

A cluster of overlapping, semi-transparent geometric shapes in shades of blue, green, and red, located in the top-left corner of the slide.

Seq2Seq

Sequence to Sequence Learning with Neural Networks

<https://arxiv.org/abs/1409.3215>

부스트캠프 AI tech 5기 NLP 김태민

A cluster of overlapping, semi-transparent geometric shapes in shades of light gray, located in the bottom-right corner of the slide.

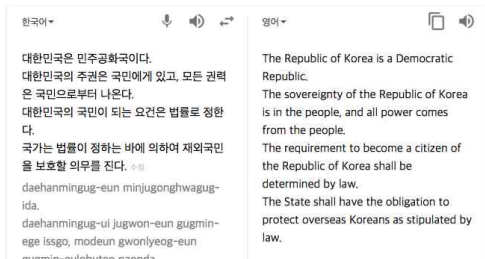
A cluster of overlapping triangles in shades of blue, green, and red in the top-left corner, and a cluster of overlapping triangles in shades of grey in the bottom-left corner.

목차

- Machine translation
- Limitations of existing studies
- Encoder-Decoder structure
- Encoder
- Decoder
- Structure
- Learning techniques
- Experiments
- Question

Seq2Seq

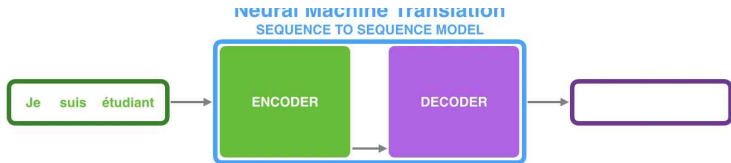
Machine translation



The screenshot shows a web-based machine translation interface. On the left, under the '한국어' (Korean) tab, there is a text input area containing the following Korean text: '대한민국은 민주공화국이다. 대한민국의 주권은 국민에게 있고, 모든 권력은 국민으로부터 나온다. 대한민국의 국민이 되는 요건은 법률로 정한다. 국가는 법률이 정하는 바에 의하여 재외국민을 보호할 의무를 진다.' Below this text is a phonetic transcription: 'daehanmingug-eun minjugonghwagug-ida. daehanmingug-ui jugwon-eun gugmin-ege issgo, modeun gwonlyeog-eun gumin-eulhutan eando.' On the right, under the '영어' (English) tab, the translated text is displayed: 'The Republic of Korea is a Democratic Republic. The sovereignty of the Republic of Korea is in the people, and all power comes from the people. The requirement to become a citizen of the Republic of Korea shall be determined by law. The State shall have the obligation to protect overseas Koreans as stipulated by law.'

- 일반적으로 우리가 흔히 쓰는 번역기등이 기계번역이다.
- 이때 기존의 통계적 방식으로 번역을 했다면 딥러닝이 발전하면서 기계번역 또한 딥러닝 방식으로 사용된다.
- 이 논문에서는 English to French 로 기계번역을 수행하였다.

Endoer-Decoder structure




- 각 단어가 시간 순으로 입력되며 ENCODER에 들어간다. 이를 하나의 CONTEXT로 압축하여 디코더로 보낸다.
- 디코더에서는 CONTEXT를 받아 이 정보를 토대로 시간 마다 하나의 단어를 출력시킨다. (EOS 토큰이 나오면 즉시 중단)
- 이때 일반적인 RNN과 다르게 입력과 출력의 차원은 동일하지않아도 된다.

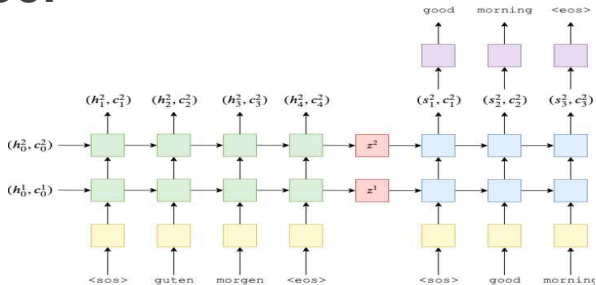


Seq2Seq

Limitations of existing studies

- 기존의 DNN 방식은 대부분 입력과 대상이 고정된 차원의 벡터로 인코딩 될수 있는 문제에만 주로 적용했다.
 - RNN과 LSTM등 여러 시퀀스 데이터를 처리하는 방식이 도입되었지만 각 언어 별로 단어의 순서등의 문제때문에 구문을 이해하는데는 상당히 어려웠다.
 - 기존의 Connectionist Sequence Classification의 방법은 입력과 출력간의 monotonic한 정렬이 있다는것을 가정한다. 즉 입력 시퀀스와 출력 시퀀스 간에 일치하는 시점이 존재하여 각 토큰은 출력 토큰의 어떠한 해당 요소와 일치한다는 가정이다. 하지만 이러한 가정이 일부 적용되지 않는 부분이 존재
- 

