신경망 기계 번역
정렬 및 번역을 공동으로 학습하여

드미트리 바흐다나우
제이콥스 대학교 브레멘, 독일

조경현 **요슈아** 벤지오*
몬트리올 대학교

초록

신경망 기계 번역은 최근에 제안된 기계 번역 접근 방식입니다. 기존의 통계적 기계 번역과 달리 신경망 기계 번역은 번역 성능을 극대화하기 위해 공동으로 조정할 수 있는 단일 신경망을 구축하는 것을 목표로 합니다. 최근 신경망 기계 번역을 위해 제안된 모델은 인코더-디코더 제품군에 속하는 경우가 많으며, 소스 문장을 디코더가 번역을 생성하는 고정 길이 벡터로 인코딩합니다. 이 논문에서는 고정 길이 벡터의 사용이 이러한 기본 인코더-디코더 아키텍처의 성능 개선에 병목 현상을 일으킨다고 가정하고, 이러한 부분을 하드 세그먼트로 명시적으로 구성하지 않고도 모델이 대상 단어 예측과 관련된 소스 문장의 일부를 자동으로 (소프트) 검색할 수 있도록 함으로써 이를 확장할 것을 제안합니다. 이러한 새로운 접근 방식을 통해 영어-프랑스어 번역 작업에서 기존의 최첨단 구문 기반 시스템과 비슷한 수준의 번역 성능을 달성했습니다. 또한 정성적 분석 결과, 모델이 발견한 (소프트) 정렬이 우리의 직관과 잘 일치하는 것으로 나타났습니다.

1 소개

인공신경망 기계 번역은 최근 Kalchbrenner와 Blunsom(2013), Sutskever *외*(2014), Cho *외*(2014b)에 의해 제안된 기계 번역에 대한 새로운 접근 방식입니다. 신경망 기계 번역은 개별적으로 조정되는 여러 개의 작은 하위 구성 요소로 구성된 기존의 구문 기반 번역 시스템(예: Koehn *et al*., 2003 참조)과 달리 문장을 읽고 정확한 번역을 출력하는 하나의 대규모 신경망을 구축하고 학습시키려고 시도합니다.

제안된 대부분의 신경 기계 번역 모델은 *인코더-디코더* 제품군에 속하며(Sutskever *외*., 2014; Cho *외*., 2014a), 각 언어마다 인코더와 디코더가 있거나 각 문장에 언어별 인코더를 적용한 다음 출력을 비교하는 방식을 사용합니다(Hermann and Blunsom, 2014). 인코더 신경망은 소스 문장을 읽고 고정 길이 벡터로 인코딩합니다. 그런 다음 디코더는 인코딩된 벡터에서 번역을 출력합니다. 한 언어 쌍에 대한 인코더와 디코더로 구성된 전체 인코더-디코더 시스템은 소스 문장이 주어졌을 때 정확한 번역 확률을 최대화하기 위해 공동으로 학습됩니다.

이 인코더-디코더 접근 방식의 잠재적인 문제는 신경망이 소스 문장의 모든 필요한 정보를 고정 길이 벡터로 압축할 수 있어야 한다는 것입니다. 이로 인해 신경망이 긴 문장, 특히 훈련 말뭉치의 문장보다 긴 문장을 처리하기 어려울 수 있습니다. Cho *등*(2014b)은 입력 문장의 길이가 길어질수록 기본 인코더-디코더의 성능이 급격히 저하된다는 사실을 보여주었습니다.

이 문제를 해결하기 위해 공동으로 정렬하고 번역하는 방법을 학습하는 인코더-디코더 모델에 확장 기능을 도입했습니다. 제안된 모델은 번역에서 단어를 생성할 때마다 소스 문장에서 가장 관련성이 높은 정보가 집중된 위치 집합을 (소프트) 검색합니다. 그런 다음 모델은 이러한 소스 위치와 이전에 생성된 모든 타깃 단어와 연관된 문맥 벡터를 기반으로 타깃 단어를 예측합니다.

[*]CIFAR 선임 연구원

이 접근 방식이 기본 인코더-디코더와 가장 중요한 차이점은 전체 입력 문장을 하나의 고정 길이 벡터로 인코딩하지 않는다는 점입니다. 대신 입력 문장을 일련의 벡터로 인코딩하고 번역을 디코딩하는 동안 이러한 벡터의 하위 집합을 적응적으로 선택합니다. 이렇게 하면 신경 번역 모델이 길이에 관계없이 원본 문장의 모든 정보를 고정 길이 벡터로 쪼갤 필요가 없습니다. 이를 통해 모델이 긴 문장에 더 잘 대처할 수 있음을 보여줍니다.

이 논문에서는 정렬과 번역을 공동으로 학습하는 제안된 접근 방식이 기본 인코더-디코더 접근 방식에 비해 번역 성능이 크게 향상되었음을 보여줍니다. 이러한 효과는 긴 문장에서 더욱 뚜렷하게 나타나지만, 모든 길이의 문장에서 관찰할 수 있습니다. 영어-프랑스어 번역 작업에서 제안된 접근 방식은 단일 모델로 기존 구문 기반 시스템과 비슷하거나 근접한 번역 성능을 달성합니다. 또한 정성적 분석을 통해 제안된 모델이 소스 문장과 해당 목표 문장 간에 언어적으로 그럴듯한 (소프트) 정렬을 찾아내는 것으로 나타났습니다.

2 배경: 신경망 기계 번역

확률론적 관점에서 번역은 소스 문장 **x가** 주어졌을 때 **y의** 조건부 확률을 최대화하는 대상 문장 **y를** 찾는 것과 같습니다(즉, arg $\max_{\mathbf{y}} p(\mathbf{y} \mid \mathbf{x})$). 신경망 기계 번역에서는 병렬 학습 말뭉치를 사용하여 문장 쌍의 조건부 확률을 최대화하기 위해 매개변수화된 모델을 맞춥니다. 번역 모델에 의해 조건부 분포가 학습되면 소스 문장이 주어지면 조건부 확률을 최대화하는 문장을 검색하여 해당 번역을 생성할 수 있습니다.

최근 여러 논문에서 이러한 조건부 분포를 직접 학습하기 위해 신경망을 사용할 것을 제안했습니다(예: Kalchbrenner and Blunsom, 2013; Cho *et al*., 2014a; Sutskever *et al*., 2014; Cho *et al*., 2014b; Forcada and $\tilde{\text{N}}$eco, 1997 참조). 이 신경 기계 번역 접근 방식은 일반적으로 두 가지 구성 요소로 구성되는데, 첫 번째 구성 요소는 소스 문장 **x를** 인코딩하고 두 번째 구성 요소는 대상 문장 **y로** 디코딩합니다. 예를 들어, (Cho *et al*., 2014a)와 (Sutskever *et al*., 2014)에서는 두 개의 순환 신경망(RNN)을 사용하여 가변 길이의 소스 문장을 고정 길이 벡터로 인코딩하고 벡터를 가변 길이의 대상 문장으로 디코딩합니다.

신경망 기계 번역은 매우 새로운 접근 방식임에도 불구하고 이미 유망한 결과를 보여주고 있습니다. Sutskever *등*(2014)은 장단기 기억(LSTM) 유닛을 갖춘 RNN을 기반으로 한 신경 기계 번역이 영어-프랑스어 번역 작업에서 기존의 구문 기반 기계 번역 시스템의 최첨단 성능에 근접하는 결과를 얻었다고 보고했습니다.[1] 예를 들어, 기존 번역 시스템에 신경 구성 요소를 추가하여 구문 테이블의 구문 쌍에 점수를 매기거나(Cho *et al*., 2014a) 후보 번역의 순위를 재지정하는 등의 작업을 수행하면 기존의 최첨단 성능 수준을 뛰어넘을 수 있습니다(Sutskever *et al*., 2014).

2.1 RNN 인코더-디코더

여기에서는 정렬과 번역을 동시에 학습하는 새로운 아키텍처를 구축하는 기반 프레임워크인 *RNN 인코더-디코더에 대해* 간략하게 설명합니다(Cho *et al.* (2014a) 및 Sutskever *et al*. (2014)).

인코더-디코더 프레임워크에서 인코더는 입력 문장, 벡터 시퀀스를 읽습니다.
x = (x_1 , - - - , x_T), 벡터 *c로 변환합니다.*[2] 가장 일반적인 접근 방식은 다음과 같이 RNN을 사용하는 것입니다.

