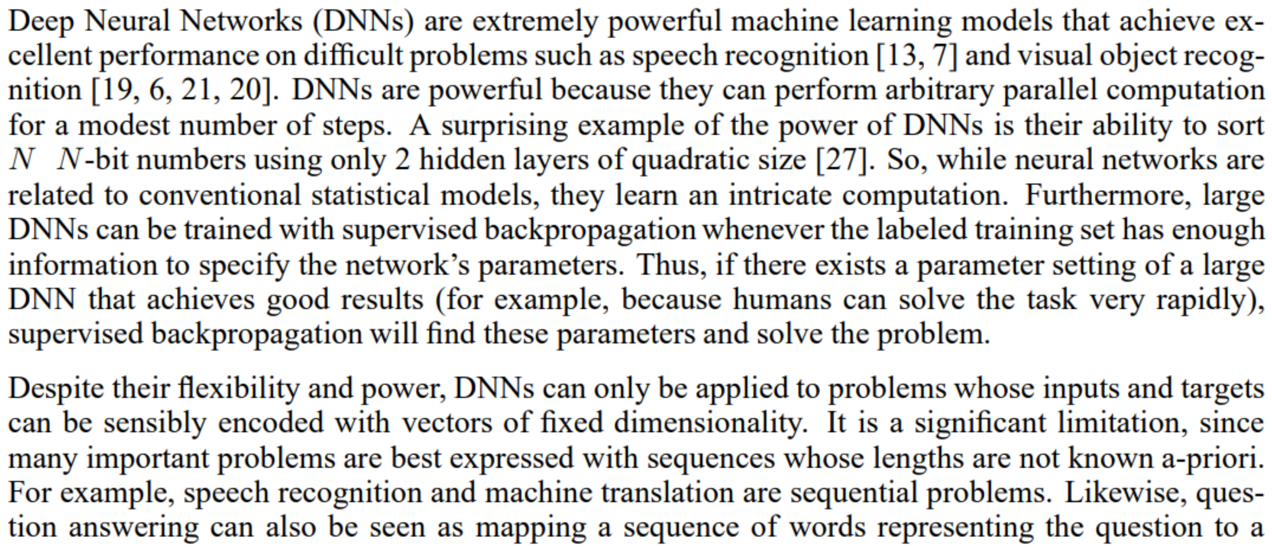
- You Love I 같이 입력하면, I와 You가 서로 관련되어 있음을 더 명확하게 학습할 수 있다

+ 입력 순서를 바꾸는 것으로 데이터셋의 연관성이 높아진다 → 학습에 필요한 데이터양이 줄어든다 → 난이도가 낮아진다

== 모델의 학습 속도, 성능을 동시에 개선할 수 있다

Introduction

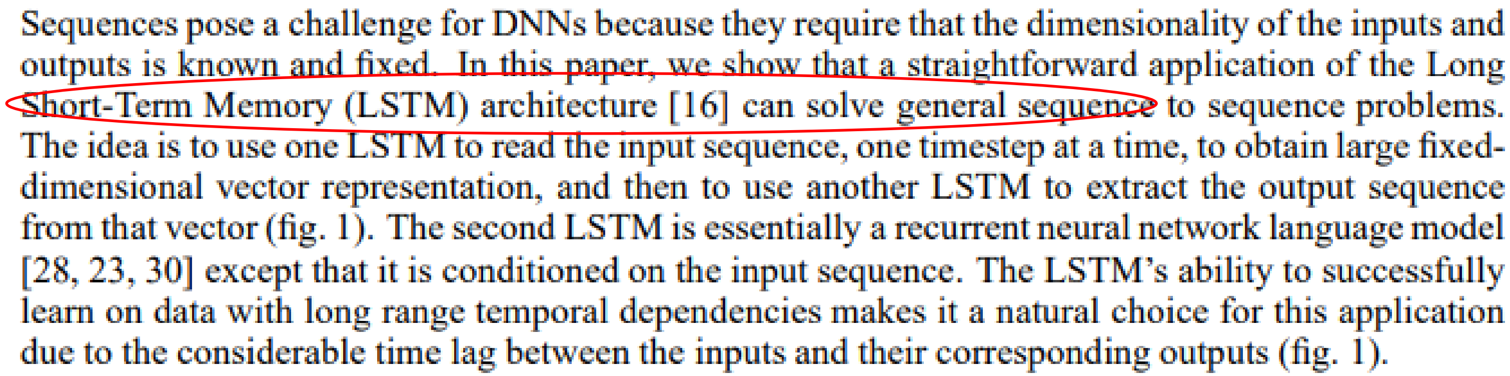
- 딥러닝은 다양한 Task에서 뛰어난 성능을 보여주고 있다
- NN은 전통적인 전통적인 통계적 모델과 많이 비슷하지만, 더 복잡하는 함수를 학습하는 과정에 있어 더 효과적이다
- 딥러닝은 역전파를 통해 학습을 진행하는데 복잡한 함수도 잘 찾아서 학습을 진행해준다
- 딥러닝은 강하지만, 일반적인 task에 대해서는 입력/출력의 차원이 고정되어있는 경우에는 한계가 존재한다
    - 입력과 타겟이 고정된 차원의 벡터로 인코딩된 문제에만 적용할 수 있다는 한계
    - Sequence와 Sequence를 매핑하는데 사용할 수 없다



