NLP paper study week-03 NLP - Sequence to Sequence Learning with Neural Networks 📈

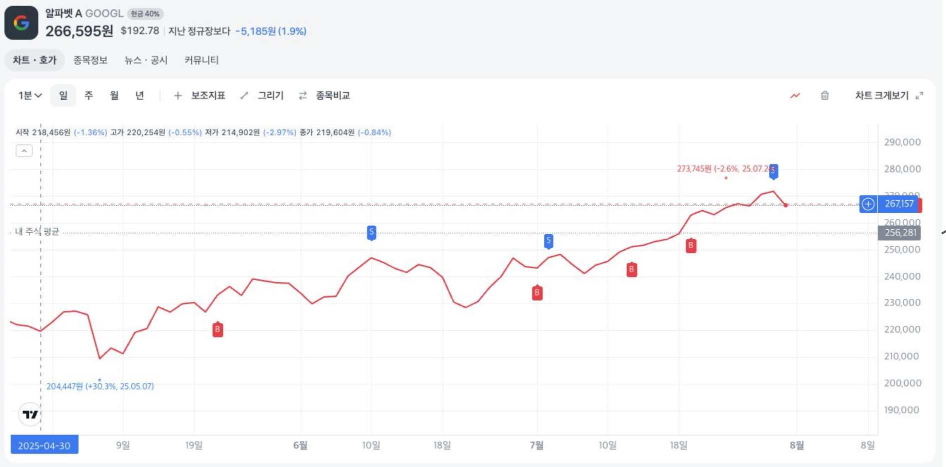
Encoder & Decoder

Cheonghae Kim

Sequence data

시퀀스 데이터란 순서대로 정렬된 데이터의 연속이다.

ex) 문자열이나 시간에 따른 이벤트 로그, 금융에서는 시간에 따른 주식 가격 변동 같은 형태



알파벳 A GOOGL 현금 40%

266,595원 $192.78 | 지난 정규장보다 -5,185원 (1.9%)

시작 218,456원 (-1.36%) 고가 220,254원 (-0.55%) 저가 214,902원 (-2.97%) 종가 219,604원 (-0.84%)

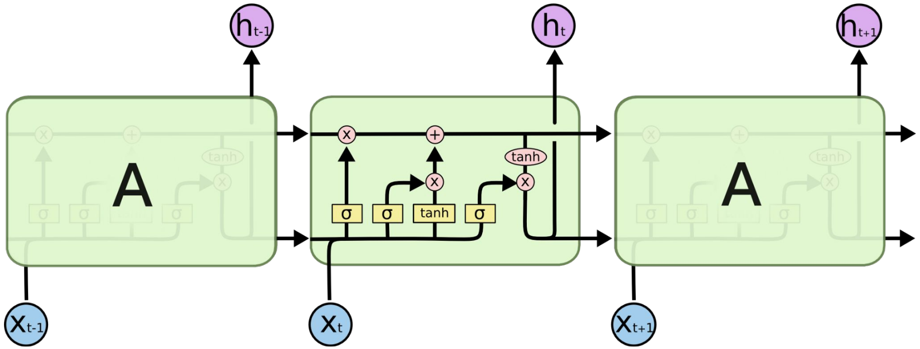
DNN은 시계열에 최적화 X

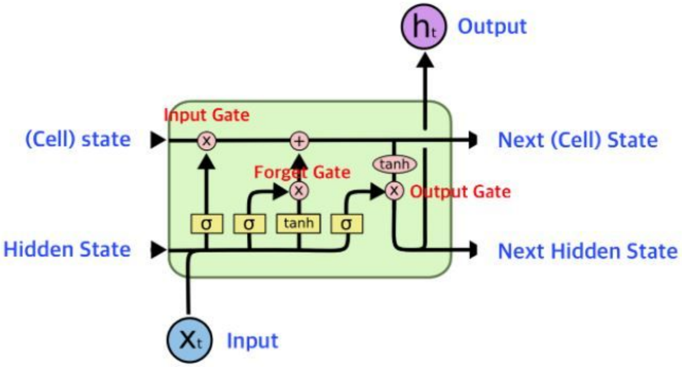
RNN이 적합

LSTM (Long Short-Term Memory)