고

그

리

$$h_t = f(x_t, h_{t-1}) \tag{1}$$

$$c = q(\{h_1, \cdots, h_T\}),$$

여기서 $h_t \in \mathbb{R}^n$ 은 t 시점의 숨겨진 상태이고, c는 숨겨진 상태의 시퀀스에서 생성된 벡터입니다. f와 q는 비선형 함수입니다. 예를 들어, Sutskever 등(2014)은 f와 $q(\{h_1, \cdots, h_T\}) = h_T$ 로 LSTM을 사용했습니다.

[1] 최첨단 성능이란 신경망 기반 구성 요소를 사용하지 않고도 기존 구문 기반 시스템( )의 성능, 즉 신경망 기반 구성 요소를 사용하지 않고도 기존 구문 기반 시스템( )의 성능을 의미합니다.

[2] 대부분의 선행 연구(예: Cho *et al*., 2014a; Sutskever *et al*., 2014; Kalchbrenner and Blunsom, 2013)에서는 가변 길이 입력 문장을 *고정 길이* 벡터로 인코딩하는 데 사용했지만, 나중에 설명하겠지만 반드시 그럴 필요는 없으며 심지어 *가변 길이* 벡터를 사용하는 것이 더 유리할 수도 있습니다.

디코더는 종종 문맥 벡터 c와 이전에 예측된 모든 단어 $\{y_1, \ldots, y_{t'-1}\}$이 주어지면 다음 단어 $y_{t'}$를 예측하도록 훈련됩니다. 즉, 디코더는 공동 확률을 정렬된 조건부로 분해하여 번역 $\mathbf{y}$에 대한 확률을 정의합니다:

$$p(\mathbf{y}) = \prod_{t=1}^{T} p(y_t \mid \{y_1, \ldots, y_{t-1}\}, c), \tag{2}$$

여기서 $\mathbf{y} = y_1, \ldots, y_T$ y . RNN을 사용하면 각 조건 확률은 다음과 같이 모델링됩니다.

$$p(y_t \mid \{y_1, \ldots, y_{t-1}\}, c) = g(y_{t-1}, s_t, c), \tag{3}$$

여기서 g는 y_t의 확률을 출력하는 비선형, 잠재적으로 다층적인 함수이고, s_t는 RNN의 숨겨진 상태입니다. RNN과 합성곱 신경망의 하이브리드와 같은 다른 아키텍처도 사용할 수 있다는 점에 유의해야 합니다(Kalchbrenner and Blunsom, 2013).

3 정렬 및 번역 학습

이 섹션에서는 신경망 기계 번역을 위한 새로운 아키텍처를 제안합니다. 새로운 아키텍처는 인코더(3.2절)로서의 양방향 RNN과 번역을 디코딩하는 동안 소스 문장을 검색하는 것을 에뮬레이션하는 디코더(3.1절)로 구성됩니다.

3.1 DECODER: 일반 설명

새로운 모델 아키텍처에서는 방정식 (2)의 각 조건부 확률을 다음과 같이 정의합니다:

$$p(y_i \mid y_1, \ldots, y_{i-1}, \mathbf{x}) = g(y_{i-1}, s_i, c_i), \tag{4}$$

여기서 s_i는 시간 i에 대한 RNN 숨겨진 상태로, 다음과 같이 계산됩니다.

$$s_i = f(s_{i-1}, y_{i-1}, c_i).$$

기존 인코더-디코더 아프로치(식 (2) 참조)와 달리 여기서는 각 대상 단어 y_i에 대해 별개의 컨텍스트 벡터 c_i를 조건으로 확률이 결정된다는 점에 유의해야 합니다.

컨텍스트 벡터 c_i는 인코더가 입력 문장을 매핑하는 일련의 *주석*$(h_1, \ldots, h_T)$에 따라 달라집니다. 각 어노테이션 h_i에는 전체 입력 시퀀스에 대한 정보가 포함됩니다.
입력 시퀀스의 *i번째*단어를 둘러싼 부분에 중점을 둡니다. 다음 섹션에서 주석이 어떻게 결합되는지 자세히 설명합니다.

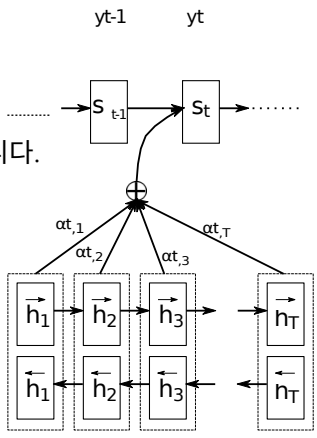
그러면 컨텍스트 벡터 c_i는 이러한 주석의 가중치 합계 h_i로 계산됩니다:

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j. \tag{5}$$

각 어노테이션 h_j의 가중치 α_{ij}는 다음과 같이 계산됩니다.

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}, \tag{6}$$

어디



X X X X1 2 3 T

그림 1: 소스 문장$(x_1, x_2, \ldots, x_T)$이 주어졌을 때, 제안된 모델에서 단어 y_t를 생성하려고 시도하는 그래픽 일루전.

EXP $(E)_{ik}$

$$e_{ij} = a(s_{i-1}, h)_j$$

는 위치 j 주변의 입력과 위치 i의 출력이 얼마나 잘 일치하는지 점수를 매기는 *정렬 모델입니다*. 점수는 RNN 숨겨진 상태 s_{i-1} (y_i , 방정식 (4)를 방출하기 직전)와 입력 문장의 **j 번째** 주석 h_j 을 기반으로 합니다.

정렬 모델 a를 제안된 시스템의 다른 모든 구성 요소와 공동으로 학습되는 피드포워드 신경망으로 매개변수화합니다. 기존 기계 번역과는 다릅니다,

로 설정하면 정렬은 잠재 변수로 간주되지 않습니다. 대신 정렬 모델이 직접 소프트 정렬을 컴퍼팅하여 비용 함수의 기울기가 역전파될 수 있도록 합니다. 이 기울기는 정렬 모델뿐만 아니라 전체 번역 모델을 함께 훈련하는 데 사용할 수 있습니다.

모든 *주석의* 가중치 합을 취하는 접근 방식은 가능한 정렬에 대한 기대치를 계산하는 것으로 이해할 수 있습니다. α_{ij} 는 대상 단어 y_i 가 소스 단어 x_j 에 정렬되거나 이로부터 번역될 확률이라고 가정합니다. 그런 다음, i 번째 문맥 벡터 c_i 는 α_{ij} 의 확률로 모든 주석에 대해 예상되는 주석입니다.

확률 α_{ij} 또는 관련 에너지 e_{ij} 는 다음 상태 s_i 를 결정하고 y_i 를 생성할 때 이전 숨겨진 상태 s_{i-1} 에 대한 주석 h_j 의 중요성을 반영합니다. 직관적으로, 이것은 디코더에서 주의 메커니즘을 구현합니다. 디코더는 소스 문장의 어떤 부분에 주의를 기울일지 결정합니다. 디코더가 주의 메커니즘을 갖도록 함으로써 인코더가 소스 문장의 모든 정보를 고정 길이 벡터로 인코딩해야 하는 부담을 덜어줍니다. 이 새로운 접근 방식을 사용하면 주석 시퀀스 전체에 정보를 분산시킬 수 있으며, 이에 따라 디코더가 선택적으로 검색할 수 있습니다.

3.2 인코더: 시퀀스에 주석을 달기 위한 양방향 RNN

식 (1)에 설명된 일반적인 RNN은 입력 시퀀스 $\mathbf{x}$를 첫 번째 심볼 x_1 부터 마지막 심볼 x_T x 까지 순서대로 읽습니다. 그러나 제안하는 방식에서는 각 단어의 주석이 앞의 단어뿐만 아니라 뒤의 단어도 요약하기를 원합니다. 따라서 우리는 최근 음성 인식에 성공적으로 사용되고 있는 양방향 RNN(BiRNN, Schuster and Paliwal, 1997)을 사용할 것을 제안합니다(예: Graves *et al.*, 2013 참조).

BiRNN은 순방향 및 역방향 RNN으로 구성됩니다. 순방향 RNN $\overrightarrow{f}$ 은 입력 시퀀스를 읽습니다. 순서대로 (x에서 to x) 및 *순방향 숨겨진 상태* $\overrightarrow{h}$ 시퀀스를 계산합니다. (H_1 , - - - , $H X_T$).

역방향 RNN $\overleftarrow{f}$ 는 시퀀스를 역순으로 읽습니다(x_T x에서 x_1), 그 결과 *역방향 숨겨진 상태의* 시퀀스 $(\overleftarrow{h}_1, \cdots, \overleftarrow{h}_{T_x})$.

(H_1 , - - - , $H X_T$).