Deep Neural Networks (DNNs) are extremely powerful machine learning models that achieve excellent performance on difficult problems such as speech recognition [13, 7] and visual object recognition [19, 6, 21, 20]. DNNs are powerful because they can perform arbitrary parallel computation for a modest number of steps. A surprising example of the power of DNNs is their ability to sort N N-bit numbers using only 2 hidden layers of quadratic size [27]. So, while neural networks are related to conventional statistical models, they learn an intricate computation. Furthermore, large DNNs can be trained with supervised backpropagation whenever the labeled training set has enough information to specify the network's parameters. Thus, if there exists a parameter setting of a large DNN that achieves good results (for example, because humans can solve the task very rapidly), supervised backpropagation will find these parameters and solve the problem.

Despite their flexibility and power, DNNs can only be applied to problems whose inputs and targets can be sensibly encoded with vectors of fixed dimensionality. It is a significant limitation, since many important problems are best expressed with sequences whose lengths are not known a-priori. For example, speech recognition and machine translation are sequential problems. Likewise, question answering can also be seen as mapping a sequence of words representing the question to a

Introduction



Sequences pose a challenge for DNNs because they require that the dimensionality of the inputs and outputs is known and fixed. In this paper, we show that a straightforward application of the Long Short-Term Memory (LSTM) architecture [16] can solve general sequence to sequence problems. The idea is to use one LSTM to read the input sequence, one timestep at a time, to obtain large fixed-dimensional vector representation, and then to use another LSTM to extract the output sequence from that vector (fig. 1). The second LSTM is essentially a recurrent neural network language model [28, 23, 30] except that it is conditioned on the input sequence. The LSTM's ability to successfully learn on data with long range temporal dependencies makes it a natural choice for this application due to the considerable time lag between the inputs and their corresponding outputs (fig. 1).