일반 RNN의 긴 시퀀스 처리의 문제를 보완한 RNN계열 뉴럴넷

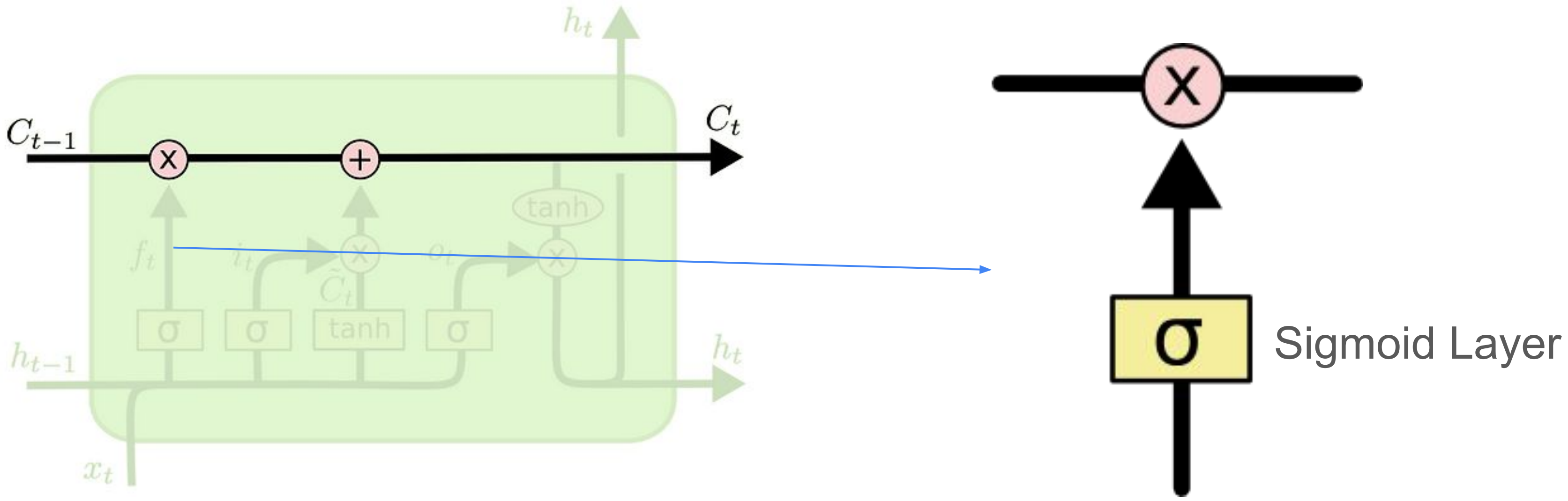
LSTM은 중요한 정보는 오래 기억하고 불필요한 정보는 잊어버리도록 설계됨





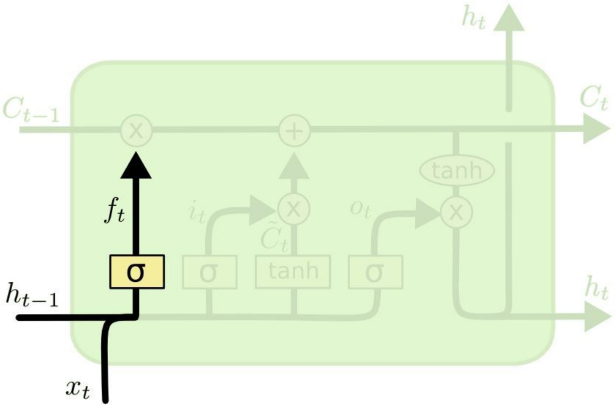
Cell State

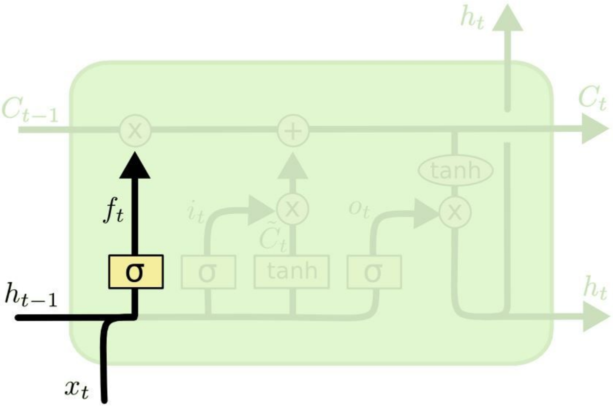
장기기억을 담당하는 부분이 Cell State이며 시간에 따라 정보를 조절한다.