각 단어 x_j 에 대한 어노테이션을 얻습니다. 순방향 숨겨진 상태 $\overrightarrow{h}_j$ 와 역방향 $\overleftarrow{h}_j$ 를 연결하여 $h_j = \begin{bmatrix} \overrightarrow{h}_j^\top ; \overleftarrow{h}_j^\top \end{bmatrix}^\top$ 그리고 . 이런 식으로 주석 h_j 에는 요약이 포함되어 있습니다.

의 앞 단어와 뒤 단어 모두에 주석을 달 수 있습니다. RNN은 최근 입력을 더 잘 표현하는 경향이 있기 때문에 주석 h_j 는 x_j 주변의 단어에 집중됩니다. 이 주석 시퀀스는 나중에 디코더와 정렬 모델에서 컨텍스트 벡터를 계산하는 데 사용됩니다(방정식 (5)-(6)).

제안된 모델의 그래픽 그림은 그림 1을 참조하세요.

4 실험 설정

영어에서 프랑스어로 번역하는 작업에 대해 제안된 접근법을 평가합니다. ACL WMT '14에서 제공하는 이중 언어 병렬 코퍼스를 사용합니다.[3] 비교를 위해 최근 Cho 등(2014a)이 제안한 RNN 인코더-디코더의 성능도 보고합니다. 두 모델 모두 동일한 훈련 절차와 동일한 데이터 세트를 사용했습니다.[4]

4.1 데이터 세트

WMT '14에는 다음과 같은 영어-프랑스 병렬 말뭉치가 포함되어 있습니다: 유로팔(6,100만 단어), 뉴스 해설(550만 단어), UN(421만 단어), 그리고 각각 9,000만 단어와 2억 7,250만 단어의 크롤링 된 말뭉치 2개, 총 8억 5,000만 단어입니다. Cho *et al.* (2014a)에 설명된 절차에 따라 Axelrod *et al*. (2011)의 데이터 선택 방법을 사용하여 결합 코퍼스의 크기를 3억 4,800만 단어로 줄였습니다.[5] 인코더를 사전 훈련하기 위해 훨씬 더 큰 단일 언어 말뭉치를 사용할 수도 있지만, 앞서 언급한 병렬 코퍼스 이외의 단일 언어 데이터는 사용하지 않습니다. 뉴스-테스트-

[3] http://www.statmt.org/wmt14/translation-task.html

[4] 구현은 https://github.com/lisa-groundhog/GroundHog 에서 확인할 수 있습니다.

[5] 온라인에서 http://www-lium.univ-lemans.fr/~schwenk/cslm_joint_paper/에서 확인할 수 있습니다.

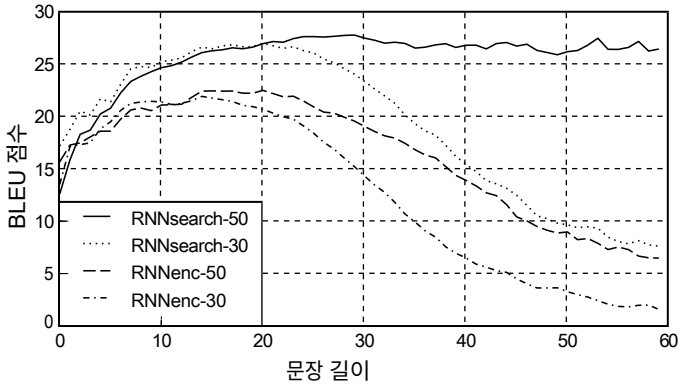


그림 2: 문장 길이와 관련하여 테스트 세트에서 생성된 번역의 BLEU 점수. 결과는 모델에 알려지지 않은 단어가 포함된 문장을 포함하는 전체 테스트 세트에 대한 것입니다.

2012 및 news-test-2013을 사용하여 개발(검증) 세트를 만들고, 훈련 데이터에 없는 3003개의 문장으로 구성된 WMT '14의 테스트 세트(news-test-2014)로 모델을 평가합니다.

일반적인 토큰화[6]을 수행한 후 각 언어에서 가장 빈번하게 사용되는 단어 30,000개의 후보 목록을 사용하여 모델을 학습시킵니다. 후보 목록에 포함되지 않은 모든 단어는 특수 토큰([UNK])에 매핑됩니다. 데이터에 소문자 또는 어간 제거와 같은 다른 특별한 전처리는 적용하지 않습니다.

4.2 모델

두 가지 유형의 모델을 훈련합니다. 첫 번째는 RNN 인코더-디코더(RNNencdec, Cho *et al.*, 2014a)이고, 다른 하나는 제안된 모델로서, 우리는 이를 RNNsearch라고 부릅니다. 각 모델을 두 번 훈련합니다. 먼저 최대 30단어 길이의 문장(RNNencdec-30, RNNsearch-30)으로 훈련하고, 그 다음 최대 50단어 길이의 문장(RNNencdec-50, RNNsearch-50)으로 훈련합니다.

RNNencdec의 인코더와 디코더에는 각각 1000개의 숨겨진 유닛이 있습니다.[7] RNNsearch의 인코더는 각각 1000개의 숨겨진 유닛을 가진 순방향 및 역방향 순환 신경망(RNN)으로 구성됩니다. 디코더에는 1000개의 숨겨진 유닛이 있습니다. 두 경우 모두 각 대상 단어의 조건부 확률을 계산하기 위해 단일 최대치(Goodfellow *et al.*, 2013) 숨겨진 레이어를 가진 다층 네트워크를 사용합니다 (Pascanu *et al*., 2014).

각 모델을 훈련하기 위해 Adadelta(Zeiler, 2012)와 함께 미니배치 확률적 경사 하강(SGD) 알고리즘을 사용합니다. 각 SGD 업데이트 방향은 80개의 센텐스로 구성된 미니배치를 사용하여 계산됩니다. 각 모델을 약 5일 동안 훈련했습니다.

모델이 학습되면 빔 검색을 사용하여 조건부 확률을 대략적으로 최대화하는 번역을 찾습니다(예: Graves, 2012; Boulanger-Lewandowski *et al.*, 2013 참조). Sutskever *등*(2014)은 신경망 기계 번역 모델에서 번역을 생성하는 데 이 접근 방식을 사용했습니다.

실험에 사용된 모델의 아키텍처와 훈련 절차에 대한 자세한 내용은 부록 A와 B를 참조하세요.

5 결과

5.1 정량적 결과

표 1에는 BLEU 점수로 측정한 번역 성능이 나열되어 있습니다. 표에서 알 수 있듯이 모든 경우에서 제안된 RNNsearch가 기존 RNNencdec보다 성능이 뛰어납니다. 더 중요한 것은 알려진 단어로 구성된 문장만 고려했을 때 RNNsearch의 성능이 기존 구문 기반 번역 시스템(Moses)의 성능만큼

높다는 점입니다. 이는 RNNsearch와 RNNencdec 학습에 사용한 병렬 말뭉치 외에 별도의 단일 언어 말뭉치(4억 1,800만 단어)를 사용한다는 점을 고려할 때 상당한 성과입니다.

[6] 저희는 오픈 소스 기계 번역 패키지인 Moses의 토큰화 스크립트를 사용했습니다.

[7] 본 문서에서 '숨겨진 유닛'이란 항상 게이트형 숨겨진 유닛을 의미합니다(부록 A.1.1 참조).

