Sequence to Sequence Learning with Neural Networks

Author	Ilya Sutskever, Oriol Vinyals, Quoc V. Le
Field	NLP
Journal	NIPS
# Published Year	2014
Speaker	나보영 조태완
Summary	Seq2Seq
status	Finished!
link	https://arxiv.org/pdf/1409.3215.pdf

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

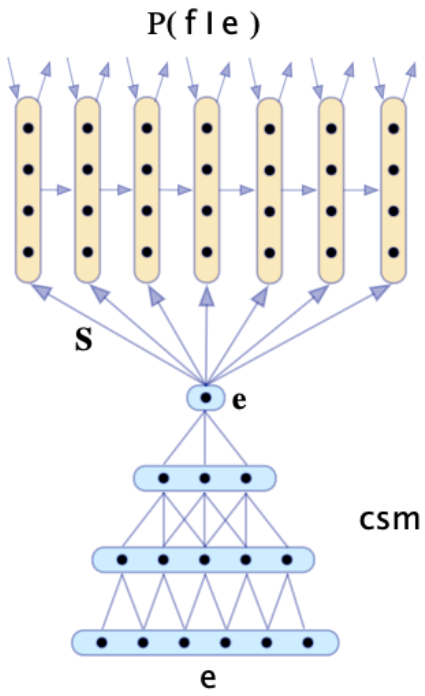
1. 논문이 풀고자 하는 문제

- Although DNNs work well whenever large labeled training sets are available, they cannot be used to map **sequences to sequences**.
- Deep Neural Network (DNNs)가 고정된 차원의 벡터로만 입력과 목표를 합리적으로 인코딩할 수 있는 문제에 제한되어 있다는 문제를 해결하려고 합니다. 특히, 이 제한은 시퀀스의 길이가 미리 알려져 있지 않은 많은 중요한 문제에 적용되기 어렵게 만듭니다.

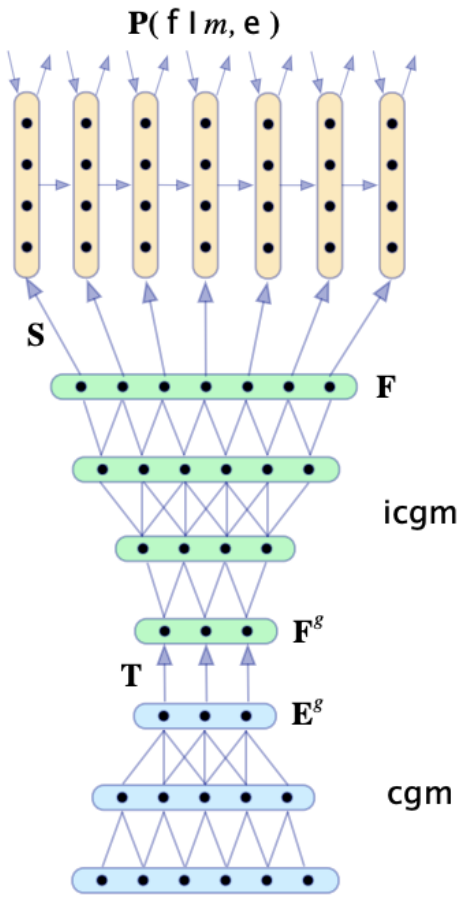
2. 기존에 문제를 풀었던 방법들

Kalchbrenner와 Blunsom의 접근법(RCTMs)

- Our approach is closely related to Kalchbrenner and Blunsom [18] who were the first to map the entire input sentence to vector
  → 이 접근법의 핵심은 Convolutional Neural Networks (CNN)를 사용하여 입력 문장을 처리하고, **이를 고정된 크기의 벡터 표현으로 압축**하는 것입니다. 이 벡터 표현은 후속 디코딩 과정에서 목표 문장을 생성하는 데 사용되었습니다.



RCTM I

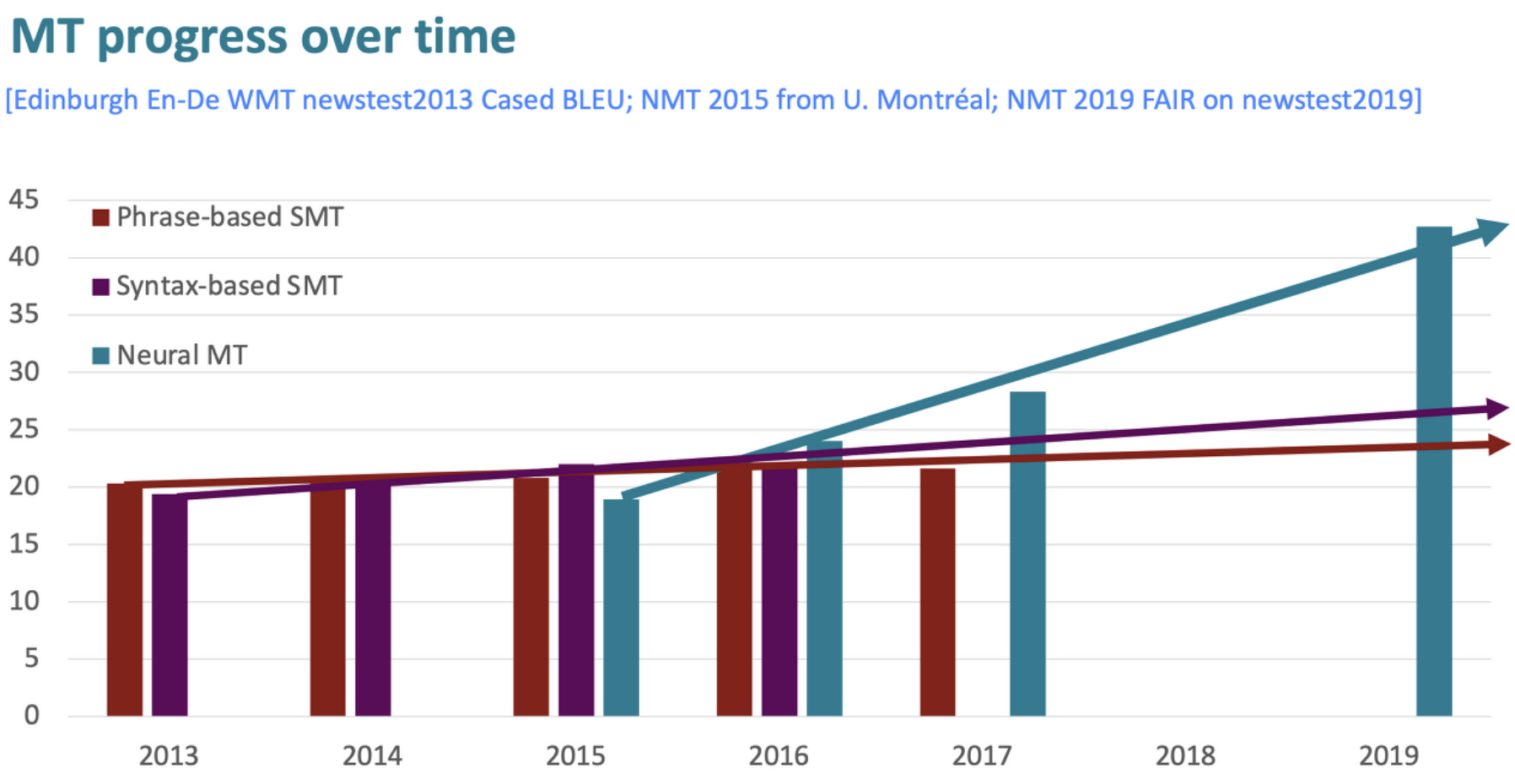


RCTM II

- 시퀀스-투-시퀀스 변환의 일반적인 문제를 해결하려는 동일한 목표를 가지고 있지만, 사용하는 신경망 구조와 메커니즘에 차이가 있습니다. Kalchbrenner와 Blunsom의 방법은 seq2seq 모델의 발전에 있어서 중요한 이정표로 볼 수 있으며, 고정된 크기의 벡터를 사용하는 아이디어는 seq2seq 모델에도 영향을 미쳤습니다.