Forget Gate

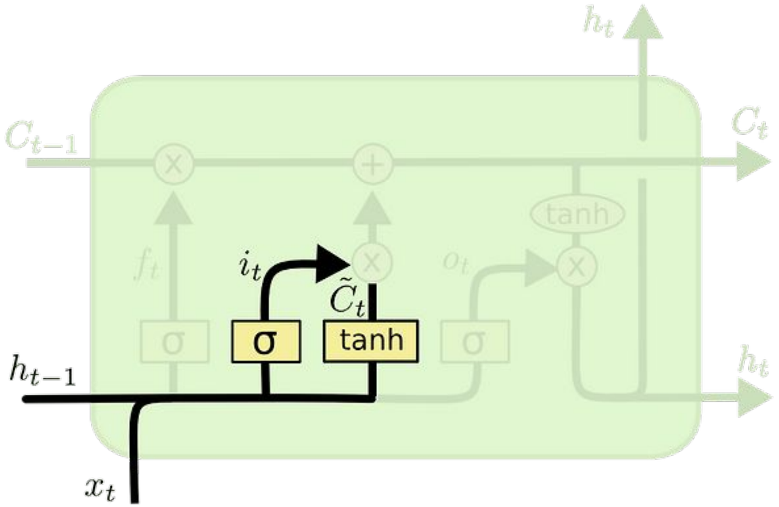
과거의 정보를 버릴지 말지 결정하는 부분





- h_{t-1}: 이전 시점의 은닉 상태
- x_t: 현재 시점 입력

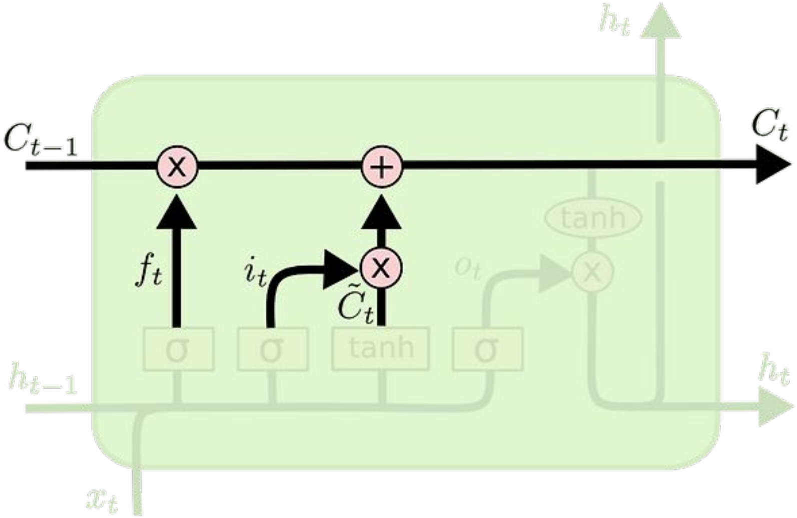
LSTM (Long Short-Term Memory)



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$

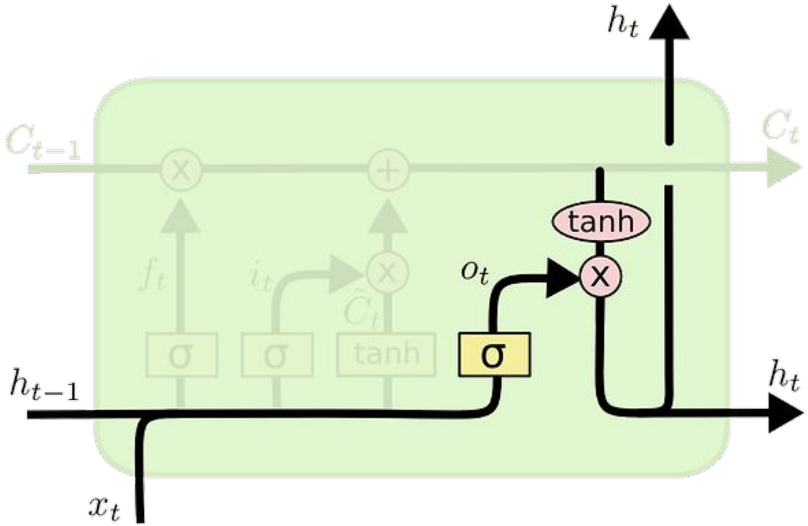
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

Update



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Output Gate



o_t: 출력 게이트 벡터 (0~1)

σ: 시그모이드 함수 → 0~1로 스케일링

W_o: 출력 게이트 가중치 행렬

b_o: 출력 게이트 편향

$[h_{t-1}, x_t]$: 이전 은닉 상태와 현재 입력을 **concat**한 벡터

$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

$\tanh(C_t)$: 장기 메모리를 -1~1 사이로 압축

$o_t \odot \tanh(C_t)$: 출력 게이트가 허용한 만큼만 은닉 상태에 반영

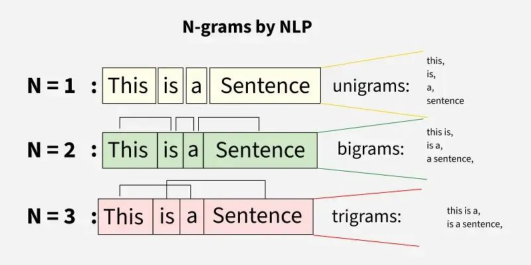
최종 h_t는 **단기 기억이자 RNN의 출력**

BLEU (Bilingual Evaluation Understudy) - Precision

1. n-그램 정밀도

   n-gram(연속된 n개 단어)에 정답 문장이

   얼마릏나 포함되는지 계산



2. 클리핑 (Clipping)

   모델이 같은 단어를 반복하여 정답처리를 하면

   정밀도가 부풀려짐에 따라 일정 횟수 이상부터 인정 X

$$r(\theta) = \frac{\pi_\theta(a|s)}{\pi_{\theta_{\text{old}}}(a|s)}$$

$$L^{\text{CLIP}} = \min\left(r(\theta)A, \text{clip}(r(\theta), 1-\epsilon, 1+\epsilon)A\right)$$