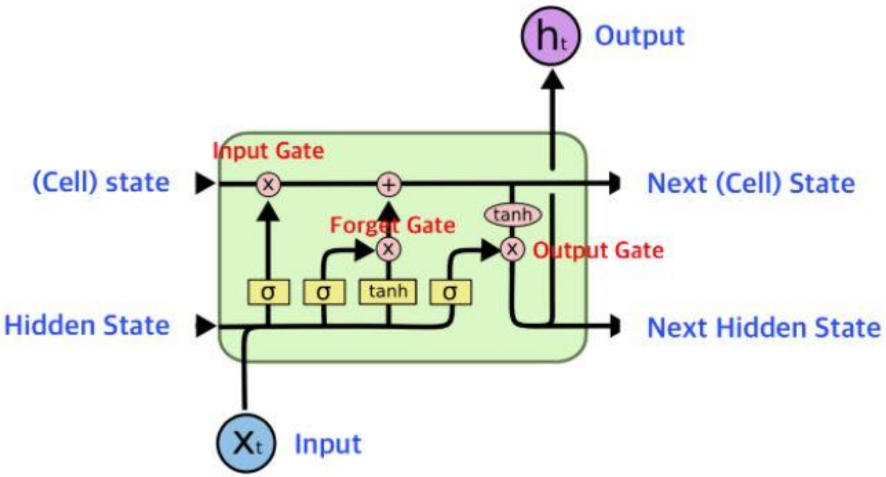
Sequence : 단어, 문장처럼 연속적으로 나열된 데이터

이전까지의 기계 번역 방법 과정

1. 문장을 작은 조각으로 나눈다
2. 각각의 조각을 번역한다
3. 다시 합친다

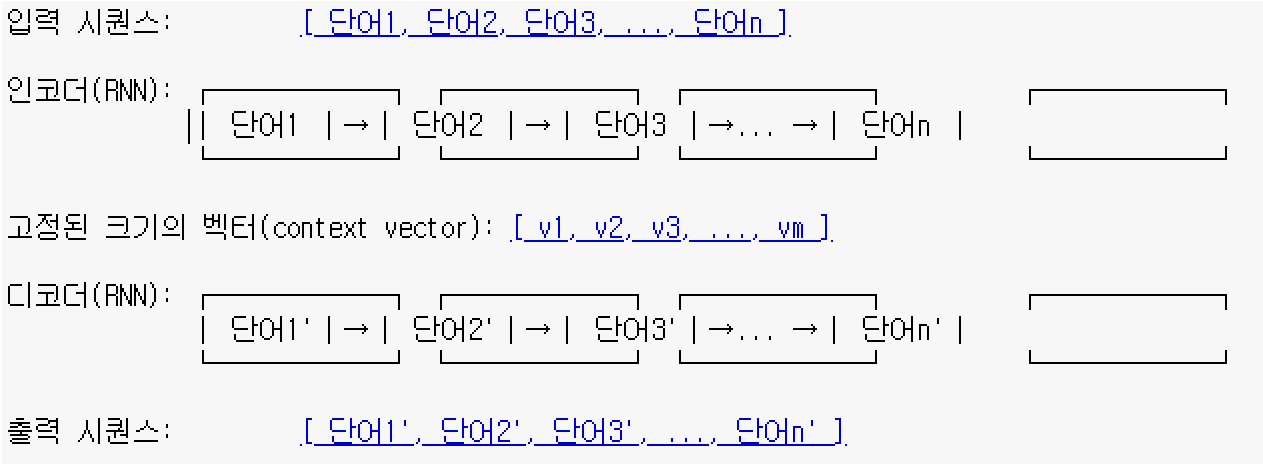
== 의미 전달이 제대로 되지가 않는다

Idea(Only Using LSTM)



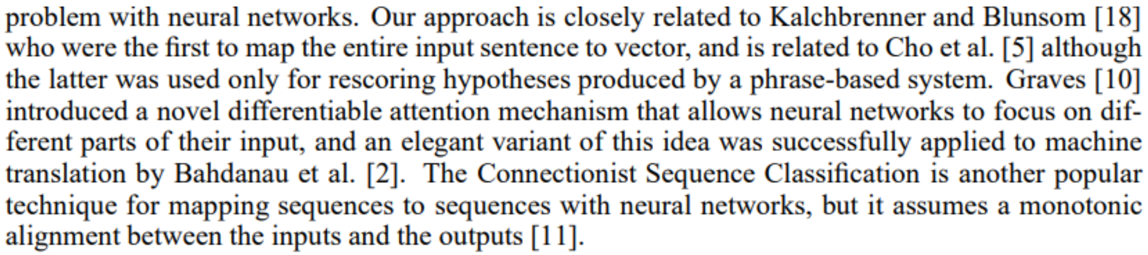
LSTM

- RNN의 Memory Cell에 Gate를 추가하여 불필요한 정보는 삭제하고 중요한 정보만 보존한다

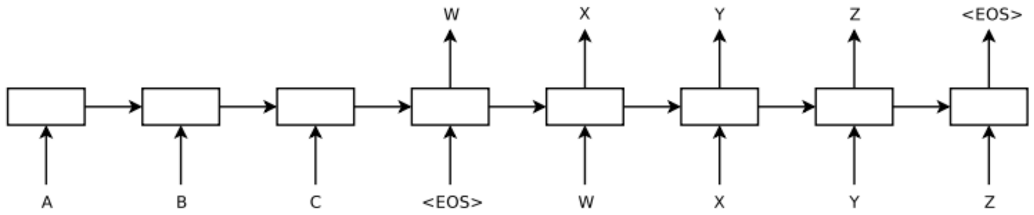


```
입력 시퀀스:         [ 단어1, 단어2, 단어3, ..., 단어n ]

인코더(RNN):
            || 단어1  |→| 단어2  |→| 단어3  |→...  →| 단어n  |

고정된 크기의 벡터(context vector): [ v1, v2, v3, ..., vm ]

디코더(RNN):
             | 단어1' |→| 단어2' |→| 단어3' |→...  →| 단어n' |

출력 시퀀스:         [ 단어1', 단어2', 단어3', ..., 단어n' ]
```