통계적 기계 번역 사용

- 번역 부분에서는, Statistical Machine Translate (통계적 기계 번역을 사용함)
- 통계적 기계 번역(Statistical Machine Translation, SMT)은 1990년대부터 2010년대까지 주요한 기계 번역 방법론이었습니다. SMT는 대규모 양방향 병렬 텍스트 코퍼스에서 통계적 패턴을 학습하여 원문을 대상 언어로 번역하는 방식을 사용합니다. SMT는 대량의 데이터에서 통계적 패턴을 학습하는 능력이 있지만, 복잡한 언어 구조와 패턴을 완전히 이해하거나 학습하는 데는 한계가 있습니다.
- 2016년 이후로 성능이 압도적으로 증가함. → 2016년 이후 구글 부터 시작해서 번역 시스템이 모두 바뀜



54

NMT: the first big success story of NLP Deep Learning

Neural Machine Translation went from a fringe research attempt in **2014** to the leading standard method in **2016**

- **2014**: First seq2seq paper published
- **2016**: Google Translate switches from SMT to NMT – and by 2018 everyone has

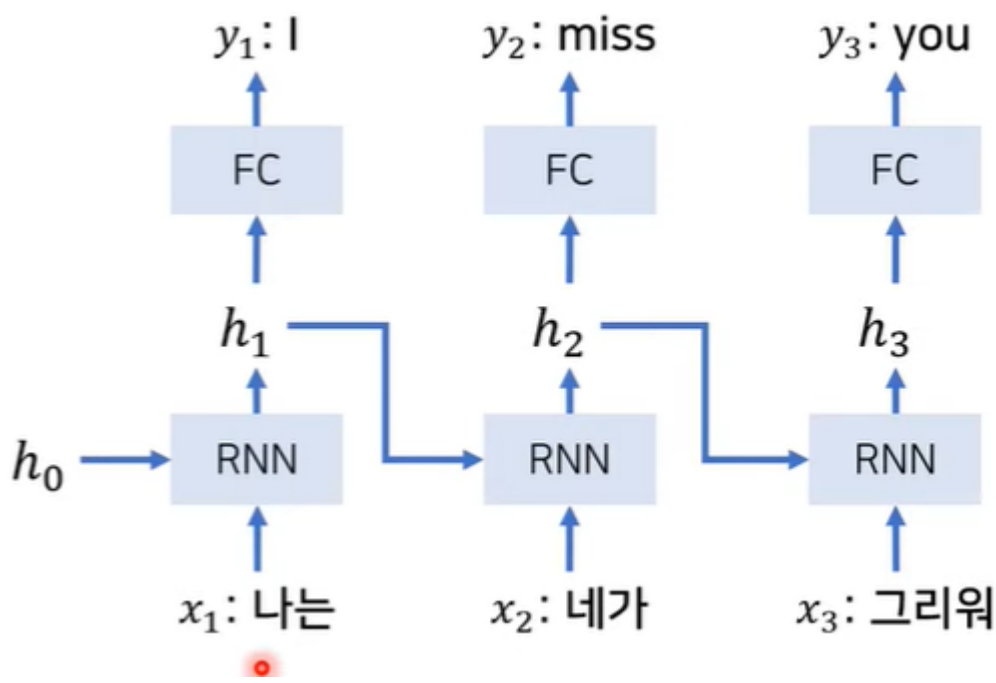
Microsoft SYSTRAN beyond language Google facebook

Baidu 百度 網易 NetEase www.163.com Tencent 腾讯 搜狗搜索

- This is amazing!
  - **SMT** systems, built by hundreds of engineers over many years, outperformed by NMT systems trained by a small group of engineers in a few months

RNN을 이용한 sequence to sequence

- $h_t = sigmoid(W^{hx}x_t + W^{hh}h_{t-1})$
- $y_t = W^{yh}h_t$



- 위의 사진과 같이 입력 토큰과 출력 토큰의 개수 즉 크기가 같다.
- h의 경우에는 지금까지 들어왔던 문장에 대한 정보를 포함.
- 따라서 입력 sequence 과 출력 sequence 가 다른 경우에는 활용 x, 또한 한 단어의 입력당 하나의 출력을 가지기 때문에 문맥이 다른 언어를 번역할 경우 정확성이 떨어지게 된다.

3. 논문에서 제시한 아이디어

- In this paper, we present a general end-to-end approach to sequence learning that makes minimal assumptions on the sequence structure.
- 일반적인 시퀀스-투-시퀀스 문제를 해결하기 위한 방법을 제시하는 것입니다. 이를 위해, 저자는 장기 단기 메모리(LSTM) 아키텍처의 직접적인 적용을 사용하여, **입력 시퀀스를 고정된 차원의 벡터로 매핑하고**, 그 벡터에서 출력 시퀀스를 추출하는 방식을 제안합니다.

- 이 아이디어는 입력 시퀀스를 한 번에 한 타임스텝씩 읽어 큰 고정 차원 벡터 표현을 얻고, 그 벡터에서 출력 시퀀스를 추출하기 위해 또 다른 LSTM을 사용하는 것입니다.