3. 길이 패널티 (Brevity Penalty, BP)

   너무 짧은 문장에 대한 패널티

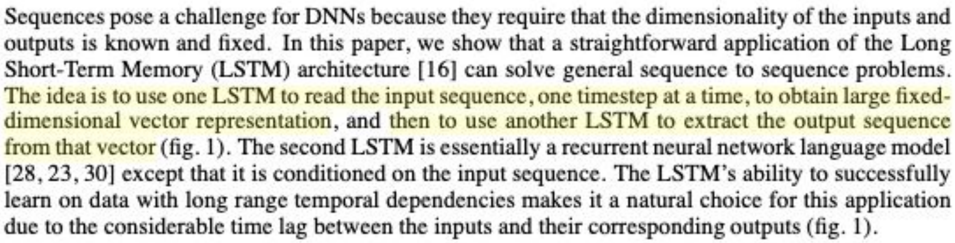
$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}$$

4. 최종 BLEU 스코어

   n-그램 정밀도의 기하 평균 × 길이 패널티

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

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).

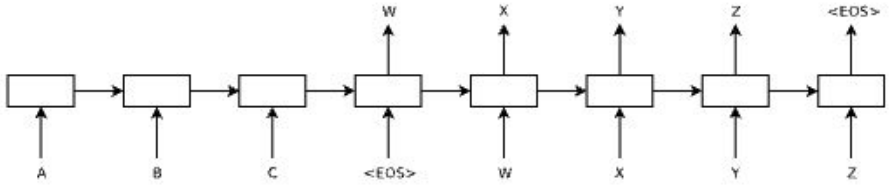
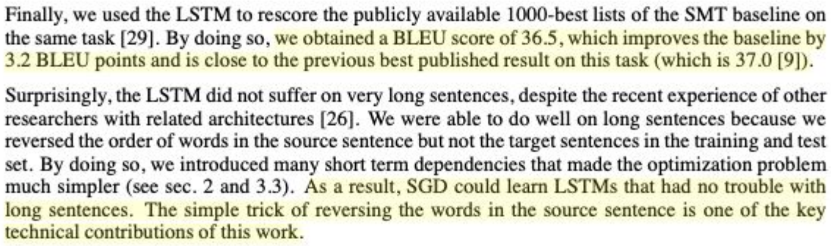


Figure 1: Our model reads an input sentence "ABC" and produces "WXYZ" as the output sentence. The model stops making predictions after outputting the end-of-sentence token. Note that the LSTM reads the input sentence in reverse, because doing so introduces many short term dependencies in the data that make the optimization problem much easier.



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]).

Surprisingly, the LSTM did not suffer on very long sentences, despite the recent experience of other researchers with related architectures [26]. We were able to do well on long sentences because we reversed the order of words in the source sentence but not the target sentences in the training and test set. By doing so, we introduced many short term dependencies that made the optimization problem much simpler (see sec. 2 and 3.3). As a result, SGD could learn LSTMs that had no trouble with long sentences. The simple trick of reversing the words in the source sentence is one of the key technical contributions of this work.

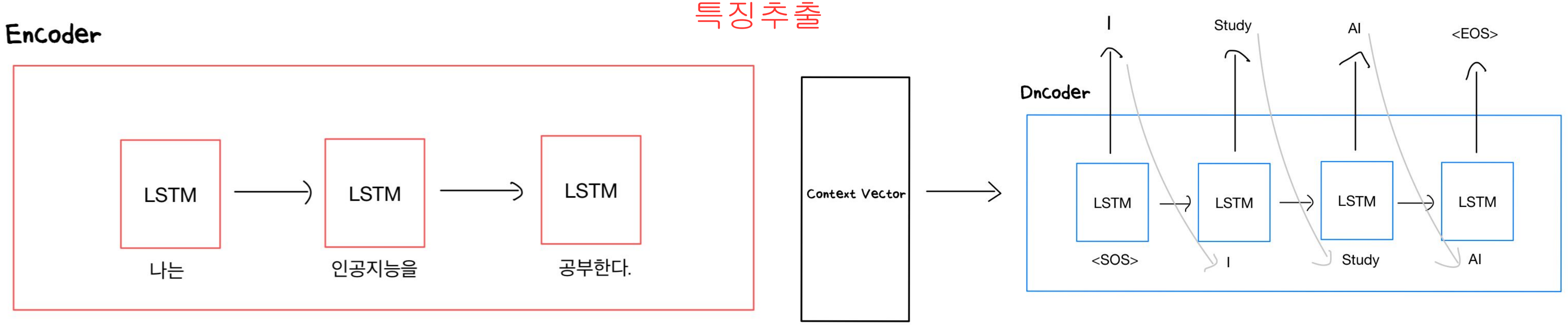
```
원래 입력:  I am going to the store
뒤집은 입력: store the to going am I
```

- 인코더 마지막 hidden state는 `"I"` 를 본 직후 상태
- 디코더 입장에서는 **출력 첫 단어와 연관된 입력 초반 단어 정보가** 인코더 hidden state에 더 강하게 남아 있음

The model

논문에서는 입력을
뒤집어서 넣음 !