Endoer



- Encoder에서는 각 임베딩된 단어 벡터들을 입력으로 받아 LSTM을 통과 시킨다. 이때 LSTM에서는 가변 길이의 문장을 고정된 CONTEXT 벡터로 mapping하는 방법을 학습이 된다.

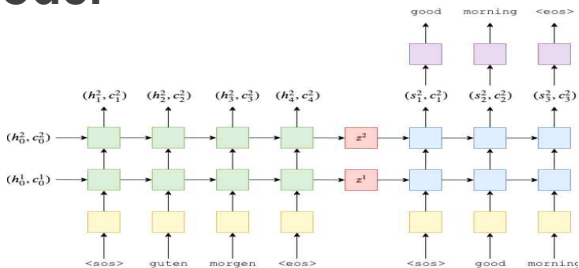
- CONTEXT 벡터로 압축이 되면 이는 순서에 대한 문맥적인 정보 또한 담아내고 있다.

- 기본적으로 Encoder에는 2-layers의 LSTM이 사용된다.

<https://github.com/bentrevett>

- 2-layers 사용 시 2개의 CONTEXT 벡터가 위와 같이 적용된다. /pytorch-seq2seq

Decoder



- Decoder 또한 2-layers 로 구성되며 첫 time step에는 $\langle \text{sos} \rangle$ 토큰과 CONTEXT 벡터인 z^1 을 입력으로 받아 2번째 LSTM과 1번째 LSTM 보내며 2번째 LSTM은 총 2번째 CONTEXT 벡터 그리고 1번째 LSTM에 나온 출력값을 입력으로 받아 첫번째 단어를 출력시킨다.

- 2번째 time step에서는 첫번째 단어와 출력값과 첫 time step에서의 hidden state를 입력으로 받아 다시 2번째 LSTM에 전달해주고 2번째 LSTM에서 다시 첫 time step의 2번째 LSTM의 hidden state의 입력과 1번째 LSTM의 출력 값을 받아 2번째 token을 출력해주게 된다.

-이를 주기적으로 반복하여 $\langle \text{eos} \rangle$ 토큰이 나오면 종료시킨다. <https://github.com/bentrevett/pytorch-seq2seq>



Structure


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

- 위 수식에서 결국

조건부 확률 x 가 주어졌을때 y 를 최대화 하는 것인데 이를 다시 쓰면 조건부 확률 $x_{\{1\}} \sim x_{\{t\}}$ 가 주어 졌을때 이를 v 로 압축하고 t 스텝마다 $y_{\{t\}}$ 조건부 확률을 계산(최대화)하는것이라고 볼 수 있다.

- 최종적인 Seq2Seq은 위 수식을 추정하는 것이다. 이때 위 수식의 분포에서 $y_{\{t\}}$ 가 나올 확률은 전체 어휘 집합에서 softmax한 값이다.

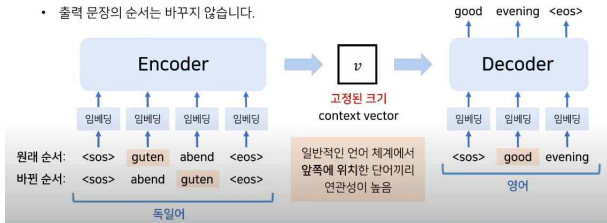
- 이때 $\langle \text{eos} \rangle$ 토큰을 통하여 우리는 가능한 모든 길이의 시퀀스에 대한 분포 또한 정의 할 수 있다.



Learning techniques

Reversing the Source Sentences

- 출력 문장의 순서는 바꾸지 않습니다.



- 문장의 순서를 뒤집어서 입력에 집어넣는다. 논문에서는 LSTM에서 정답 레이블은 그대로 두고 입력의 순서를 바꾸면 훨씬 더 잘 학습한다고 나와있다. 결과적으로는 문장을 뒤집을시 better memory utilization 한 LSTM이 생성된다고 나와있다.