4. 구현

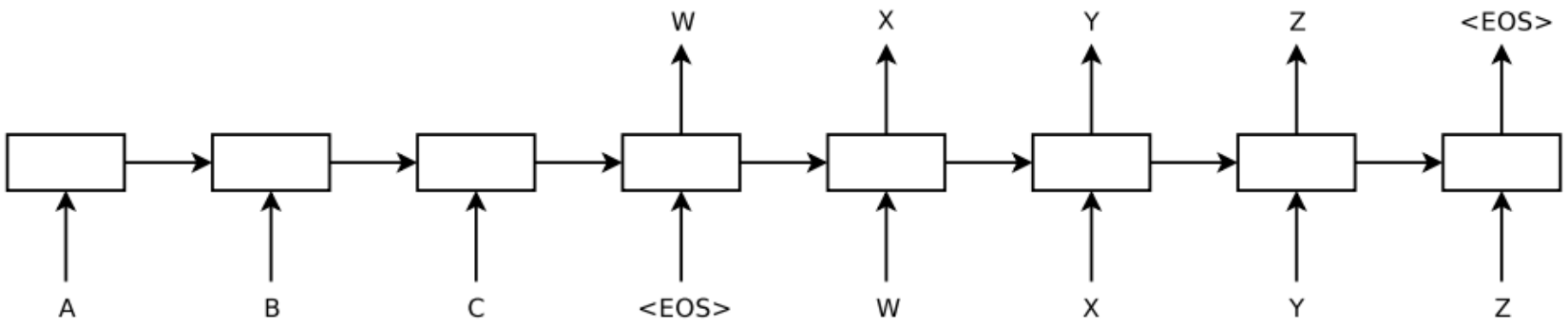
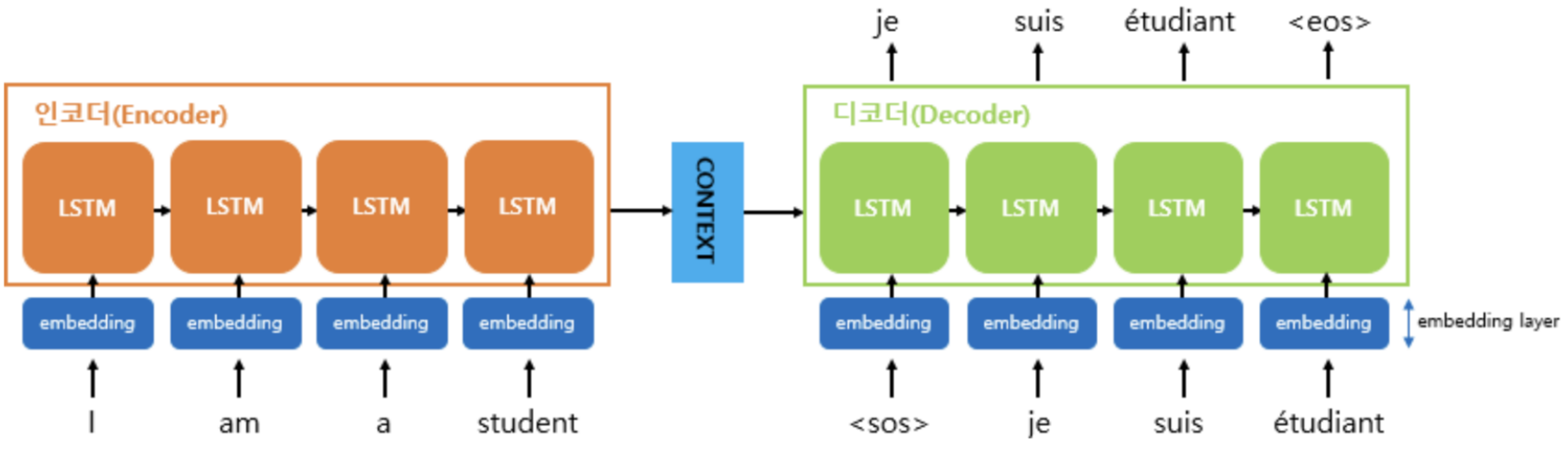
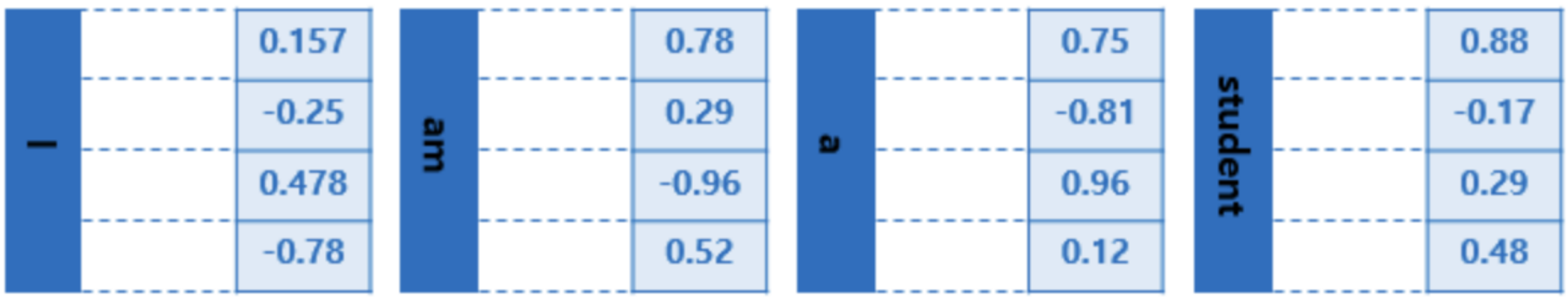


Figure 1: Our model reads an input sentence "ABC" and produces "WXYZ" as the output sentence. The model stops making predictions after outputting the end-of-sentence token. Note that the LSTM reads the input sentence in reverse, because doing so introduces many short term dependencies in the data that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