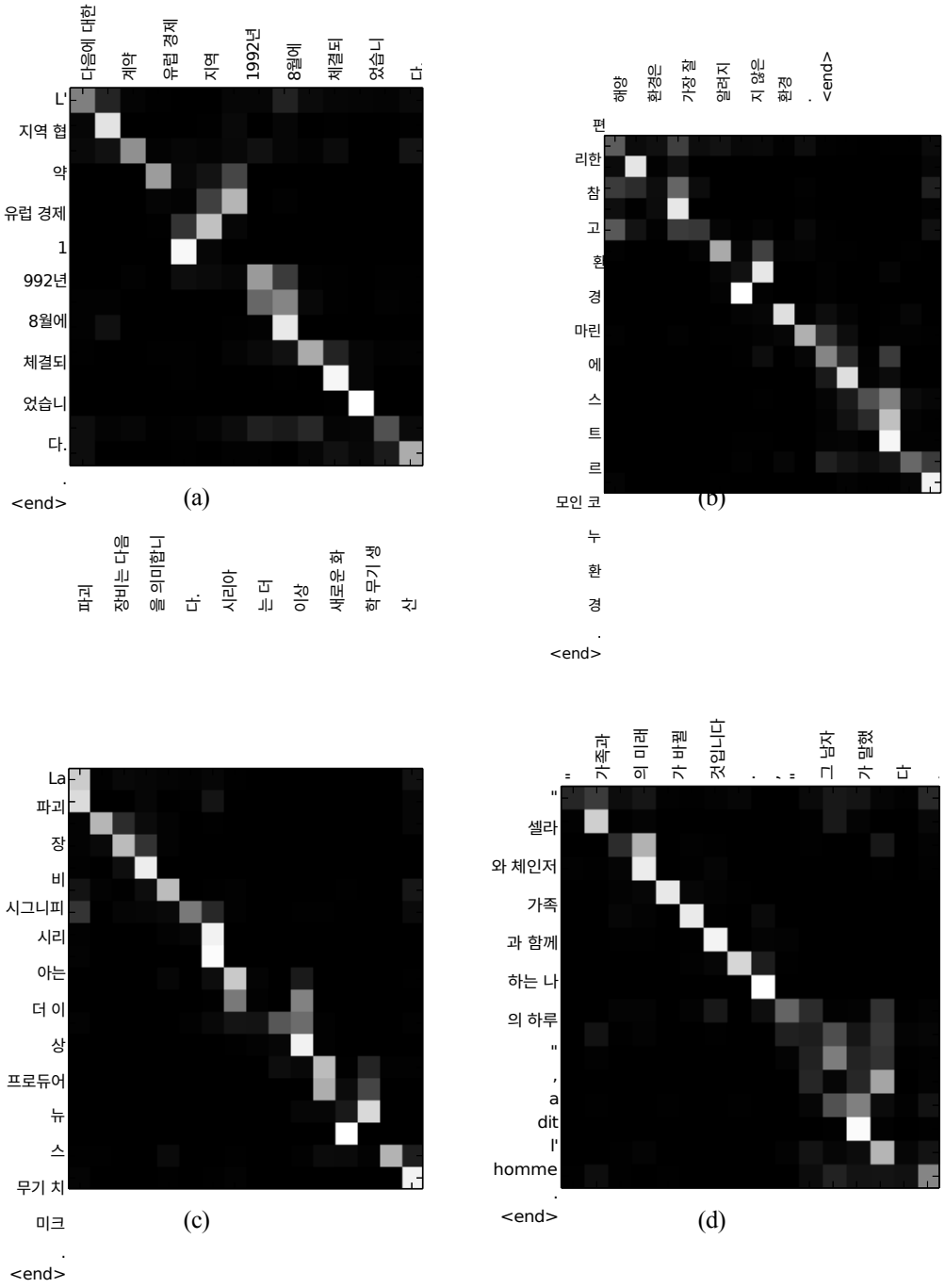


그림 3: RNNsearch-50에서 찾은 4개의 샘플 정렬. 각 플롯의 X축과 Y축은 각각 소스 문장(영어)과 생성된 번역(프랑스어)의 단어에 해당합니다. 각 픽셀은 i번째 목표 단어에 대한 j번째 소스 단어의 주석 가중치 α_{ij} 를 회색조(0: 검정, 1: 흰색)로 표시합니다(방정식 (6) 참조). (a) 임의의 문장. (b-d) 테스트 세트에서 알 수 없는 단어가 없고 길이가 10~20단어인 문장 중에서 무작위로 선택한 3개의 샘플.

제안된 접근 방식의 동기 중 하나는 기본 인코더-디코더 접근 방식에서 고정 길이 컨텍스트 벡터를

사용했기 때문입니다. 이러한 제한으로 인해 기본 인코더-디코더 접근 방식이 긴 문장에서 성능이 저하될 수 있다고 추측했습니다. 그림 2를 보면 문장의 길이가 길어질수록 RNNencdec의 성능이 급격히 떨어지는 것을 볼 수 있습니다. 반면, RNNsearch-30과 RNNsearch-50은 문장의 길이에 더 강합니다. 특히 RNNsearch-50은 50개 이상의 문장에서도 성능 저하가 나타나지 않았습니다. 기본 인코더-디코더에 비해 제안한 모델의 이러한 우수성은 RNNsearch-30이 RNNencdec-50보다 성능이 뛰어나다는 사실에서도 확인할 수 있습니다(표 1 참조).

모델	모두	UNK 없음(◦)
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

표 1: 테스트 세트에 투입된 학습된 모델의 BLEU 점수. 두 번째와 세 번째 열은 각각 모든 문장, 그리고 모르는 단어가 없는 문장 자체와 참조 번역에 대한 점수를 보여줍니다. RNNsearch-50⋆ 은 개발 세트의 성능이 개선되지 않을 때까지 훨씬 더 오래 훈련되었습니다. (◦) 모델이 [UNK]를 생성하는 것을 허용하지 않았습니다.
알 수 없는 문장만 있는 경우 토큰
단어가 평가되었습니다(마지막 열).

5.2 정성적 분석

5.2.1 정렬

제안된 접근 방식은 생성된 번역의 단어와 소스 문장의 단어 간의 (소프트) 정렬을 검사하는 직관적인 방법을 제공합니다. 이는 그림 3에서와 같이 식 (6)의 주석 가중치 α_{ij} 를 시각화하여 수행됩니다. 각 플롯에서 행렬의 각 행은 주석과 관련된 가중치를 나타냅니다. 이를 통해 대상 단어를 생성할 때 소스 문장에서 어느 위치가 더 중요하게 고려되었는지 알 수 있습니다.

그림 3의 정렬을 보면 영어와 프랑스어 사이의 단어 정렬이 대체로 단조롭다는 것을 알 수 있습니다. 각 행렬의 대각선을 따라 강한 가중치를 볼 수 있습니다. 그러나 사소하지 않고 단조롭지 않은 정렬도 많이 관찰됩니다. 형용사와 명사는 일반적으로 프랑스어와 영어에서 서로 다른 순서를 가지며, 그림 3 (a)에서 그 예를 볼 수 있습니다. 이 그림에서 모델이 [유럽 경제 지역]이라는 구문을 [zone e´conomique europe´en]으로 올바르게 번역하는 것을 볼 수 있습니다. RNNsearch는 [zone]과 [Area]를 올바르게 정렬하여 두 단어([European] 및 [Economic])를 건너뛴 다음, 한 번에 한 단어씩 되돌아가 전체 구문 [zone e´conomique europe´enne]을 완성할 수 있었습니다.

예를 들어 그림 3 (d)에서 하드 얼라인먼트와 반대되는 소프트 얼라인먼트의 강점을 알 수 있습니다. [남자]라는 원문 구문이 [남자]로 번역된 것을 생각해 보세요. 하드 얼라인먼트는 [the]를 [l']로, [man]을 [homme]로 매핑합니다. 이는 번역에 도움이 되지 않는데, [the] 뒤에 오는 단어를 고려하여 [le], [la], [les] 또는 [l']로 번역할지 여부를 결정해야 하기 때문입니다. 소프트 얼라인먼트는 모델이 [the]와 [man]을 모두 살펴보도록 함으로써 이 문제를 자연스럽게 해결하며, 이 예제에서는 모델이 [the]를 [l']로 정확하게 번역할 수 있음을 알 수 있습니다. 그림 3에 제시된 모든 사례에서 유사한 동작을 관찰할 수 있습니다. 소프트 정렬의 또 다른 이점은 일부 단어를 아무데도 매핑하지 않고([NULL]) 직관적이지 않은 방식으로 매핑할 필요 없이 길이가 다른 소스 구문과 대상 구문을 자연스럽게 처리한다는 것입니다(예: Koehn, 2010의 4장 및 5장 참조).

5.2.2 긴 문장

그림 2에서 명확하게 볼 수 있듯이 제안된 모델(RNNsearch)은 긴 문장을 번역할 때 기존 모델(RNNencdec)보다 훨씬 더 우수합니다. 이는 RNNsearch가 긴 문장을 고정 길이 벡터로 완벽하게 인코딩할 필요 없이 입력 문장에서 특정 단어를 둘러싼 부분만 정확하게 인코딩하면 되기 때문일 수 있습니다.

예를 들어 테스트 세트의 이 소스 문장을 살펴보겠습니다:

입원 권한은 의사가 병원에서 의료 종사자로서의 지위에 따라 진단 또는 시술을

수행하기 위해 환자를 병원이나 의료 센터에 입원시킬 수 있는 권리입니다.

RNNencdec-50은 이 문장을 다음과 같이 번역했습니다:

입원 특권이란 의료인이 환자를 병원이나 의료 센터에 입원시키거나 자신의 건강 상태에 따라 진단을 내릴 수 있는 권리입니다.