Approach

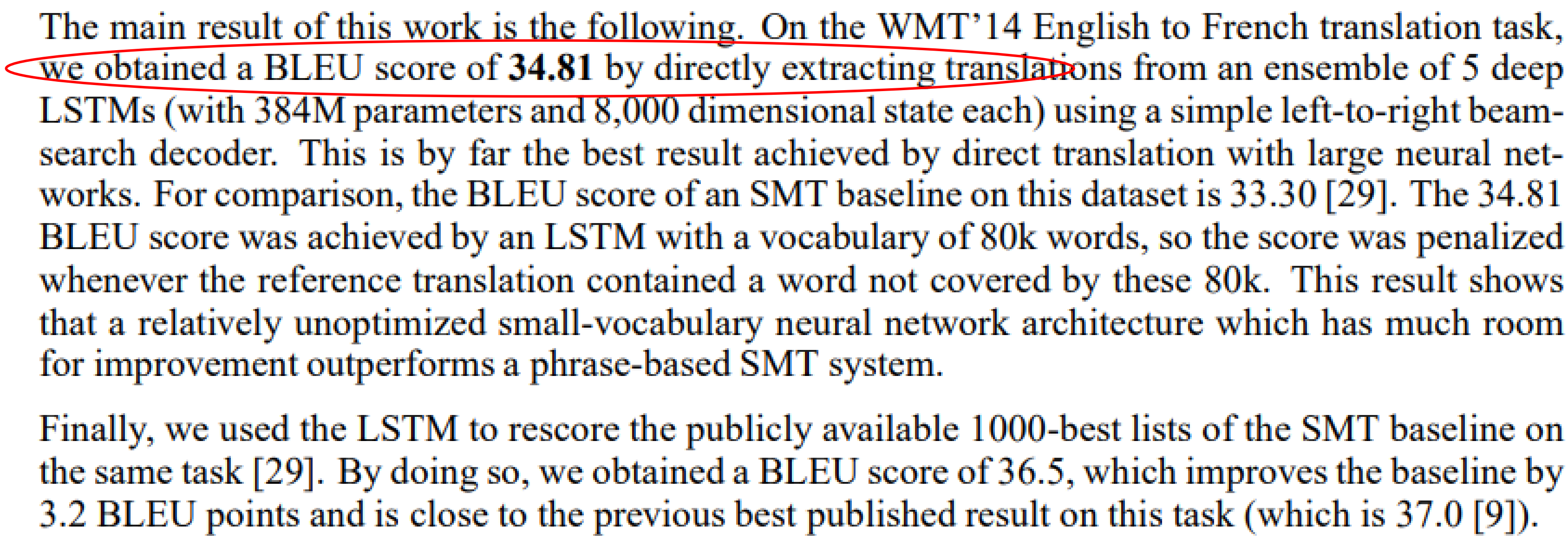


problem with neural networks. Our approach is closely related to Kalchbrenner and Blunsom [18] who were the first to map the entire input sentence to vector, and is related to Cho et al. [5] although the latter was used only for rescoring hypotheses produced by a phrase-based system. Graves [10] introduced a novel differentiable attention mechanism that allows neural networks to focus on different parts of their input, and an elegant variant of this idea was successfully applied to machine translation by Bahdanau et al. [2]. The Connectionist Sequence Classification is another popular technique for mapping sequences to sequences with neural networks, but it assumes a monotonic alignment between the inputs and the outputs [11].



1. 인코더 LSTM은 입력 시퀀스 a,b,c를 순차적으로 처리하고 각 단계에서 hidden state를 업데이트한다
2. 입력 시퀀스 끝(Eos)에 도달한 후, 인코더의 마지막 hidden state가 context vector로 사용된다
   해당 벡터는 입력 시퀀스의 정보를 압축한 것으로 디코더에 전달, 디코더의 입력값으로 활용된다
3. 디코더 LSTM은 입력값으로 context vecto와 이전 생성된 단어를 함께 받는다
   단 첫 단계에서는 SOS(시작토큰)을 이전에 생성된 단어로 사용한다
4. 디코더는 각 단계에서 hidden state를 업데이트하며 새로운 단어를 생성한다. 디코더의 입력으로는 context vector와 이전에 생성된 단어가 함께 사용된다
5. 출력시퀀스를 생성할 때까지 해당 과정을 반복한다