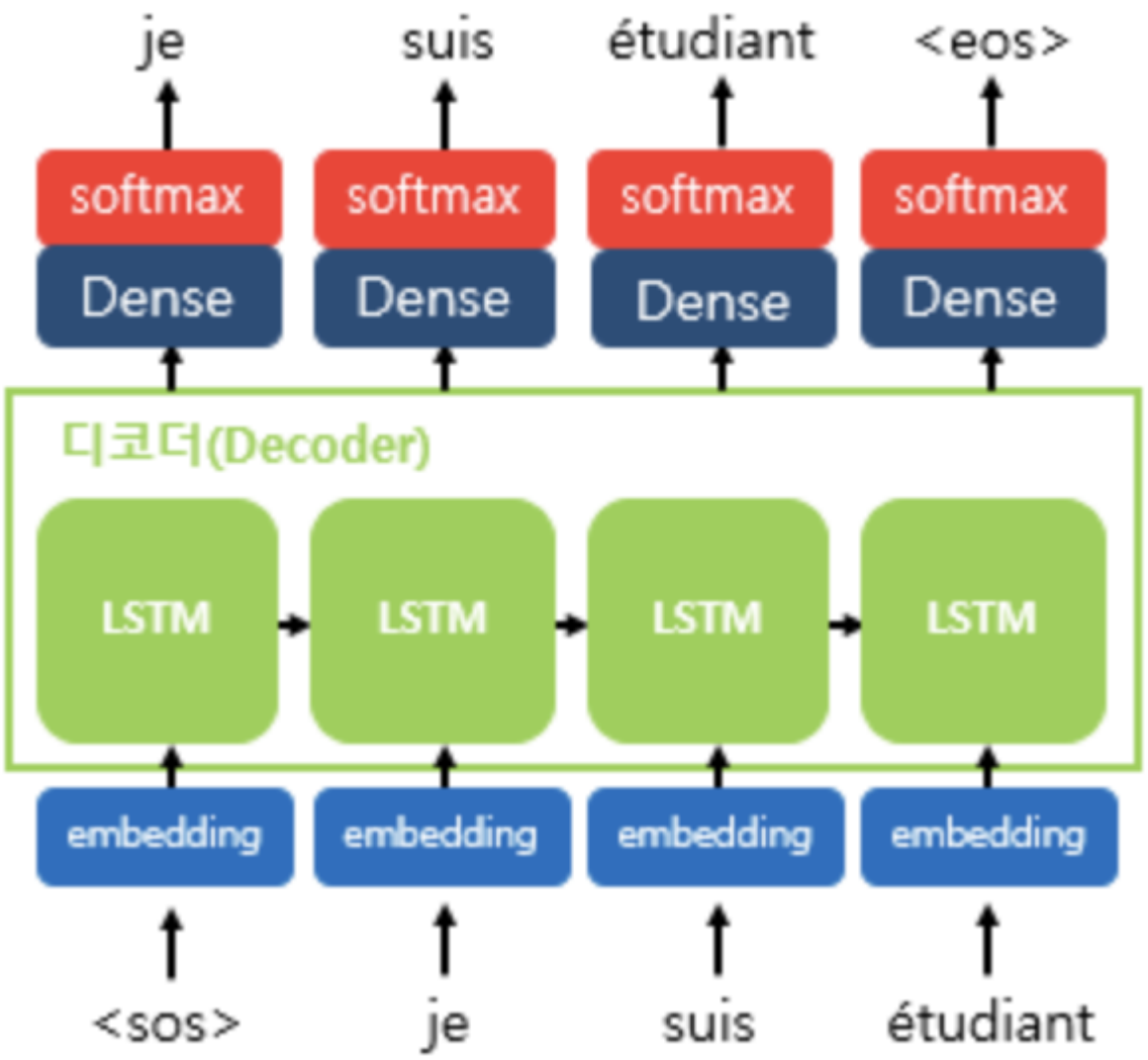
- 훈련 과정에서는 디코더에게 인코더가 보낸 컨텍스트 벡터와 실제 정답인 상황인 <sos> je suis étudiant를 입력 받았을 때, je suis étudiant <eos>가 나와야 된다고 정답을 알려주면서 훈련합니다. 이후 context와 <sos> 토큰을 넣어주면 결과값을 만들어 냅니다.
- **Teacher forcing** : 훈련 과정에서는 이전 시점의 디코더 셀의 출력을 현재 시점의 디코더 셀의 입력으로 넣어주지 않고, 이전 시점의 실제 값을 현재 시점의 디코더 셀의 입력값으로 하는 방법을 사용할 겁니다. 그 이유는 이전 시점의 디코더 셀의 예측이 틀렸는데 이를 현재 시점의 디코더 셀의 입력으로 사용하면 현재 시점의 디코더 셀의 예측도 잘못될 가능성이 높고 이는 연쇄 작용으로 디코더 전체의 예측을 어렵게 합니다. 이런 상황이 반복되면 훈련 시간이 느려집니다. 이러한 문제를 해결하고자 모든 시점에 대해서 이전 시점의 예측값 대신 실제값을 입력으로 주는 방법을 Teacher forcing이라고 합니다.



- seq2seq에서 사용되는 모든 단어들은 임베딩 벡터로 변환 후 입력으로 사용됩니다. 위 그림은 모든 단어에 대해서 임베딩 과정을 거치게 하는 단계인 임베딩 층(embedding layer)의 모습을 보여줍니다.
- 이때 인코더의 LSTM과 디코더의 LSTM의 파라미터는 다른 값을 사용합니다.



- 디코더의 마지막 층에서 각 단어에 대한 점수(raw score)를 계산한 뒤, 이 점수를 softmax 함수를 통해 확률로 변환합니다. (여러 단어중 선택하는 기능)



pseudo code

인코더는 입력 시퀀스를 고정 길이의 벡터로 인코딩합니다.

- 입력 시퀀스의 길이 = T
- 각 단어 = X_t
- LSTM 셀 = h_t
- 반환값 = 입력 단어들의 정보를 담고있는 인코더

```
function ENCODER(X):
  h_t = INITIAL_STATE
  for t = 1 to T:
    h_t = LSTM(X_t, h_t)
  end for
  return h_t
end function
```

디코더는 인코더의 출력 벡터를 기반으로 목표 시퀀스를 생성합니다.

- 인코더의 상태 = c
- 이전 시점의 출력 토큰 = y_{t-1}
- 각 출력값들 = Y
- T' = 최대 출력 길이

```
function DECODER(c, y_{t-1}):
  s_t = INITIAL_STATE
  for t = 1 to T':
    s_t = LSTM([y_{t-1}, c], s_t) # 인코더 + 이전단어, 이전상태 -> 상태 업데이트
    y_t = SOFTMAX(s_t) # 각 생성 단어
  end for
  return Y
end function
```

5. 실험

두 가지 방법으로 WMT’14 English to French MT task를 수행함.

1. SMT를 사용하지 않은 번역 작업 : NMT
2. SMT 기준선의 n-best 목록을 재 점수화 하는 데 사용

데이터셋

- WMT’14 English to French, 348M의 프랑스어 단어와 304M의 영어 단어로 구성된 12M 문장의 하위 집합에서 모델을 훈련시킴. → 토큰화된 훈련 및 테스트 세트와 1000-best 목록이 공개적으로 사용 가능
- 어휘 사전에 없는 모든 단어는 특수한 "UNK" 토큰으로 대체되었습니다.