RNNencdec-50은 [의료 센터]까지 원문 문장을 올바르게 번역했습니다. 그러나 그 이후(밑줄 친 부분)부터는 원문 문장의 원래 의미에서 벗어났습니다. 예를 들어, 원본 문장에서 [병원에서 의료 종사자로서의 그의 지위에 따라]를 [en fonction de son e´tat de sante´]("그의 건강 상태에 따라")로 대체했습니다.

반면에 RNNsearch-50은 입력 문장의 전체 의미를 생략하지 않고 다음과 같이 정확한 번역을 생성했습니다:

> *입원 특권이란 의료 종사자 신분에 따라 환자를 병원 또는 의료 센터에 입원시켜 진단 또는 치료를 시행할 수 있는 의료인의 권리입니다.*

테스트 세트의 다른 문장을 고려해 보겠습니다:

> *이러한 경험은 "시리즈의 수명을 연장하고 점점 더 중요해지고 있는 디지털 플랫폼을 통해 관객과 새로운 관계를 구축하기 위한 디즈니의 노력의 일환"이라고 그는 덧붙였습니다.*

RNNencdec-50의 번역은 다음과 같습니다.

> *이러한 유형의 경험은 "새로운 사람들의 수명을 연장하고 더 복잡하게 변화하는 수많은 독자와의 관계를 개발"하기 위한 디즈니의 이니셔티브의 일부입니다.*

이전 예제에서와 마찬가지로, 약 30개의 단어를 생성한 후 RNNencdec은 소스 문장의 실제 의미에서 벗어나기 시작했습니다(밑줄 친 문구 참조). 그 이후에는 닫는 따옴표가 없는 등 기본적인 실수가 발생하면서 번역의 품질이 저하됩니다.

이번에도 RNNsearch-50은 이 긴 문장을 정확하게 번역할 수 있었습니다:

> *이 장르의 경험은 "더 많은 중요한 플랫폼을 통해 대중과 새로운 관계를 형성하고 그들의 삶의 질을 연장"하기 위한 디즈니의 노력의 일부라고 덧붙였습니다.*

이미 제시된 정량적 결과와 함께 이러한 정성적 관찰은 RNNsearch 아키텍처가 표준 RNNencdec 모델보다 긴 문장을 훨씬 더 안정적으로 번역할 수 있다는 우리의 가설을 확고히 해줍니다.

부록 C에서는 참조 번역과 함께 RNNencdec-50, RNNsearch-50 및 Google 번역에서 생성된 긴 소스 문장의 샘플 번역을 몇 가지 더 제공합니다.

6 관련 작업

6.1 정렬하는 법 배우기

출력 기호를 입력 기호와 정렬하는 유사한 접근 방식은 최근 손글씨 합성의 맥락에서 Graves(2013)에 의해 제안되었습니다. 필기 합성은 모델에 주어진 문자 시퀀스의 필기를 생성하도록 요청하는 작업입니다. 그의 작업에서 그는 가우시안 커널의 혼합을 사용하여 주석의 가중치를 계산했으며, 각 커널의 위치, 너비 및 혼합 계수는 정렬 모델에서 예측했습니다. 보다 구체적으로, 그의 정렬은 위치가 단조롭게 증가하도록 위치를 예측하도록 제한되었습니다.

우리의 접근 방식과 가장 큰 차이점은 (Graves, 2013)에서는 주석의 가중치 모드가 한 방향으로만 이동한다는 점입니다. 기계 번역의 경우 문법적으로 올바른 번역(예: 영어-독일어)을 생성하기 위

해 (장거리) 재정렬이 필요한 경우가 많기 때문에 이는 심각한 제한 사항입니다.

반면에 우리의 접근 방식은 번역의 각 단어에 대해 소스 문장의 모든 단어의 주석 가중치를 계산해야 합니다. 대부분의 입력 및 출력 문장이 15~40단어에 불과한 번역 작업에서는 이러한 단점이 심각하지 않습니다. 그러나 이는 다른 작업에 대한 제안 체계의 적용 가능성을 제한할 수 있습니다.

6.2 기계 번역을 위한 신경망

벤지오 *등*(2003)이 신경망을 사용하여 앞 단어의 조건부 확률을 모델링하는 신경 확률 언어 모델을 소개한 이후 신경망은 기계 번역에 널리 사용되어 왔습니다. 그러나 신경망의 역할은 기존 통계적 기계 번역 시스템에 단순히 하나의 기능을 제공하거나 기존 시스템에서 제공한 번역 후보 목록의 순위를 다시 매기는 정도에 그쳤습니다.

예를 들어, Schwenk(2012)는 피드포워드 신경망을 사용하여 한 쌍의 소스 및 대상 구문의 점수를 계산하고 이 점수를 구문 기반 통계적 기계 번역 시스템의 추가 기능으로 사용할 것을 제안했습니다. 최근에는 Kalchbrenner와 Blunsom(2013)과 Devlin *등*(2014)이 기존 번역 시스템의 하위 구성 요소로 신경망을 성공적으로 사용했다고 보고했습니다. 전통적으로 목표 측 언어 모델로 훈련된 신경망은 후보 번역 목록의 점수를 재조정하거나 순위를 재조정하는 데 사용되었습니다(예: Schwenk *et al*., 2006 참조).

위의 접근 방식은 최첨단 기계 번역 시스템보다 번역 성능을 향상시키는 것으로 나타났지만, 우리는 신경망에 기반한 완전히 새로운 번역 시스템을 설계하는 보다 야심 찬 목표에 더 관심이 있습니다. 따라서 이 논문에서 고려하는 신경망 기계 번역 접근 방식은 이러한 이전 작업에서 급진적으로 벗어난 것입니다. 기존 시스템의 일부로 신경망을 사용하는 대신, 우리 모델은 자체적으로 작동하며 소스 문장에서 직접 번역을 생성합니다.

7 결론

인코더-디코더 접근 방식이라고 하는 신경 기계 번역에 대한 기존의 접근 방식은 전체 입력 문장을 번역을 디코딩할 고정 길이의 벡터로 인코딩합니다. 최근 Cho *등*(2014b)과 Pouget-Abadie *등*(2014)이 보고한 경험적 연구에 따르면 고정 길이 문맥 벡터를 사용하는 것은 긴 문장을 번역하는 데 문제가 있다고 합니다.

이 논문에서는 이 문제를 해결할 수 있는 새로운 아키텍처를 제안했습니다. 각 대상 단어를 생성할 때 모델이 입력 단어 세트 또는 인코더에 의해 입력된 주석을 (소프트) 검색하도록 함으로써 기본 인코더-디코더를 확장했습니다. 이렇게 하면 모델이 전체 소스 문장을 고정 길이 벡터로 인코딩할 필요가 없으며, 다음 목표 단어 생성에 관련된 정보에만 집중할 수 있습니다. 이는 신경망 기계 번역 시스템이 긴 문장에서 좋은 결과를 도출하는 능력에 큰 긍정적인 영향을 미칩니다. 기존 기계 번역 시스템과 달리 정렬 메커니즘을 포함한 번역 시스템의 모든 부분이 공동으로 학습되어 정확한 번역을 생성할 확률이 높아집니다.

제안한 모델인 RNNsearch를 영어-프랑스어 번역 작업에 적용하여 테스트했습니다. 실험 결과, 제안된 RNNsearch는 문장 길이에 관계없이 기존의 인코더-디코더 모델(RNNencdec)을 크게 능가하며, 소스 문장의 길이에 훨씬 더 잘 부합하는 것으로 나타났습니다. RNNsearch가 생성한 (소프트) 정렬을 조사한 정성적 분석을 통해 이 모델이 정확한 번역을 생성할 때 각 대상 단어를 소스 문장의 관련 단어 또는 주석과 올바르게 정렬할 수 있다는 결론을 내릴 수 있었습니다.

더 중요한 것은 제안된 접근 방식이 기존의 구문 기반 통계적 기계 번역에 필적하는 번역 성능을 달성했다는 점입니다. 제안된 아키텍처, 즉 신경망 기계 번역의 전체 제품군이 올해에야 제안되었다는 점을 고려하면 놀라운 결과입니다. 이번에 제안된 아키텍처는 더 나은 기계 번역과 자연어 전

반에 대한 더 나은 이해를 위한 유망한 단계라고 생각합니다.

앞으로 남은 과제 중 하나는 알려지지 않았거나 희귀한 단어를 더 잘 처리하는 것입니다. 이는 모델이 더 널리 사용되고 모든 상황에서 현재 최첨단 기계 번역 시스템의 성능과 일치하기 위해 필요한 작업입니다.