Score



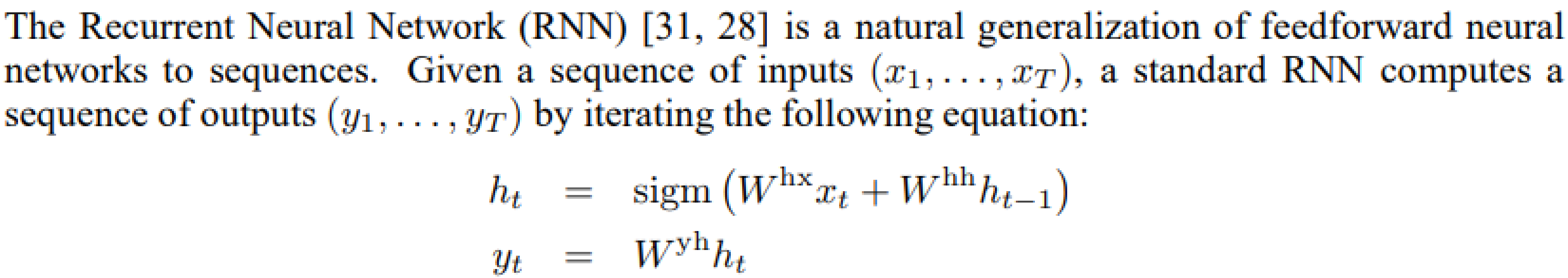
The main result of this work is the following. On the WMT'14 English to French translation task, we obtained a BLEU score of **34.81** by directly extracting translations from an ensemble of 5 deep LSTMs (with 384M parameters and 8,000 dimensional state each) using a simple left-to-right beam-search decoder. This is by far the best result achieved by direct translation with large neural networks. For comparison, the BLEU score of an SMT baseline on this dataset is 33.30 [29]. The 34.81 BLEU score was achieved by an LSTM with a vocabulary of 80k words, so the score was penalized whenever the reference translation contained a word not covered by these 80k. This result shows that a relatively unoptimized small-vocabulary neural network architecture which has much room for improvement outperforms a phrase-based SMT system.

Finally, we used the LSTM to rescore the publicly available 1000-best lists of the SMT baseline on the same task [29]. By doing so, we obtained a BLEU score of 36.5, which improves the baseline by 3.2 BLEU points and is close to the previous best published result on this task (which is 37.0 [9]).

- WMT 2014 데이터셋에 대해서 state or the art을 보였다

- 5개의 LSTM 모델을 조합한 것이 논문에서 언급하는 앙상블이다

- 384백만개의 파라미터와 8000개의 차원을 상태를 갖는다



The Recurrent Neural Network (RNN) [31, 28] is a natural generalization of feedforward neural networks to sequences. Given a sequence of inputs $(x_1, \ldots, x_T)$, a standard RNN computes a sequence of outputs $(y_1, \ldots, y_T)$ by iterating the following equation:

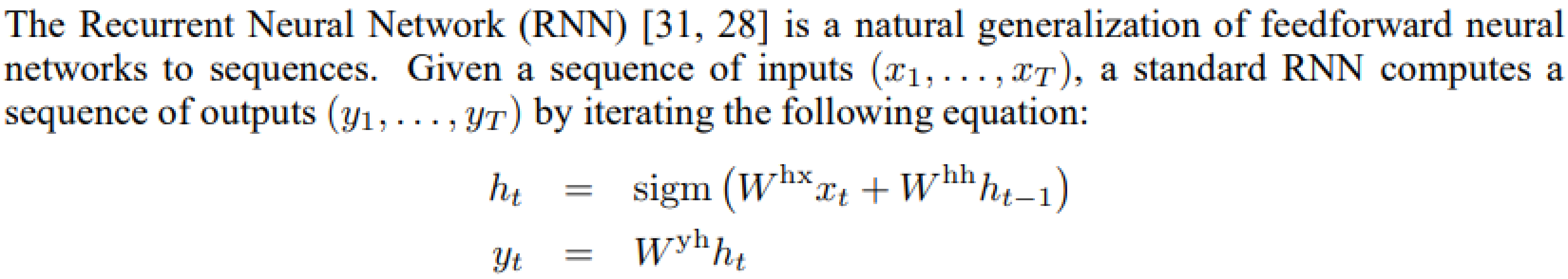
$$\begin{aligned} h_t &= \mathrm{sigm}\left(W^{\mathrm{hx}} x_t + W^{\mathrm{hh}} h_{t-1}\right) \\ y_t &= W^{\mathrm{yh}} h_t \end{aligned}$$