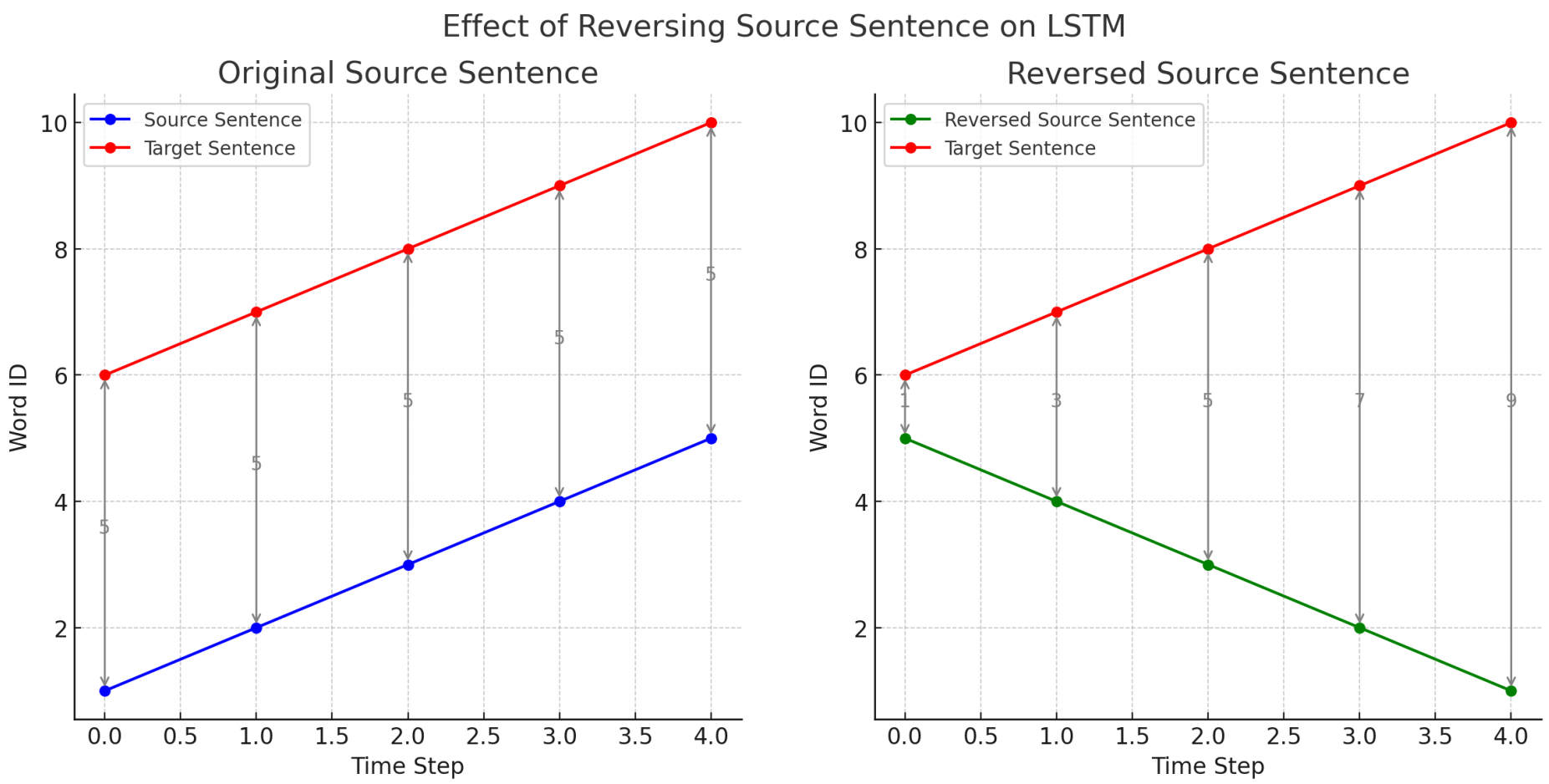
디코딩(번역 작업)

1. **훈련 과정**: LSTM 모델은 소스 문장 *S*가 주어졌을 때 올바른 번역 *T*를 생성하도록 훈련됩니다. 이 때 사용되는 '목표 함수'는 올바른 번역의 로그 확률을 최대화하는 것입니다.
2. **번역 생성**: 훈련이 끝나면, 모델은 소스 문장을 입력받아 가장 확률이 높은 번역 문장을 생성합니다. 이를 위해 "Beam Search"이라는 특별한 알고리즘을 사용합니다.
3. **Beam Search**: 이 알고리즘은 번역의 '부분적인' 문장들을 동시에 고려하면서 가장 가능성 높은 완전한 문장을 찾습니다. 이 '부분적인' 문장들을 "부분 가설"이라고 부릅니다.

재점수화(SMT와 함께 사용)

- **기존 번역 리스트**: 이미 다른 기계 번역 시스템 (여기서는 SMT)에서 생성된 '상위 1000개의 가능한 번역' 목록이 있습니다.
- **재점수화**: LSTM 모델을 사용해서 이 상위 1000개 번역 각각에 대한 '새로운' 확률 점수를 계산합니다. 그리고 이 새로운 점수를 기존의 점수와 평균내어, 최종적으로 가장 높은 점수를 받은 번역을 선택합니다.

Reversing the Source Sentences



- LSTM이 긴 시간 의존성 문제를 해결할 수 있는 능력이 있음에도 불구하고, 소스 문장을 반전시킬 때 더 잘 학습한다는 사실을 발견했다고 설명하고 있습니다. 즉, 소스 문장의 단어 순서를 뒤집으면 LSTM의 테스트 복잡도(perplexity)가 5.8에서 4.7로, 테스트 BLEU 점수가 25.9에서 30.6으로 상승했다고 합니다.
- 소스 문장을 뒤집으면 (오른쪽 그래프), 대상 문장의 처음 몇 개 단어와의 '시간적 거리'가 크게 줄어듭니다.
- 이로 인해 LSTM이 더 쉽게 정보를 전달할 수 있고, 따라서 학습이 더 잘 이루어집니다.

Training details

모델 구조

- 4개의 계층을 가진 깊은 LSTM을 사용했습니다. → **Deep Network**

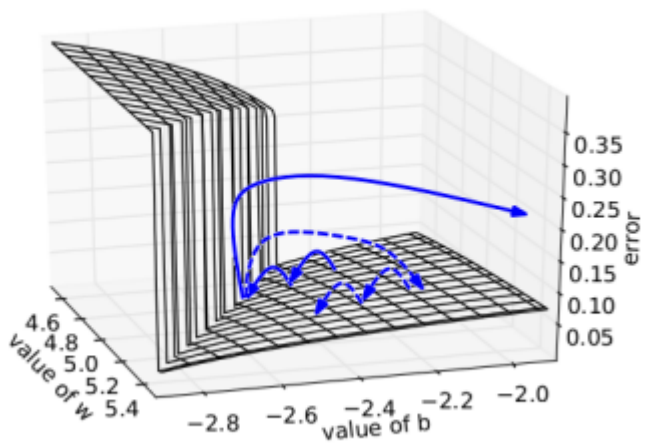
- 각 계층에는 1000개의 셀이 있고, 1000차원의 단어 임베딩을 사용했습니다. → **복잡성**
- 입력 어휘 크기는 160,000이고, 출력 어휘 크기는 80,000입니다. → **적당한 크기 제한**
- 따라서 깊은 LSTM은 문장을 표현하기 위해 8000개의 실수를 사용합니다. → **복잡성**
- 결과적으로 이 LSTM 모델은 총 384M 개의 매개변수를 가지고 있습니다. → **오버피팅 방지**
- 깊은 LSTM은 얕은 LSTM보다 훨씬 더 좋은 성능을 보였습니다.
- 추가적인 계층은 퍼플렉서티(perplexity)를 거의 10%씩 줄였습니다. → **낮아질 수록 더 좋은 성능**

학습 알고리즘 초기화

- 모든 LSTM의 매개변수는 -0.08과 0.08 사이의 균일 분포로 초기화되었습니다. → **빠르고 안정적으로 수렴**
- 확률적 경사 하강법(SGD)을 사용했고, 모멘텀은 사용하지 않았습니다. → **더 좋은 성능**
- 학습률은 0.7로 고정되었고, 5번의 에폭(epoch) 후에는 학습률을 반으로 줄였습니다. → **정밀 조정**

배치와 그래디언트

- 배치 크기는 128이며, 이를 사용하여 그래디언트를 계산했습니다. → **메모리 효율**
- LSTM은 소실 그래디언트 문제(vanishing gradient problem)에는 잘 노출되지 않지만, 폭발하는 그래디언트(exploding gradients) 문제는 발생할 수 있습니다. 이를 방지하기 위해 그래디언트의 노름에 제약을 두었습니다. → **Gradient Clipping :** 기울기 폭주를 막기 위해 임계값을 넘지 않도록 값을 자릅니다. 기울기의 방향은 유지한 채 크기만 달라집니다.