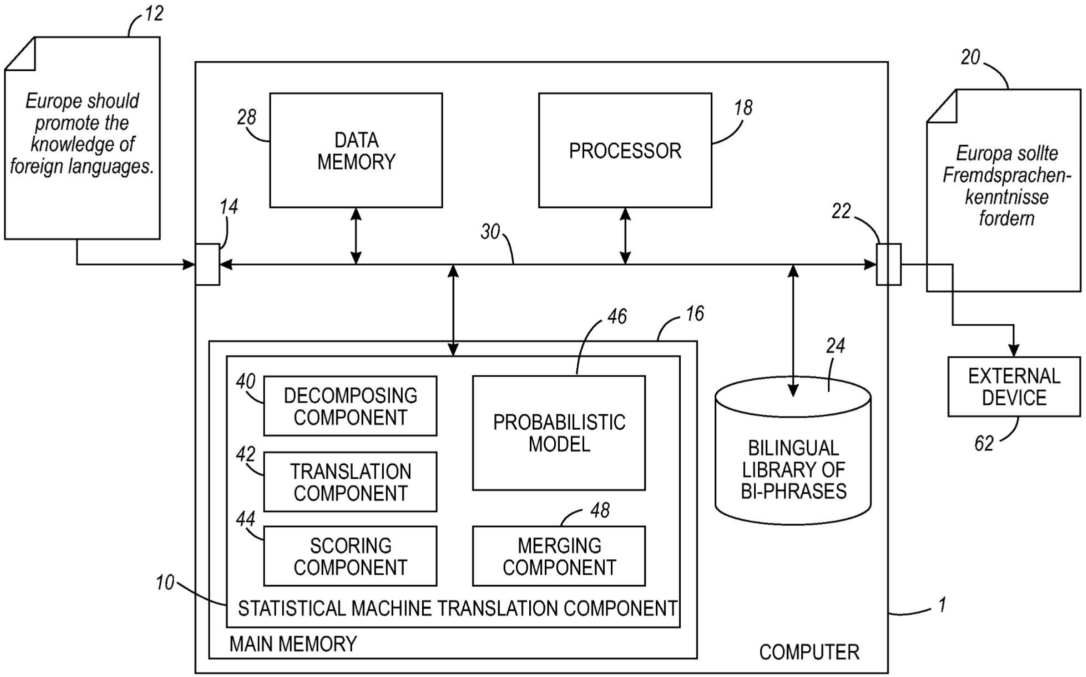
Autoregressive방식으로
예측을 수행



특징추출

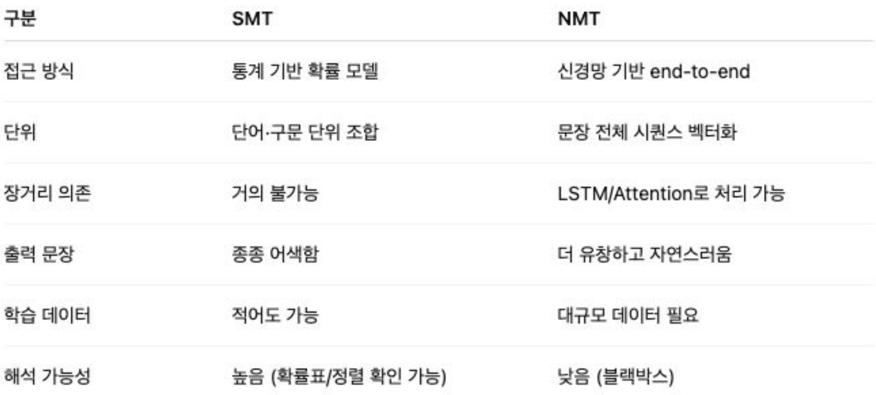
$$p(y_1,\ldots,y_{T'}|x_1,\ldots,x_T)=\prod_{t=1}^{T'}p(y_t|v,y_1,\ldots,y_{t-1})$$

SMT (Statistical Machine Translation)

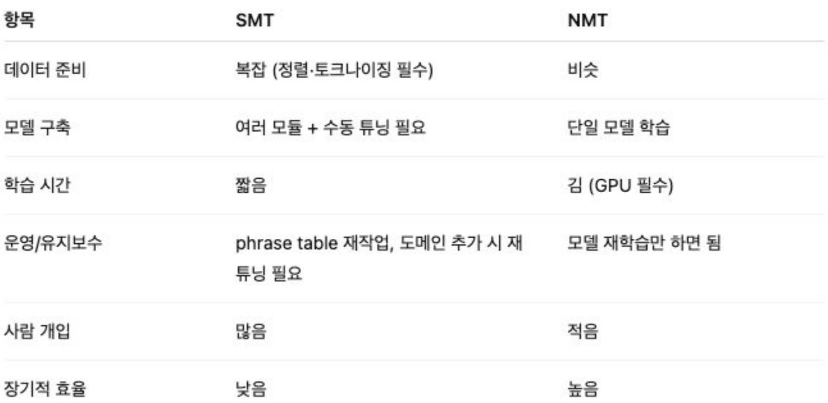


출처: https://eureka.patsnap.com/patent-US20110178791A1

SMT(Statistical) vs NMT (Neural)