- 이유로는 언어 체계에서 '나는' 과 'I' 는 비슷한 벡터상의 위치에 표시 되며 서로 앞쪽에 연관될 확률이 높다. 하지만 LSTM과 RNN같은 시퀀셜한 모델은 맨처음 단어의 정보량이 time step마다 조금씩 소실됨으로써 context vector를 decoder에 넣었을때 decoder는 처음 나온 단어가 입력으로 들어감으로써 처음 나온 단어의 중요성이 높아진다. 하지만 처음 단어의 정보량은 적어 잘못 예측확률이 높지만 문장의 순서를 바꾸게 되면 '나는'의 정보량이 문장의 순서를 안바꾼것 보다 높아져 디코더의 첫 단어의 예측 확률이 올라가는것 이라고 생각된다.

Experiments

Dataset

United Nations Parallel Corpus

	French ↔ English	
Sentences	12,886,831	
Words	411,916,781	360,341,450
Distinct words	565,553	666,077


- wmt14에서 위의 데이터셋을 쓴 것 같다. sentences의 길이가 일치 (논문에서 명확히 설명하지 않음)

-여기서 실제 학습은 프랑스,영어 단어 각각 3억4800만개와 sentences 12000만 개로 학습을 진행하였다.
이때 학습을 원활하게 진행하기 위해 소스 언어에서 빈도수가 높은 단어 16만개 target 언어 에서 빈도수가 높은 단어 8만개로 어휘 사전을 만들었다. 이때 출력 및 입력 시 어휘 사전에 없는 단어는 UNK 토큰으로 대체 되었다.



Experiments

Training details

- 실제 훈련시 각 레이어 별로 1000개의 셀과 임베딩 벡터는 1000차원으로 진행한다.
 - 모든 파라미터를 -0.08~0.08 사이의 uniform distribution로 초기화 한다.
 - SGD를 사용했으며 $Lr = 0.7$, 5 epoch 이후 0.5 epoch마다 Lr 을 절반으로 감소 시킨다. 총 7.5 epoch 훈련
 - `batch_size = 128`이다.
 - LSTM에서 gradients exploding을 방지하기 위해 그레디언트가 임계값을 초과할시 스케일링을 진행
 - `mini_batch`를 만들때 최대한 모든 문장의 길이가 유사하도록 만들어 학습 속도를 2배 가속화
 - 8개의 GPU를 사용하여 모델을 병렬화 LSTM각 레이어 별로 서로 다른 GPU를 사용하였다. 4개는 LSTM 4개는 softmax를 병렬화 하는데 사용했으며 학습 시간은 총 10일 소요됐다.
- 

Experiments

BLEU SCORE

$$BLEU = \min(1, \frac{\text{output length}(\text{예측 문장})}{\text{reference length}(\text{정답 문장})}) (\prod_{i=1}^4 \text{precision}_i)^{\frac{1}{4}}$$

- 점수의 측정 방법은 BLEU를 사용하였다. 기계 번역결과와 사람이 번역한 결과의 유사도를 측정하여 번역 성능을 측정하는 지표이다.

- 이때 뒤에 있는 $(\prod_{i=1}^4 \text{precision}_i)^{\frac{1}{4}}$ 는 정답 문장과 예측 문장 사이에 n-gram(1~4)가 겹치는 정도의 기하 평균이라고 나와있다.

Experiments

Experiments

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

- 실험결과로 baseline만 사용했을 시 33.30 이라는 점수를 얻었으며 앙상블과 beam size를 늘렸을 시 최종적으로 34.81의 결과를 얻어냈다.

- 통계적 기반인 SMT의 시스템과 함께 사용시 36.5라는 점수를 기록했으며 최고 36.5를 얻어냈으며 당시 최고 점수인 SOTA 37.0보다 0.5점 낮은 점수를 기록 하였다.



Question

Question

1. 앙상블에 대해 어떠한 다른 방식을 적용했는지에 대한 의문점
 2. 데이터 셋에 대한 정확한 출처?
 3. SMT시스템과 어떻게 함께 신경망을 사용했는지에 대한 의문
 4. CONTEXT vector를 추출하여 하는 방식 외의 순수 RNN,LSTM으로 번역 테스트를 진행한 방식
- 