문장 길이와 계산 효율

- 문장의 길이가 다르므로, 미니배치 내에서는 대략적으로 같은 길이의 문장을 함께 묶어서 2배의 속도 향상을 달성했습니다. → 길이가 다르게 묶으면 가장 긴 길이의 문장에 맞춰야 되기 때문에 **비효율적**

Parallelization

- 단일 GPU에서는 초당 약 1,700단어를 처리할 수 있었는데, 이 속도는 연구 목적에는 너무 느렸습니다. 따라서 8-GPU 머신을 사용하여 모델을 병렬화했습니다.
- 모델에는 총 4개의 LSTM 계층이 있고, 각 계층은 별도의 GPU에서 실행되었습니다. 계산이 완료되는 즉시, 해당 계층의 GPU는 다음 계층의 GPU로 활성화 값을 전달했습니다.
- 남은 4개의 GPU는 소프트맥스 계산을 병렬화하는 데 사용되었습니다. 각 GPU는 1000 × 20000 크기의 행렬에 곱셈을 수행했습니다.
- 이러한 병렬화 구현을 통해 모델은 초당 6,300단어를 처리할 수 있게 되었고, 미니배치 크기는 128로 설정되었습니다.
- 병렬화된 구현을 사용하여 학습은 대략 열 일 정도 소요되었습니다.

Experimental Results

LSTM의 성능 (WMT'14 영어-프랑스어 테스트 세트)

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.

- 조합된 5개의 역방향 LSTM, Beam size가 12일때 가장 성능이 좋다.
- 참고로, 5개의 역방향 LSTM, Beam size가 2인것이 가장 저렴하다. 성능은 비슷
- 역방향 LSTM이 큰 규모의 MT작업에서 높은 성능을 달성할 수 있음.

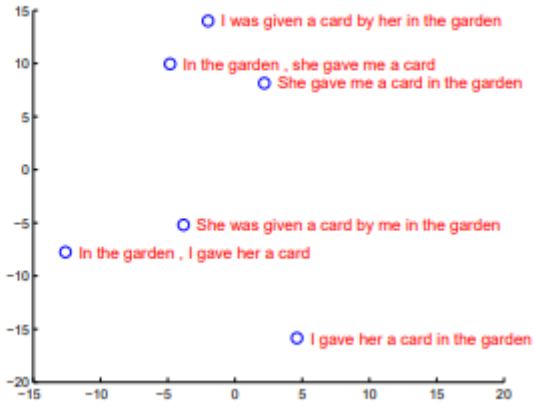
신경망과 SMT 시스템을 함께 사용한 방법

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

Table 2: Methods that use neural networks together with an SMT system on the WMT'14 English to French test set (ntst14).

- **5개의 역방향 LSTM 앙상블로 기준선 1000-Best 사용**: 36.5의 BLEU 점수, 앙상블 방법이 더 높은 성능을 보였습니다.
- LSTM은 기존의 SMT 시스템과 함께 사용될 때도 높은 성능을 보이며, 이는 신경망 기반의 번역 시스템이 SMT와 잘 통합될 수 있음을 나타냅니다.

Model Analysis



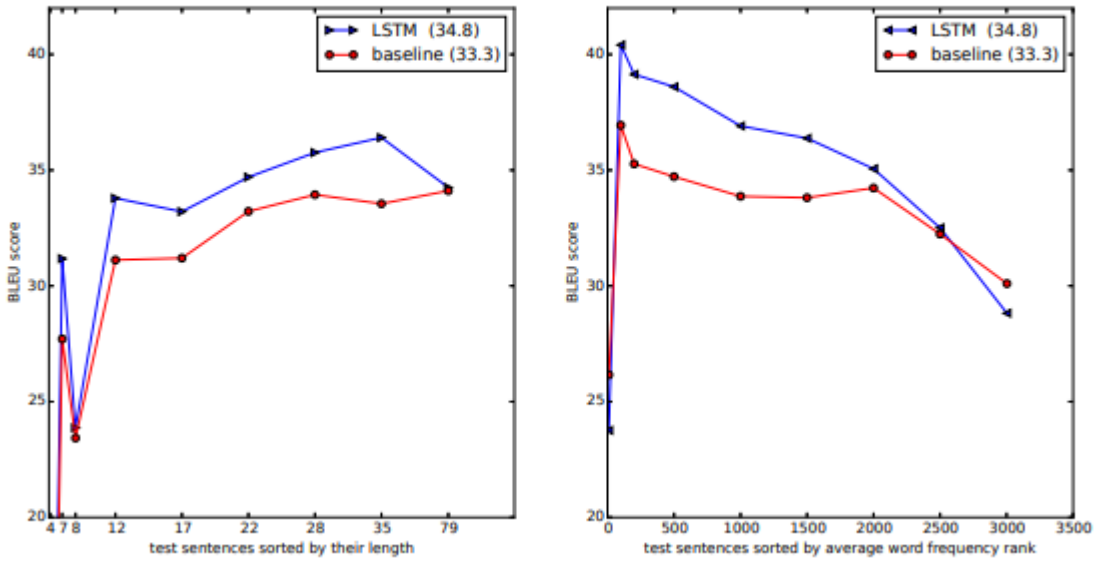
- 이 그림은 문구를 처리한 후 얻어진 LSTM의 은닉 상태의 2차원 PCA(주성분 분석) 투영을 보여줍니다. 문구들은 **주로 단어의 순서에 따른 의미로 군집화**되어 있습니다. 이는 순서를 고려하지 않는 bag-of-words 모델로는 캡처하기 어려운 특성입니다. → "단어의 순서에 따른 의미로 군집화"라는 부분은, 같은 문맥이나 의미를 가진 문장이나 문구가 그래프 상에서 가까이 위치하게 된다는 것을 의미합니다.

- 이 설명은 LSTM의 은닉 상태가 단어의 순서를 고려한 의미적인 정보를 잘 캡처하고 있다는 것을 시각적으로 보여줍니다. 또한, 이러한 의미적 군집화는 단순한 bag-of-words 모델로는 어려운 작업임을 강조하고 있습니다.

Type	Sentence
Our model	Ulrich UNK , membre du conseil d' administration du constructeur automobile Audi , affirme qu' il s' agit d' une pratique courante depuis des années pour que les téléphones portables puissent être collectés avant les réunions du conseil d' administration afin qu' ils ne soient pas utilisés comme appareils d' écoute à distance .
Truth	Ulrich Hackenberg , membre du conseil d' administration du constructeur automobile Audi , déclare que la collecte des téléphones portables avant les réunions du conseil , afin qu' ils ne puissent pas être utilisés comme appareils d' écoute à distance , est une pratique courante depuis des années .
Our model	" Les téléphones cellulaires , qui sont vraiment une question , non seulement parce qu' ils pourraient potentiellement causer des interférences avec les appareils de navigation , mais nous savons , selon la FCC , qu' ils pourraient interférer avec les tours de téléphone cellulaire lorsqu' ils sont dans l' air " , dit UNK .
Truth	" Les téléphones portables sont véritablement un problème , non seulement parce qu' ils pourraient éventuellement créer des interférences avec les instruments de navigation , mais parce que nous savons , d' après la FCC , qu' ils pourraient perturber les antennes-relais de téléphonie mobile s' ils sont utilisés à bord " , a déclaré Rosenker .
Our model	Avec la crémation , il y a un " sentiment de violence contre le corps d' un être cher " , qui sera " réduit à une pile de cendres " en très peu de temps au lieu d' un processus de décomposition " qui accompagnera les étapes du deuil " .
Truth	Il y a , avec la crémation , " une violence faite au corps aimé " , qui va être " réduit à un tas de cendres " en très peu de temps , et non après un processus de décomposition , qui " accompagnerait les phases du deuil " .