Input Sequence(x) : 총 t개

각각의 small x는 전부 하나의 단어를 의미

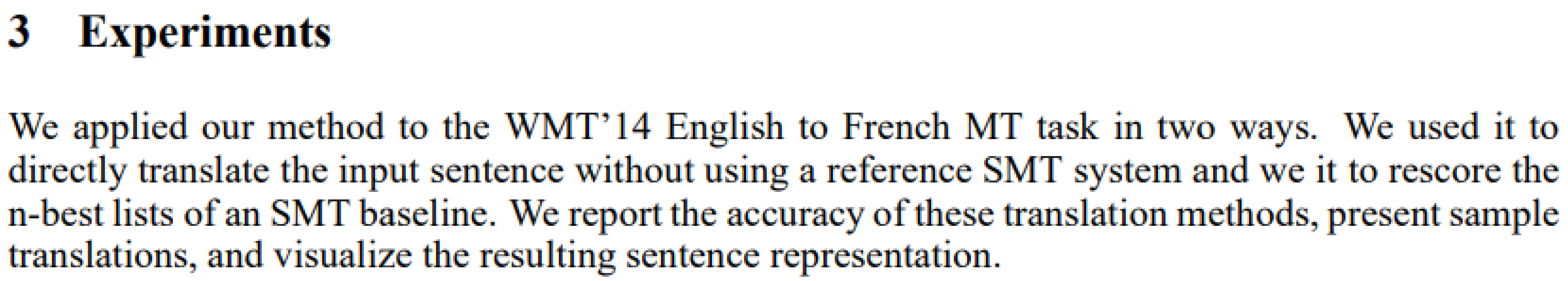
Map the input Sequence to a fixed-sized vector using one RNN

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



The Recurrent Neural Network (RNN) [31, 28] is a natural generalization of feedforward neural networks to sequences. Given a sequence of inputs $(x_1, \ldots, x_T)$, a standard RNN computes a sequence of outputs $(y_1, \ldots, y_T)$ by iterating the following equation:

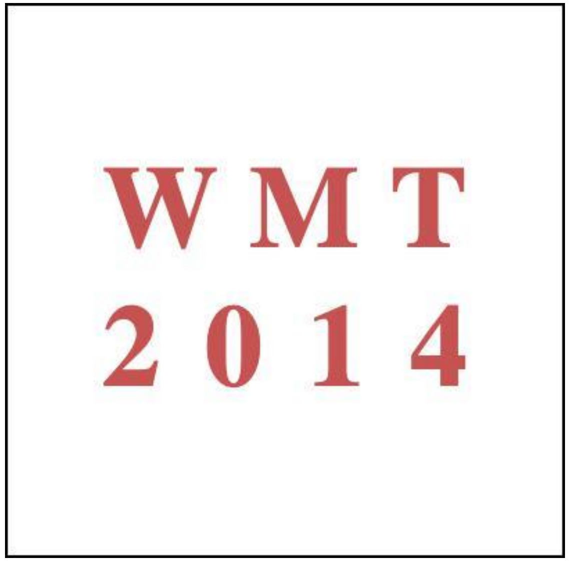
$$\begin{aligned} h_t &= \mathrm{sigm}\left(W^{\mathrm{hx}} x_t + W^{\mathrm{hh}} h_{t-1}\right) \\ y_t &= W^{\mathrm{yh}} h_t \end{aligned}$$