감사

저자들은 Theano의 개발자들에게 감사를 표합니다(Bergstra *외.*, 2010; Bastien *외.*, 2012). 연구 자금 및 컴퓨팅 지원을 위해 다음 기관의 지원에 감사드립니다: NSERC, Calcul Que´bec, Compute Canada, 캐나다 연구 의장 및 CIFAR. Bah- danau는 Planet Intelligent Systems GmbH의 지원에 감사드립니다. 또한 펠릭스 힐, 바트 반 메리엔보어, 장 푸제-아바디, 콜린 데빈, 김태호에게도 감사드립니다.

참조

Axelrod, A., He, X., Gao, J. (2011). 의사 도메인 내 데이터 선택을 통한 도메인 적응. *자연어 처리의 경험적 방법에 관한 ACL 컨퍼런스 (EMNLP)*, 355-362 페이지. 전산 언어학 협회.

Bastien, F., Lamblin, P., Pascanu, R., Bergstra, J., Goodfellow, I. J., Bergeron, A., Bouchard, N. 및 Bengio, Y. (2012). Theano: 새로운 기능과 속도 개선. 딥 러닝 및 비지도 특징 학습 NIPS 2012 워크샵.

Bengio, Y., Simard, P. 및 Frasconi, P. (1994). 경사 하강으로 장기 의존성을 학습하는 것은 어렵습니다. *IEEE 신경망 트랜잭션*, **5**(2), 157-166.

벤지오, Y., 듀샤르메, R., 빈센트, P., 잔빈, C. (2003). 신경 확률론적 언어 모델.
J. Mach. Learn. Res., **3**, 1137-1155.

Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D. 및 Bengio, Y. (2010). Theano: CPU 및 GPU 수학 표현식 컴파일러. *과학 컴퓨팅을위한 파이썬 컨퍼런스 (SciPy) 절차*. 구두 발표.

Boulanger-Lewandowski, N., Bengio, Y., Vincent, P. (2013). 반복 신경망을 사용한 오디오 코드 인식. In *ISMIR*.

Cho, K., van Merrienboer, B., Gulcehre, C., Bougares, F., Schwenk, H., and Bengio, Y. (2014a). 통계적 기계 번역을 위해 RNN 인코더-디코더를 사용한 구문 표현 학습. *자연어 처리의 경험적 방법(EMNLP 2014)* 논문집에 게재되었습니다.

조, K., 반 메리엔보어, B., 바흐다나우, D., 벤지오, Y. (2014b). 신경 기계 번역의 특성에 대해: 인코더-디코더 접근법. *통계 번역의 구문, 의미 및 구조에 관한 여덟 번째 워크숍에서*.

Devlin, J., Zbib, R., Huang, Z., Lamar, T., Schwartz, R. 및 Makhoul, J. (2014). 통계적 기계 번역을 위한 빠르고 강력한 신경망 공동 모델. *전산 언어학 협회*.

Forcada, M. L. and N˜ eco, R. P. (1997). 번역을위한 재귀 적 이종 연관 기억. In
J. Mira, R. Moreno-D´ıaz, 및 J. Cabestany, 편집자, *생물학적 및 인공 계산: 신경 과학에서 기술까지*, *컴퓨터 과학 강의 노트* 1240 권, 453-462 페이지. 스프링거 베를린 하이델베르크.

Goodfellow, I., Warde-Farley, D., Mirza, M., Courville, A. 및 Bengio, Y. (2013). Maxout networks. *제30회 국제 기계 학습 컨퍼런스 논문집*, 1319-1327 페이지.

그레이브스, A. (2012). 순환 신경망을 사용한 시퀀스 변환. *제 29 회 국제 기계 학습 컨퍼런스 (ICML 2012) 절차*.

그레이브스, A. (2013). 반복 신경망으로 시퀀스 생성. *arXiv:*1308.0850 [cs.NE].

그레이브스, A., 자이틀리, N., 모하메드, A.-R. (2013). 딥 바이렉셔널 LSTM을 사용한 하이브리드 음성 인식. *자동 음성 인식 및 이해(ASRU), 2013 IEEE 워크샵*, 273-278페이지.

Hermann, K. and Blunsom, P. (2014). 단어 정렬이 없는 다국어 분산 표현. *제2회 학습 표현에 관한 국제 컨퍼런스(ICLR 2014) 논문집*.

Hochreiter, S. (1991). 동적 신경망에 대한 연구. 디플로마 학위 논문, 정보학 연구소, 브라우어 교수, 뮌헨 공과대학교.

Hochreiter, S. 및 Schmidhuber, J. (1997). 장단기 기억. *신경 계산*, **9**(8), 1735-1780.

Kalchbrenner, N. & Blunsom, P. (2013). 반복적 연속 번역 모델. *자연어 처리의 경험적 방법에 관한 ACL 컨퍼런스(EMNLP) 논문집*, 1700-1709페이지. 전산 언어학 협회.

Koehn, P. (2010). *통계적 기계 번역*. 캠브리지 대학 출판부, 뉴욕, 뉴욕, 미국.

Koehn, P., Och, F. J., & Marcu, D. (2003). 통계적 구문 기반 번역. *2003년 인간 언어 기술에 관한 전산 언어학 협회 북미 지부 컨퍼런스 논문집 - 1권*, NAACL '03, 48-54쪽, 미국 펜실베이니아 주 스트로즈버그. 미국 전산 언어학 협회.

파스카누, R., 미콜로프, T., 벤지오, Y. (2013a). 순환 신경망 훈련의 어려움에 대해. In *ICML'2013*.

파스카누, R., 미콜로프, T., 벤지오, Y. (2013b). 순환 신경망 훈련의 어려움에 대해. *제30회 국제 기계 학습 컨퍼런스(ICML 2013) 논문집*.

파스카누, R., 굴체레, C., 조, K., 벤지오, Y. (2014). 심층 순환 신경망을 구축하는 방법. *제2회 학습 표현에 관한 국제 컨퍼런스(ICLR 2014)* 논문집.

Pouget-Abadie, J., Bahdanau, D., van Merrie¨nboer, B., Cho, K., and Bengio, Y. (2014). 자동 분할을 이용한 신경 기계 번역에서 문장 길이의 저주 극복. *통계 번역의 구문, 의미 및 구조에 관한 제8회 워크숍에서*.

슈스터, M. 과 팔리 왈, K. K. (1997). 양방향 순환 신경망. *신호 처리, IEEE 트랜잭션 온*, **45**(11), 2673-2681.

Schwenk, H. (2012). 구문 기반 통계적 기계 번역을위한 연속 공간 번역 모델. M. Kay 및 C. Boitet, 편집자, *제 24 회 국제 전산 언어학 컨퍼런스 (COLIN) 논문집*, 1071-1080 페이지. 인도 공과 대학 봄베이.

Schwenk, H., Dchelotte, D., & Gauvain, J.-L. (2006). 통계적 기계 번역을 위한 연속 공간 언어 모델. *메인 컨퍼런스 포스터 세션에 관한 COLING/ACL 논문집,* 723-730 페이지. 전산 언어학 협회.

Sutskever, I., Vinyals, O., Le, Q. (2014). 신경망을 사용한 시퀀스 투 시퀀스 학습. *신경 정보 처리 시스템의 발전 (NIPS 2014)*에서.

Zeiler, M. D. (2012). ADADELTA : 적응 형 학습 속도 방법. *arXiv:*1212.5701 [cs.LG].

A 모델 아키텍처

A.1 아키텍처 선택

섹션 3에서 제안한 방식은 순환 신경망(RNN)의 활성화 함수 f와 정렬 모델 a를 자유롭게 정의할 수 있는 일반적인 프레임워크입니다. 여기서는 이 백서의 실험을 위해 선택한 방식을 설명합니다.