Table 3: A few examples of long translations produced by the LSTM alongside the ground truth translations. The reader can verify that the translations are sensible using Google translate.

- LSTM이 생성한 긴 번역의 몇 가지 예시와 정확한 번역(ground truth)을 나란히 보여줍니다. 독자는 Google 번역을 사용하여 번역이 타당한지 확인할 수 있습니다.
- 이 예시들을 통해 LSTM 모델이 긴 문장에서도 의미론적으로 타당한 번역을 생성할 수 있음을 확인할 수 있습니다.



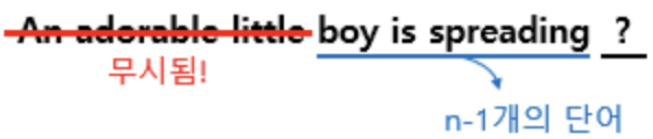
- 왼쪽 그래프는 문장의 길이에 따른 우리 시스템의 성능을 보여줍니다. x축은 테스트 문장들을 길이별로 정렬한 것으로, 실제 시퀀스 길이로 표시되어 있습니다. 35단어 미만의 문장에서는 성능 저하가 없으며, 가장 긴 문장에서도 약간의 성능 저하만 발생합니다. → **장기 의존성 문제 해결**
- 오른쪽 그래프는 점점 더 희귀한 단어가 포함된 문장에서 LSTM의 성능을 보여줍니다. 여기서 x축은 "평균 단어 빈도 순위"로 테스트 문장들을 정렬한 것입니다. → **희귀 단어에 대한 일반화 성능**

6. 결론

- 이 연구에서는 제한된 어휘와 거의 문제 구조에 대한 가정이 없는 큰 깊은 LSTM이 대규모 MT 작업에서 어휘가 무제한인 표준 SMT 기반 시스템을 능가할 수 있음을 보였습니다. 우리의 간단한 LSTM 기반 접근법이 MT에서 성공했다는 것은 충분한 훈련 데이터가 제공되는 한 다른 많은 시퀀스 학습 문제에서도 잘 작동할 것이라고 제안합니다.
- 가장 중요하게는, 간단하고 직접적이며 상대적으로 최적화되지 않은 접근법이 SMT 시스템을 능가할 수 있음을 보였으므로, 추가 작업은 더 높은 번역 정확도를 가져올 가능성이 높습니다. 이러한 결과는 이 접근법이 다른 어려운 시퀀스-투-시퀀스 문제에서도 잘 작동할 것이라고 제안합니다.

Appendix

BLEU SORE



$$P(w|\text{boy is spreading}) = \frac{\text{count}(\text{boy is spreading } w)}{\text{count}(\text{boy is spreading})}$$

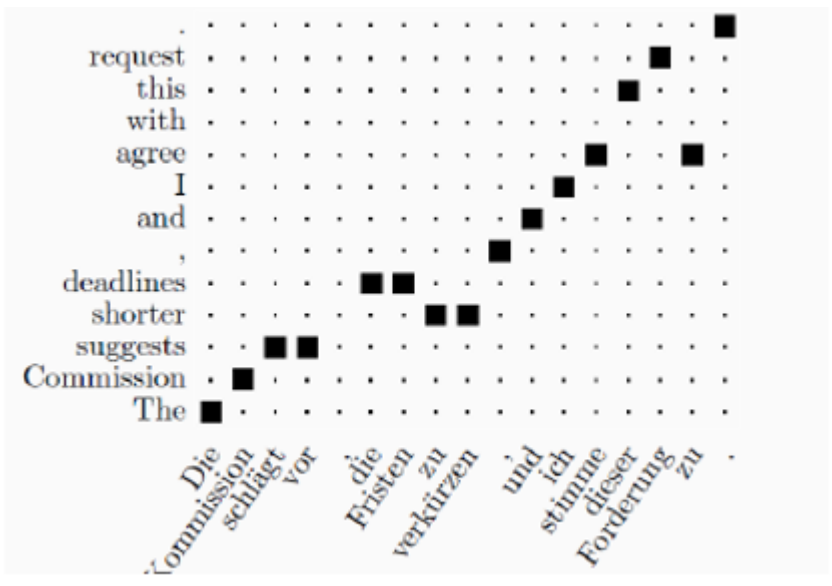
만약 갖고있는 코퍼스에서 boy is spreading가 1,000번 등장했다고 합시다. 그리고 boy is spreading insults가 500번 등장했으며, boy is spreading smiles가 200번 등장했다고 합시다. 그렇게 되면 boy is spreading 다음에 insults가 등장할 확률은 50%이며, smiles가 등장할 확률은 20%입니다. 확률적 선택에 따라 우리는 insults가 더 맞다고 판단하게 됩니다.

$$P(\text{insults}|\text{boy is spreading}) = 0.500$$

$$P(\text{smiles}|\text{boy is spreading}) = 0.200$$

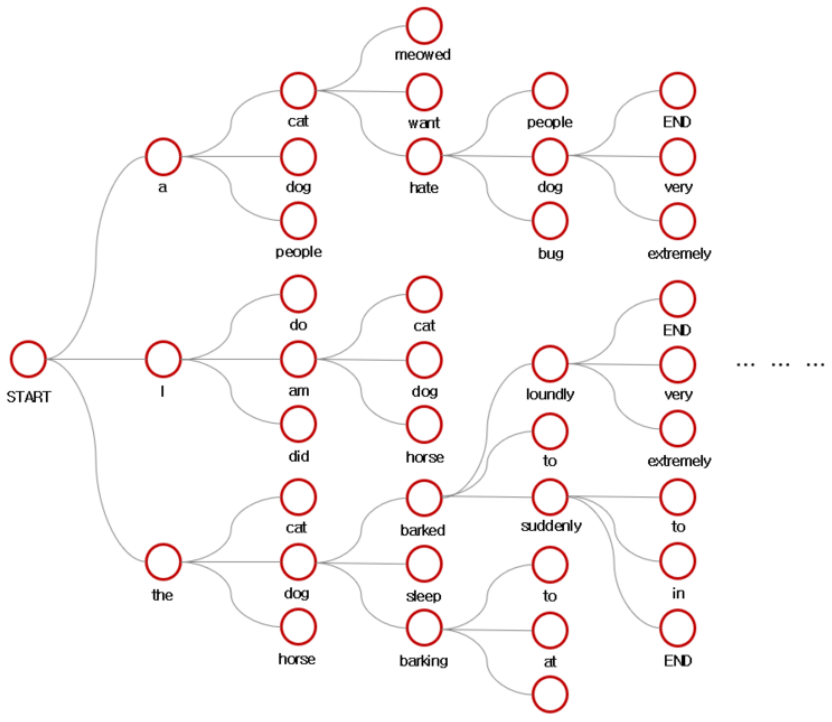
- 기계 번역 결과와 사람이 직접 번역한 결과가 얼마나 유사한지 비교하여 번역에 대한 성능을 측정하는 방법
- 측정 기준은 N-GRAM에 기반