구분	SMT	NMT
접근 방식	통계 기반 확률 모델	신경망 기반 end-to-end
단위	단어·구문 단위 조합	문장 전체 시퀀스 벡터화
장거리 의존	거의 불가능	LSTM/Attention로 처리 가능
출력 문장	종종 어색함	더 유창하고 자연스러움
학습 데이터	적어도 가능	대규모 데이터 필요
해석 가능성	높음 (확률표/정렬 확인 가능)	낮음 (블랙박스)



항목	SMT	NMT
데이터 준비	복잡 (정렬·토크나이징 필수)	비슷
모델 구축	여러 모듈 + 수동 튜닝 필요	단일 모델 학습
학습 시간	짧음	김 (GPU 필수)
운영/유지보수	phrase table 재작업, 도메인 추가 시 재 튜닝 필요	모델 재학습만 하면 됨
사람 개입	많음	적음
장기적 효율	낮음	높음

Experiments

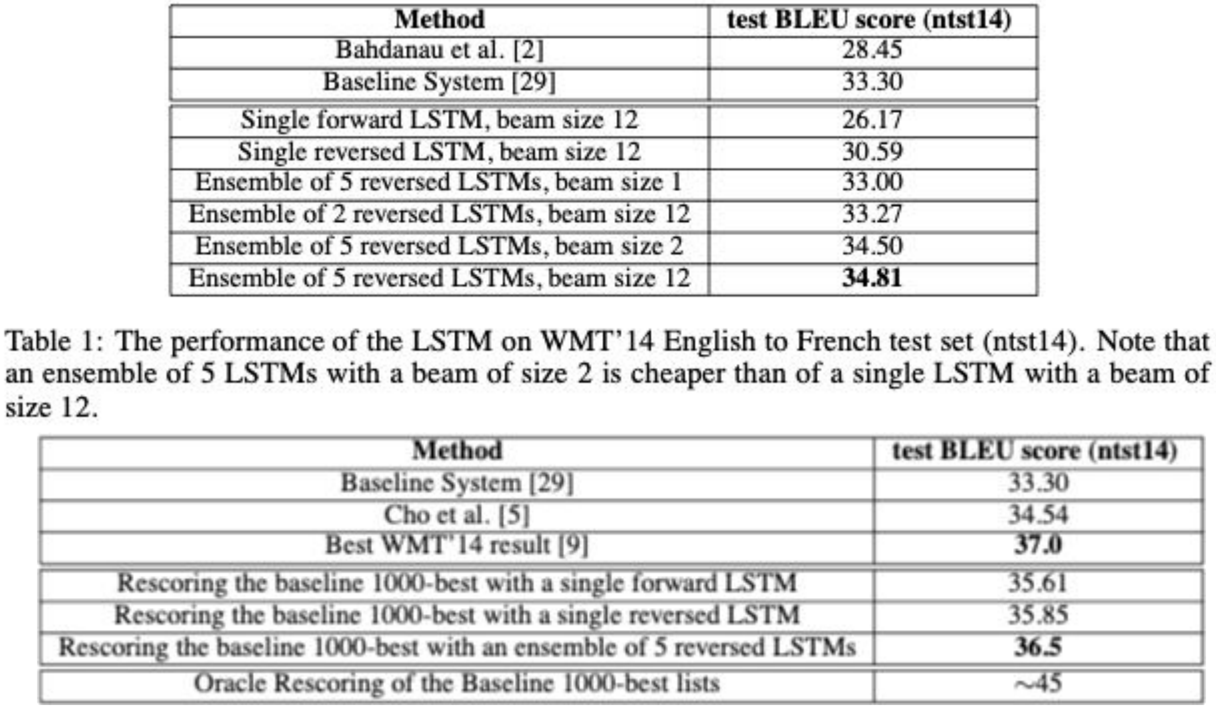
Data: WMT’14 English to French

Data size: subset of 12M sen-tences consisting of 348M French words and 304M English words,

Model architeture: 4Layer LSTM (for long sentence), Word Embedding d =1000, Param 384M, batch size 128, epoch 7.5

reverse input -> perplexity 1.1 감소 + BLEU 4.7 향상

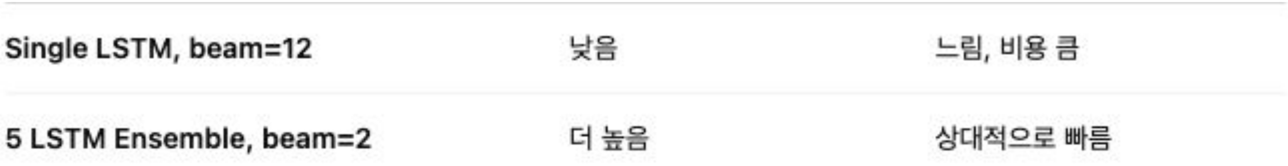
Experiments Result (Metric)



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**

Table 1: The performance of the LSTM on WMT'14 English to French test set (ntst14). Note that an ensemble of 5 LSTMs with a beam of size 2 is cheaper than of a single LSTM with a beam of size 12.