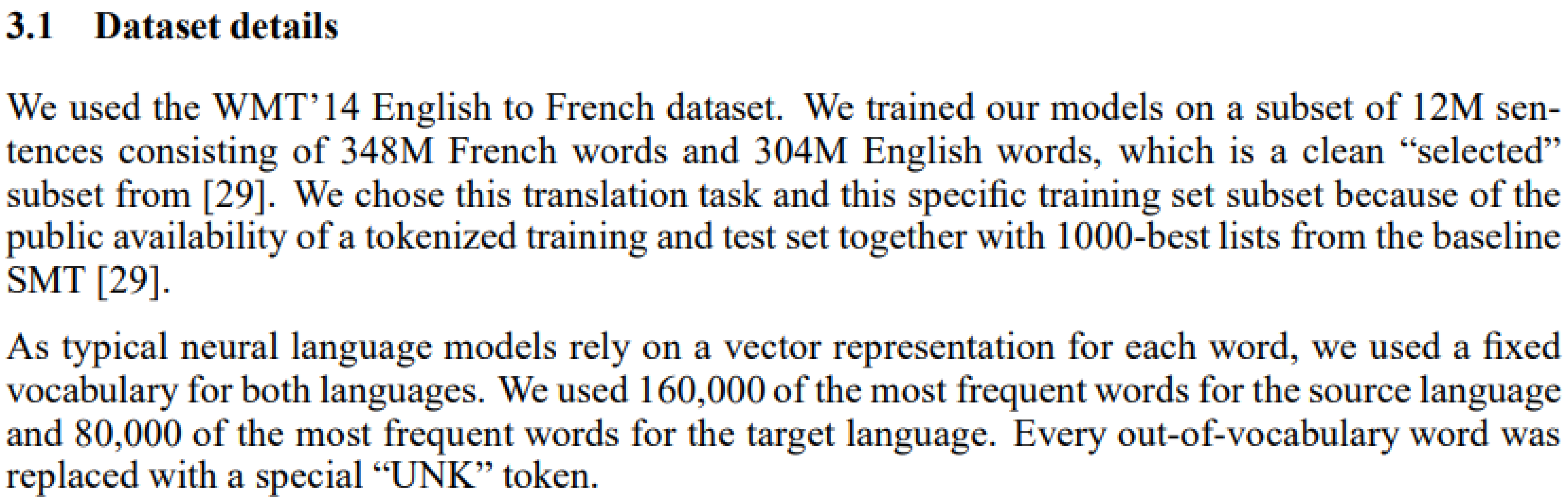


3 Experiments

We applied our method to the WMT'14 English to French MT task in two ways. We used it to directly translate the input sentence without using a reference SMT system and we it to rescore the n-best lists of an SMT baseline. We report the accuracy of these translation methods, present sample translations, and visualize the resulting sentence representation.

사용한 데이터셋

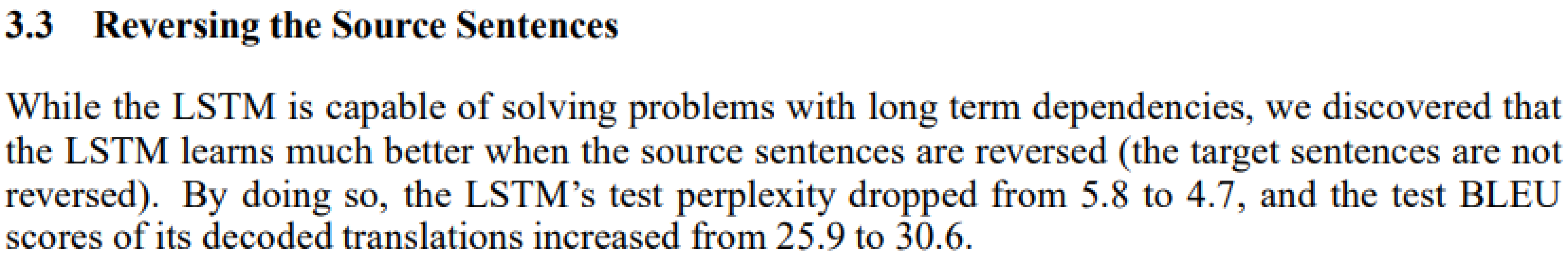




3.1 Dataset details

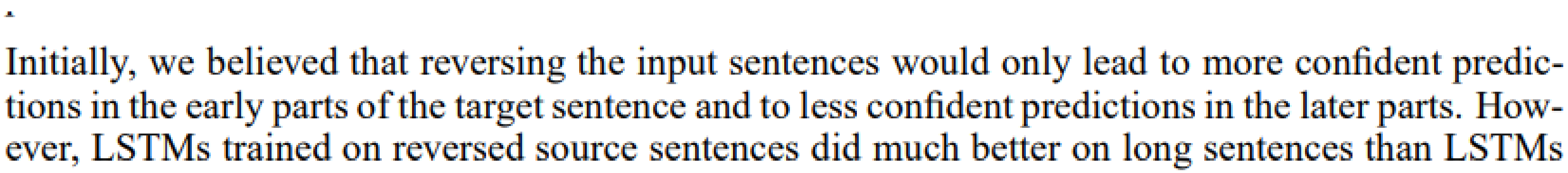
We used the WMT'14 English to French dataset. We trained our models on a subset of 12M sentences consisting of 348M French words and 304M English words, which is a clean "selected" subset from [29]. We chose this translation task and this specific training set subset because of the public availability of a tokenized training and test set together with 1000-best lists from the baseline SMT [29].