SMT



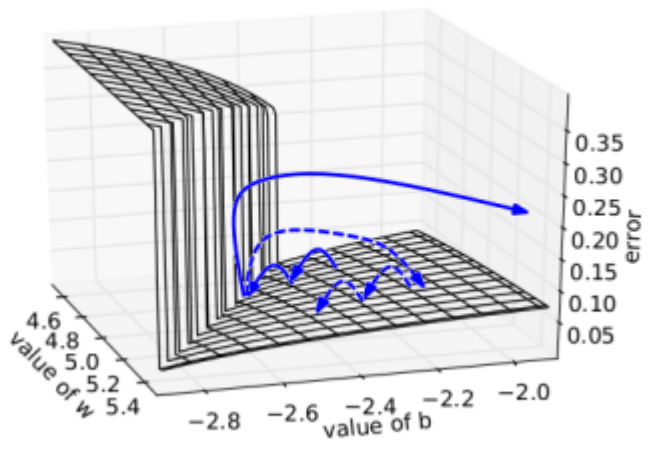
- https://m.blog.naver.com/bcj1210/221581535580
- 번역 모델은 말 그대로 번역 할 언어(Source Language)와 번역 될 언어(Target Language)의 번역 모델을 만드는 것으로 병렬 말뭉치가 필요합니다.

beam search



- https://blog.naver.com/PostView.naver?blogId=sooftware&logNo=221809101199
- 해당 시점에서 가장 확률이 높은 후보를 선택하는 것이다. k 갯수만큼 선택함 (그림 k=3)

Gradient Clipping



- 그래디언트 클리핑은 파라미터를 업데이트할 때, 그래디언트가 너무 크다고 판단되면 그 값을 줄이는 방식으로 작동합니다. 이는 그래디언트의 크기가 너무 커져서 발생하는 너무 큰 학습 단계를 방지하고, 학습의 안정성을 유지하는데 도움이 됩니다.

출처

https://wikidocs.net/24996

https://web.stanford.edu/class/cs224n/

https://aclanthology.org/D13-1176.pdf

https://wikidocs.net/61375

질문

1. Seq2Seq 모델이 기존에 사용되던 통계적 기계 번역(SMT)이나 CNN 기반의 접근법과 비교하여 어떤 장점과 단점이 있는지 설명해주실 수 있나요?

   ▼ 답변

   - SMT의 단점 → SMT는 고정된 구조와 통계 패턴에 의존합니다. 유연한 번역 x
   - CNN은 지역적인 패턴을 잘 캡처하지만, 긴 범위의 의존성을 처리하는 데 제한이 있을 수 있습니다.
   - CNN은 학습이 상대적으로 쉽고 안정적일 수 있는 반면, Seq2Seq 모델은 긴 시퀀스에 대한 학습이 어려울 수 있으며, 종종 특별한 기술(예: 어텐션 메커니즘)이 필요합니다.

2. Teacher forcing 기법이 훈련 과정에서 어떻게 도움이 되는지, 그리고 이 기법을 사용하지 않았을 때 어떤 문제가 발생할 수 있는지 설명해 주실 수 있나요?

▼ 답변

eacher forcing은 실제 이전 출력을 다음 입력으로 사용함으로써 훈련을 더 안정적이고 빠르게 만듭니다. 이를 사용하지 않으면, 잘못된 예측이 후속 예측에 영향을 미쳐 훈련 시간이 느려질 수 있습니다.

3. 기존 RNN을 사용한 sequence to sequence 모델에서 입력 토큰과 출력 토큰의 개수가 같아야 했다고 하셨는데, 이 제한이 왜 발생하는지 그리고 어떻게 이 문제를 해결했는지 자세히 설명해주실 수 있나요?

▼ 답변

기존의 간단한 RNN 구조에서는 입력과 출력 시퀀스를 처리하는 데 동일한 네트워크를 사용하곤 했습니다. 이 경우, 네트워크의 각 시간 단계는 특정 입력 토큰과 해당 출력 토큰 사이의 매핑을 학습해야 하므로, 입력과 출력 시퀀스의 길이가 동일해야 했습니다.

인코더는 입력 시퀀스를 고정된 길이의 컨텍스트 벡터로 변환하고, 디코더는 이 벡터를 사용하여 출력 시퀀스를 생성합니다. 이 구조는 입력과 출력 시퀀스의 길이가 다르더라도 문제가 되지 않습니다.

4. 임베딩 층은 각 단어를 벡터로 변환한다고 말씀하셨는데, 이 변환 과정이 어떻게 이루어지는지, 그리고 이 임베딩이 Seq2Seq 모델의 성능에 어떤 영향을 미치는지 설명해주실 수 있나요?

▼ 답변

- **의미적 표현의 캡처**: 임베딩은 단어의 의미를 수치 벡터로 표현합니다. 이 벡터는 단어가 사용되는 맥락과 관련된 정보를 포함하므로, 모델은 문장의 구조와 의미를 더 잘 이해할 수 있습니다.
- **계산 효율성**: 단어를 원-핫 벡터로 표현하는 것보다 훨씬 효율적입니다. 원-핫 벡터는 어휘 사전의 크기만큼의 차원을 가지지만, 임베딩 벡터는 훨씬 작은 차원을 가질 수 있어 계산 효율성이 향상됩니다.
- **일반화 능력의 향상**: 임베딩은 의미상 유사한 단어가 서로 가까이 위치하도록 합니다. 따라서 모델은 훈련 데이터에 없던 단어에 대해서도 일반화할 수 있는 능력을 향상시킵니다.
- **Seq2Seq 모델과의 연계**: Seq2Seq 모델에서 임베딩 층은 단어 수준의 정보를 벡터로 변환하여 LSTM 같은 순환 신경망에 공급합니다. 이렇게 하면, 모델은 단어 간의 관계와 문장 구조를 더 정확하게 캡처할 수 있으며, 번역, 요약 등의 작업에서 더 정확한 결과를 생성할 수 있습니다.

5. RNN 대신에 LSTM을 사용했을때 더 높은 정확도를 보이는 이유?

▼ 답변

RNN이 가지고 있던 단점을 LSTM이 보강해줌으로 RNN보다 LSTM에 장기 의존성의 문제도 존재하지 않고 더 긴 LONG SETP 문제도 해결할 수 있기 때문에 더 높은 정확도를 보입니다.

6. 퍼플렉서티(Perplexity)가 뭔가요?

▼ 답변

자연어 처리에서 모델의 성능을 평가하는 지표 중 하나, 낮을 수록 좋다.