A.1.1 순환 신경망

RNN의 활성화 함수 f는 최근 Cho 등(2014a)이 제안한 게이트 숨겨진 유닛을 사용합니다. 게이트 숨겨진 유닛은 원소 단위 탄과 같은 기존의 *단순* 유닛에 대한 대안입니다. 이 게이트 유닛은 Hochreiter와 Schmidhuber(1997)가 일찍이 제안한 장단기 메모리(LSTM) 유닛과 유사하며, 장기 종속성을 더 잘 모델링하고 학습할 수 있다는 공통점이 있습니다. 이는 미분의 곱이 1에 가까운 펼쳐진 RNN의 계산 경로를 가짐으로써 가능합니다. 이러한 경로를 통해 소실 효과의 영향을 크게 받지 않고 기울기가 쉽게 역방향으로 흐를 수 있습니다(Hochreiter, 1991; Bengio *et al*., 1994); 파스카누 *외*., 2013a). 따라서 비슷한 맥락에서 Sutskever 등(2014)이 수행한 것처럼 여기에 설명된 게이트 히든 유닛 대신 LSTM 유닛을 사용할 수 있습니다.

n*개의* 게이트 숨겨진 유닛을 사용하는 RNN의 새로운 상태$_i$는[8] 은 다음과 같이 계산됩니다.

$$s_i = f(s_{i-1}, y_{i-1}, c_i) = (1 - z_i) \circ s_{i-1} + z_i \circ \tilde{s}_i,$$

여기서 $\circ$는 요소별 곱셈이고, z_i는 업데이트 게이트의 출력입니다(아래 참조). 제안된 업데이트 상태 $\tilde{s}_i$는 다음과 같이 계산됩니다.

$$\tilde{s}_i = \tanh(We(y_{i-1}) + U[r_i \circ s_{i-1}] + Cc_i),$$

여기서 $e(y_{i-1}) \in \mathbb{R}^m$은 단어 y_{i-1}의 m 차원 임베딩이고, r_i은 리셋 게이트의 출력입니다(아래 참조). y_i가 *1-of-K* 벡터로 표현될 때, $e(y_i)$는 단순히 임베딩 행렬 $E \in \mathbb{R}^{m \times K}$의 열입니다. 방정식을 덜 복잡하게 만들기 위해 가능한 한 바이어스 항을 생략합니다.

업데이트 게이트 z_i는 각 숨겨진 유닛이 이전 활성화를 유지하도록 허용하고, 리셋 게이트 r_i는 이전 상태의 정보를 얼마나 많이 그리고 어떤 정보를 리셋해야 하는지 제어합니다. 다음과 같이 계산합니다.

$$z_i = \sigma(W_z e(y_{i-1}) + Us_{z\,i-1} + C c_{z\,i}), \quad r_i = \sigma(W_r e(y_{i-1}) + Us_{r\,i-1} + C c_{r\,i}),$$

여기서 σ(-)는 로지스틱 시그모이드 함수입니다.

디코더의 각 단계에서 출력 확률(방정식 (4))을 다층 함수로 계산합니다(Pascanu *et al*., 2014). 최대 출력 단위의 단일 숨겨진 계층을 사용하고(Goodfellow *et al*., 2013) 소프트맥스 함수로 출력 확률(각 단어당 하나씩)을 정규화합니다(방정식 (6) 참조).

A.1.2 정렬 모델

정렬 모델은 길이 T_x와 T_y의 각 문장 쌍에 대해 $T_x \times T_y$번 평가해야 한다는 점을 고려하여 설계해야 합니다. 계산을 줄이기 위해 다음과 같이 단일 계층 다층 퍼셉트론을 사용합니다.

$$a(s_{i-1}, h_j) = v_a^\top \tanh(Ws_{a\,i-1} + Uh_{aj}),$$

여기서 $W_a \in \mathbb{R}^{n \times n}$, $U_a \in \mathbb{R}^{n \times 2n}$, $v_a \in \mathbb{R}^n$ 는 가중치 행렬입니다. Uh_{aj} 는 i에 의존하지 않으므로 미리 계산하여 계산 비용을 최소화할 수 있습니다.

[8] 여기에서는 디코더의 공식을 보여줍니다. 컨텍스트 벡터 ci와 관련 용어를 무시하면 인코더에서도 동일한 공식을 사용할 수 있습니다.

A.2 모델에 대한 자세한 설명

A.2.1 인코더

이 섹션에서는 실험에 사용된 제안된 모델(RNNsearch)의 아키텍처에 대해 자세히 설명합니다(4-5절 참조). 여기서는 가독성을 높이기 위해 편향 용어는 모두 생략합니다.

이 모델은 1-of-K 코딩된 단어 벡터의 소스 문장을 입력으로 받습니다.

$$\mathbf{x} = (x_1, \ldots, x_{T_x}), x_i \in \mathbb{R}^{K_x}$$

를 호출하고 1-of-K 코딩된 단어 벡터로 번역된 문장을 출력합니다.

$$\mathbf{y} = (y_1, \ldots, y_{T_y}), y_i \in \mathbb{R}^{K_y},$$

여기서 K_x 및 K_y 는 각각 소스 언어와 대상 언어의 어휘 크기입니다. T_x 및 T_y 는 각각 소스 문장과 대상 문장의 길이를 나타냅니다.

먼저, 양방향 순환 신경망(BiRNN)의 순방향 상태를 계산합니다:

$$\overrightarrow{h}_i = \begin{cases} (1 - z_i) \circ h_{i-1} + z_i \circ \overrightarrow{h}_i & , \text{if } i > 0 \\ 0 & , i = 0\text{이면} \end{cases}$$

어디

$$\overrightarrow{h}_i = \tanh\left(\overrightarrow{W}Ex_i + \overrightarrow{U}\left[r_i \circ \overrightarrow{h}_{i-1}\right]\right)$$

$$\overrightarrow{z}_i = \sigma\left(\overrightarrow{W}_z Ex_i + \overrightarrow{U}_z \overrightarrow{h}_{i-1}\right)$$

$$\overrightarrow{r}_i = \sigma\left(\overrightarrow{W}_r Ex_i + \overrightarrow{U}_r \overrightarrow{h}_{i-1}\right).$$

$E \in \mathbb{R}^{m \times K_x}$ x는 단어 임베딩 행렬입니다. $\overrightarrow{W}, \overrightarrow{W}_z, \overrightarrow{W}_r \in \mathbb{R}^{n \times m}$, $\overrightarrow{U}, \overrightarrow{U}_z, \overrightarrow{U}_r \in \mathbb{R}^{n \times n}$ 는 가중치 행렬입니다. m과 n은 각각 워드 임베딩 차원과 숨겨진 단위 수입니다. σ(-)는 평소와 같이 로지스틱 시그모이드 함수입니다.

역방향 상태 $(\overleftarrow{h}_1, \ldots, \overleftarrow{h}_{T_x})$도 유사하게 계산됩니다. 임베딩 행렬이라는 단어를 공유합니다. E를 가중치 행렬과 달리 순방향과 역방향 RNN 사이에 설정합니다.

앞으로 상태와 뒤로 상태를 연결하여 주석을 얻습니다($h_1, h_2, \ldots, h_{T_x}$).

$$h_i = \begin{bmatrix} \overrightarrow{h}_i \\ \overleftarrow{h}_i \end{bmatrix} \tag{7}$$

A.2.2 디코더

인코더의 어노테이션이 주어진 디코더의 숨겨진 상태 s_i 는 다음과 같이 계산됩니다.

$$s_i = (1 - z_i) \circ s_{i-1} + z_i \circ \tilde{s}_i,$$

어디

$$\tilde{s}_i = \tanh\left(WEy_{i-1} + U\left[r_i \circ s_{i-1}\right] + Cc_i\right)$$

$$z_i = \sigma\left(W_z Ey_{i-1} + U_z s_{i-1} + C_z c_i\right)$$

$$r_i = \sigma\left(W_r Ey_{i-1} + U_r s_{i-1} + C_r c_i\right)$$

E는 대상 언어에 대한 단어 임베딩 행렬입니다. $W, W_z, W_r \in \mathbb{R}^{n \times m}$, $U, U_z, U_r \in \mathbb{R}^{n \times n}$, 그리고 $C, C_z, C_r \in \mathbb{R}^{n \times 2n}$ 은 가중치입니다. 다시 말하지만, m과 n은 단어 임베딩 차원입니다. and the number of hidden units, respectively. The initial hidden state s_0 is computed by s_0 =

tanh $W_s \overleftarrow{h}_1$ 여기서 $W_s \in \mathrm{R}^{n \times n}$.

컨텍스트 벡터 c_i 는 정렬 모델에 의해 각 단계에서 다시 계산됩니다:

$$c_i = \sum_{j=1}^{T_x} \alpha h_{ijj},$$

모델	업데이트 (×10)5	Epochs	시간	GPU	기차 NLL	Dev. NLL
RNNenc-30	8.46	6.4	109	타이탄 블랙	28.1	53.0
RNNenc-50	6.00	4.5	108	Quadro K-6000	44.0	43.6
RNNsearch-30	4.71	3.6	113	타이탄 블랙	26.7	47.2
RNNsearch-50	2.88	2.2	111	Quadro K-6000	40.7	38.1
RNNsearch-50⋆	6.67	5.0	252	Quadro K-6000	36.7	35.2

표 2: 학습 통계 및 관련 정보. 각 업데이트는 단일 미니배치를 사용하여 매개변수를 한 번 업데이트하는 것에 해당합니다. 한 에포크는 훈련 집합을 한 번 통과하는 것입니다. NLL은 훈련 세트 또는 개발 세트에 있는 문장의 평균 조건부 로그 확률입니다. 문장의 길이가 다르다는 점에 유의하세요.