As typical neural language models rely on a vector representation for each word, we used a fixed vocabulary for both languages. We used 160,000 of the most frequent words for the source language and 80,000 of the most frequent words for the target language. Every out-of-vocabulary word was replaced with a special "UNK" token.



3.3 Reversing the Source Sentences

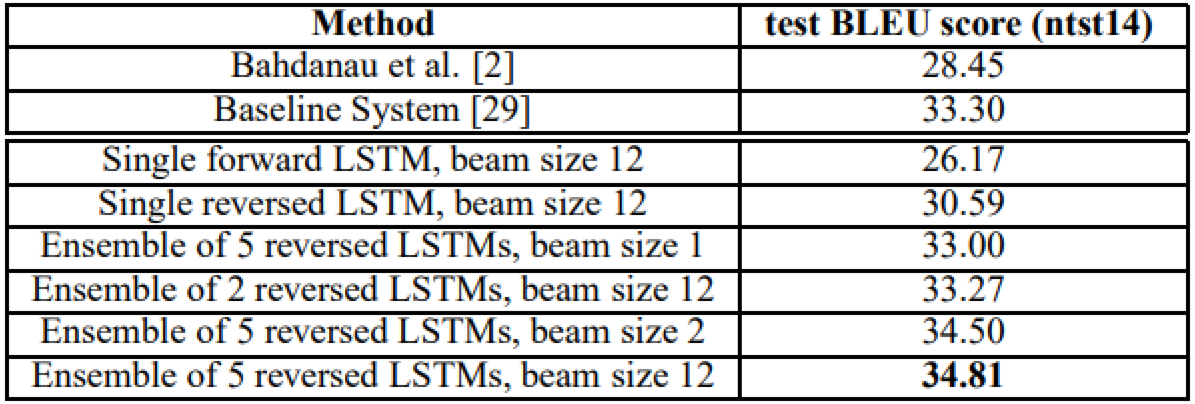
While the LSTM is capable of solving problems with long term dependencies, we discovered that the LSTM learns much better when the source sentences are reversed (the target sentences are not reversed). By doing so, the LSTM's test perplexity dropped from 5.8 to 4.7, and the test BLEU scores of its decoded translations increased from 25.9 to 30.6.



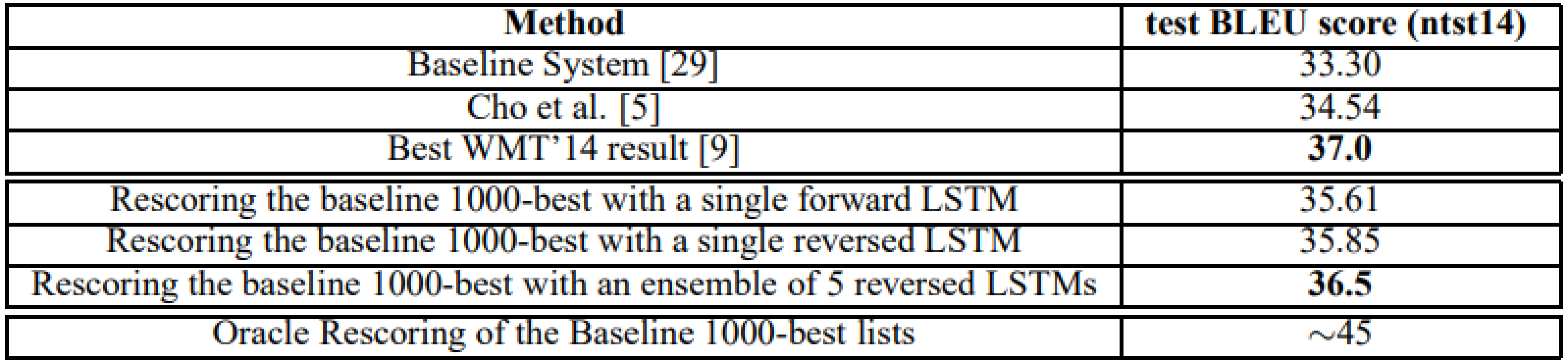
.

Initially, we believed that reversing the input sentences would only lead to more confident predictions in the early parts of the target sentence and to less confident predictions in the later parts. However, LSTMs trained on reversed source sentences did much better on long sentences than LSTMs

결과



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**



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	∼45