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



Single LSTM, beam=12	낮음	느림, 비용 큼
5 LSTM Ensemble, beam=2	더 높음	상대적으로 빠름

Experiments Result (Metric)

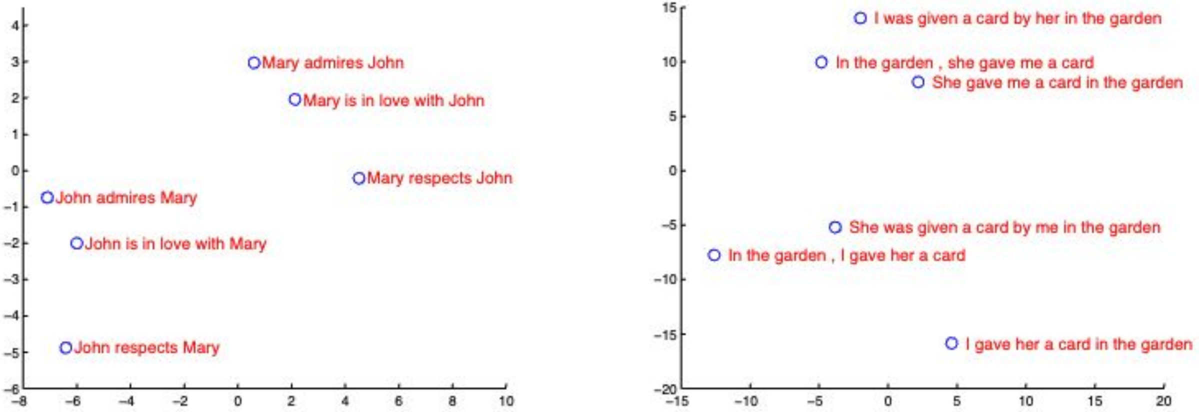
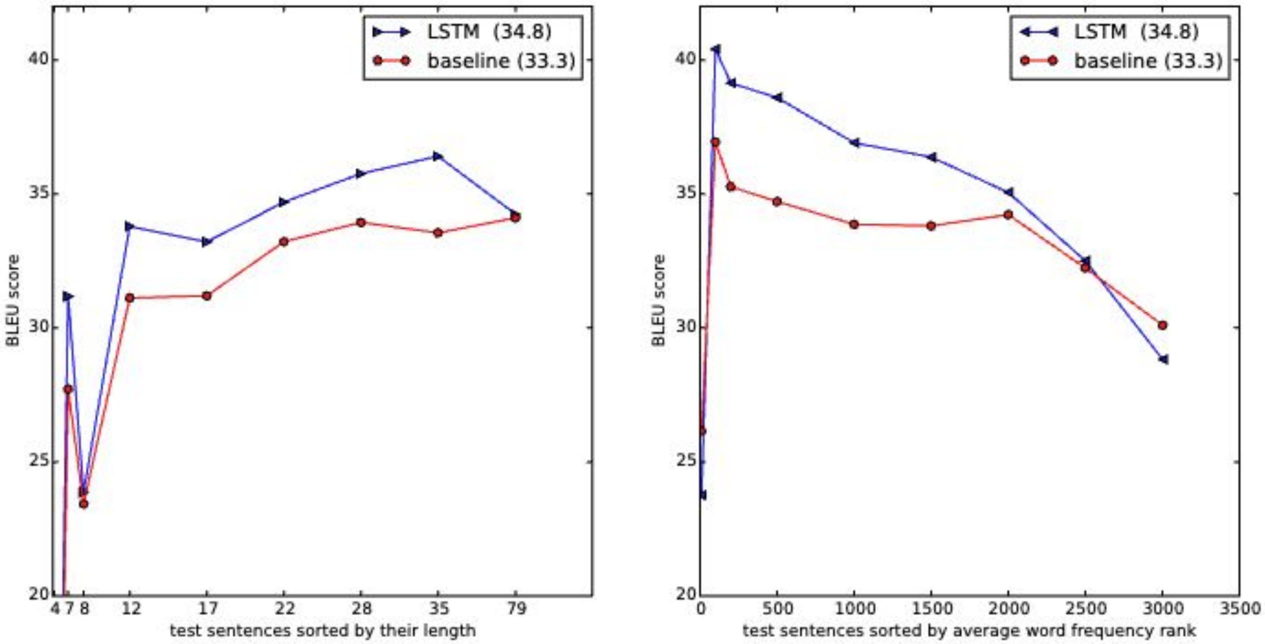
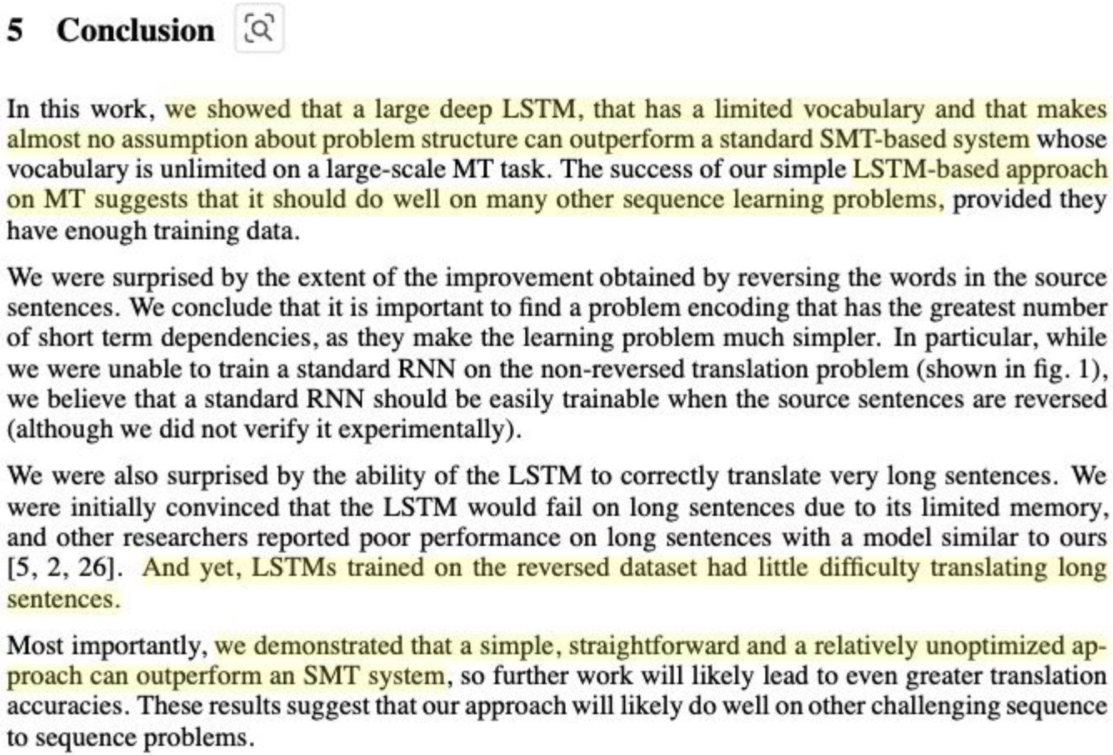