어디

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}$$

$$e_{ij} = v_a^\top \tanh(W s_{i-1} + U h_j)$$

및 h_j , $v_a \in \mathbb{R}^n$, $W_a \in \mathbb{R}^{n' \times n}$ 및 는 소스 문장의 j 번째 주석입니다(방정식 (7) 참조). $U_a \in \mathbb{R}^{n' \times 2n}$ 은 가중치 행렬입니다. 모델은 RNN 인코더-디코더가 됩니다(Cho *et al.*, 2014a), c_i 를 h_{T_x} 로 수정하면 됩니다.

디코더 상태 s_{i-1} , 컨텍스트 c_i 및 마지막으로 생성된 단어 y_{i-1} 를 사용하여 대상 단어 y_i 의 확률을 다음과 같이 정의합니다.

$$p(y_i \mid s_i, y_{i-1}, c_i) \propto \exp(y_i W t_i)$$

어디

$$t_i = \left[\max\{ t'_{i,2j-1}, t'_{i,2j} \} \right]_{j=1,\dots,l}^\top$$

는 벡터 t'_i 의 k 번째 원소이며 다음과 같이 계산됩니다.

$$t'_i = U_o s_{i-1} + V_o E y_{i-1} + C_o c_i .$$

$W_o \in \mathbb{R}^{k \times l}$, $U_o \in \mathbb{R}^{2l \times n}$, $V_o \in \mathbb{R}^{2l \times m}$ 및 $C_o \in \mathbb{R}^{2l \times 2n}$ 은 가중치 행렬입니다. 이는 단일 최대 출력 숨겨진 레이어(Goodfellow *et al.*, 2013)로 딥 출력(Pascanu *et al.*, 2014)을 갖는 것으로 이해될 수 있습니다.

A.2.3 모델 크기

이 논문에서 사용된 모든 모델에서 숨겨진 레이어 n 의 크기는 1000, 단어 내포 차원 m은 620, 심층 출력에서 최대 출력 숨겨진 레이어 l 의 크기는 500입니다. 정렬 모델 n' 의 숨겨진 단위 수는 1000입니다.

B 교육 절차

B.1 매개변수 초기화

반복 가중치 행렬을 초기화했습니다. $\overleftarrow{U}$, $\overleftarrow{U_z}$, $\overleftarrow{U_r}$, $\overrightarrow{U}$, $\overrightarrow{U_z}$, $\overrightarrow{U_r}$, U, U_z 및 U_r 무작위로 또는-행렬을 사용합니다. W_a 및 U_a 의 경우 평균 0, 분산 0.001^2 의 가우스 분포에서 각 요소를 샘플링하여 초기화했습니다. V_a 의 모든 요소와 모든 바이어스 벡터는 0으로 초기화되었습니다. 다른 모든

가중치 행렬은 평균 0, 분산 0.01의 가우스 분포에서 샘플링하여 초기화했습니다[2] .

B.2 교육

확률적 경사 하강(SGD) 알고리즘을 사용했습니다. 각 파라미터의 학습 속도를 자동으로 조정하기 위해 Adadelta(Zeiler, 2012)를 사용했습니다($\epsilon = 10^{-6}$ 및 $\rho = 0.95$). 우리는 명시적으로

비용 함수의 기울기의 L_2 -규범이 임계값보다 클 경우, 매번 미리 정의된 임계값인 최대 1이 되도록 정규화했습니다(Pascanu *et al.*, 2013b). 각 SGD 업데이트 방향은 80개의 문장으로 구성된 미니 배치로 계산되었습니다.

업데이트할 때마다 미니배치에서 가장 긴 문장의 길이에 비례하는 시간이 필요합니다. 따라서 계산 낭비를 최소화하기 위해 매 20번째 업데이트 전에 1600개의 문장 쌍을 검색하고 길이에 따라 정렬한 후 20개의 미니배치로 나누었습니다. 훈련 데이터는 훈련 전에 한 번 셔플하고 이러한 방식으로 순차적으로 트래버스했습니다.

표 2에는 실험에 사용된 모든 모델 훈련과 관련된 통계가 나와 있습니다.

C 긴 문장 번역

출처	입원 권한은 의사가 환자를 병원이나 의료 센터에 입원시킬 수 있는 권리입니다. 병원에서 의료 종사자로서의 신분에 따라 진단 또는 시술을 수행하도록 요청할 수 있습니다.
참조	입장 권한은 회원 자격에 따른 회원의 권리입니다. 진단이나 치료를 위해 병원이나 의료 센터에 환자를 입원시키는 것을 말합니다.
RNNenc-50	입원 권한은 환자가 병원이나 의료진에게 환자를 진료할 수 있는 권리입니다. 진단을 받거나 건강 상태에 따라 진단을 받을 수 있습니다.
RNNsearch-50	입원 특권이란 환자를 병원이나 병원에 입원시킬 수 있는 의사 결정권입니다. 병원 의료진 상태에 따라 진단 또는 치료 과정을 수행하기 위한 의료 센터입니다.
Google 번역	입원 권한은 병원이나 병원에서 환자를 입원시킬 수 있는 의사의 권리입니다. 병원에서 의료진으로서 자신의 상황에 따라 진단 또는 치료 과정을 수행하기 위해 의료 센터를 방문합니다.

출처	이러한 경험은 디즈니가 "시리즈의 수명을 연장하고 점점 더 중요해지고 있는 디지털 플랫폼을 통해 청중과 새로운 관계를 맺을 수 있습니다."라고 덧붙였습니다.
참조	이 유형의 경험은 디즈니의 노력에 대한 간부들 사이에서 다음과 같은 두 가지 유형의 경험을 제공합니다. 자신의 삶을 살고 대중과 새로운 관계를 구축하는 것은 '더 중요한 플랫폼'이 될 것"이라고 덧붙였습니다.
RNNenc-50	이 유형의 경험은 디즈니의 "수명 연장"을 위한 이니셔티브의 일부입니다. 더 많은 복합적인 학습자들과 소통하고 관계를 발전시킬 수 있습니다.
RNNsearch-50	이 장르의 경험은 디즈니의 '수명 연장'을 위한 노력의 일부입니다. 더 중요한 플랫폼을 통해 대중과 새로운 관계를 형성하고 창출할 수 있습니다."라고 덧붙였습니다.
Google 번역	이 장르의 경험은 디즈니가 가족의 삶의 질을 향상시키기 위해 기울인 노력의 일부입니다. CONSTRUIR 드 누벨 드 대중과 새로운 관계를 구축하기 위해 더 중요하게 변화하는 새로운 형식의 플레이트가 필요합니다"라고 덧붙였습니다.

출처	목요일 기자 회견에서 블레어 총리는 이 동영상에는 다음과 같은 내용이 없다고 밝혔습니다. 시장에게 형사 고발을 당할 수 있는 '합당한 동기'에 해당합니다.
참조	기자 회견에서 블레어 전 총리는 이 영상에 출연하지 않았다고 밝혔습니다. "합리적인 이유"를 구성하여 범죄 혐의의 근거가 될 수 있습니다.
RNNenc-50	기자 회견에서 블레어 총리는 다음과 같이 말했습니다. 는 범죄 혐의가 적용될 수 있는 '합당한 동기'가 될 수 있습니다.
RNNsearch-50	오늘 기자 회견에서 블레어 전 총리는 이 동영상에 출연하지 않았다고 밝혔습니다. '범죄 혐의'로 이어질 수 있는 '합리적인 모티브'를 구성할 수 있습니다.
Google 번역	오늘 기자 회견에서 블레어 전 총리는 이 영상에 출연하지 않았다고 밝혔습니다. 이는 '합리적 이유'를 구성할 수 있는 '모티브'가 되어 범죄 혐의가 적용될 수 있습니다.

표 3: 테스트 세트에서 선택한 긴 소스 문장(30단어 이상)에서 RNNenc-50과 RNNsearch-50이 생

성한 번역. 각 소스 문장에 대해 골드 스탠다드 번역도 표시되어 있습니다. Google 번역을 통한 번역은 2014년 8월 27일에 이루어졌습니다.