Figure 2: The figure shows a 2-dimensional PCA projection of the LSTM hidden states that are obtained after processing the phrases in the figures. The phrases are clustered by meaning, which in these examples is primarily a function of word order, which would be difficult to capture with a bag-of-words model. Notice that both clusters have similar internal structure.

Experiments Result (Metric)



Conclusion



5 Conclusion

In this work, we showed that a large deep LSTM, that has a limited vocabulary and that makes almost no assumption about problem structure can outperform a standard SMT-based system whose vocabulary is unlimited on a large-scale MT task. The success of our simple LSTM-based approach on MT suggests that it should do well on many other sequence learning problems, provided they have enough training data.

We were surprised by the extent of the improvement obtained by reversing the words in the source sentences. We conclude that it is important to find a problem encoding that has the greatest number of short term dependencies, as they make the learning problem much simpler. In particular, while we were unable to train a standard RNN on the non-reversed translation problem (shown in fig. 1), we believe that a standard RNN should be easily trainable when the source sentences are reversed (although we did not verify it experimentally).

We were also surprised by the ability of the LSTM to correctly translate very long sentences. We were initially convinced that the LSTM would fail on long sentences due to its limited memory, and other researchers reported poor performance on long sentences with a model similar to ours [5, 2, 26]. And yet, LSTMs trained on the reversed dataset had little difficulty translating long sentences.

Most importantly, we demonstrated that a simple, straightforward and a relatively unoptimized approach can outperform an SMT system, so further work will likely lead to even greater translation accuracies. These results suggest that our approach will likely do well on other challenging sequence to sequence problems.

Sequence to Sequence Learning with Neural Networks

Ilya Sutskever
Google
ilyasu@google.com

Oriol Vinyals
Google
vinyals@google.com

Quoc V. Le
Google
qvl@google.com

reference

위키독스 딥 러닝을 이용한 자연어처리 입문: https://wikidocs.net/book/2155

LSTM: https://dgkim5360.tistory.com/entry/understanding-long-short-term-memory-lstm-kr

seq2seq paper: https://arxiv.org/pdf/1409.3215