1. Introduction

- Neural Machine Translation이 떠오르고 있음 (<-> Phrase-based translation system)
- 제안된 대부분의 NMT에 encoder-decoders 가 쓰임
  - 길이가 고정된 벡터에 문장을 encode 하는 방식이 쓰임
  - 문제 : 신경망이 고정된 길이의 벡터에 source sentence의 모든 정보를 압축해야 함.
    긴 문장을 처리할 때 성능이 떨어지며 특히 Training Set 보다 긴 문장을 처리할 때 더 심함
- 이를 해결하기 위해 가장 관련성 있는 정보의 위치를 찾는 모델을 제안
  - 문장의 모든 정보를 하나의 고정된 길이의 벡터에 저장하는 대신, 벡터들의 시퀀스로 인코딩하고 decode 할 때 벡터들의 부분집합을 적응적으로 선택한다.

2. Background : NMT

- 확률적 관점에서 번역은 주어진 문장 x가 주어졌을 때 조건부 확률이 최대화 되는 목표문장 y를 찾는 것
- 모델에 조건부 분포를 학습시킨 후, 문장이 주어지면 조건부 확률을 극대화하는 문장을 찾는다.
- 기존 RNN Encoder-Decoder 모델

3. Learning to Align and Translate

- Bi-directional RNN이 인코더와 디코더로 사용된다.

3-1. Decoder

- 기존의 인코더-디코더와는 달리 확률은 각각의 y에 대한 context 벡터 c에 의해 정해진다.
- Context vector c는 input 문장의 annotation에 의해 결정된다.
  - Annotation : 각각의 t번째 input을 넣어줬을 때 생성되는 forward, backward hidden state들을 concatenate 한 값
  - i번째 주위의 정보에 집중하기 때문에 정보를 잘 담게 된다.
- Context vector는 이 annotation들의 가중합으로 결정된다.
- 가중합에 사용되는 weight는 j번째 input 근처와 i번째 output 근처의 매치를 점수로 매긴 alignment model을 사용한다.

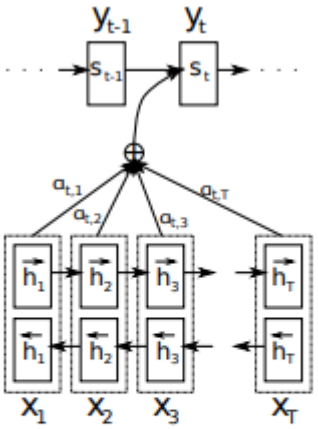


Figure 1: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence $(x_1, x_2, \ldots, x_T)$.

- 기존의 기계 번역과는 달리 alignment model은 비용 함수의 gradient를 backpropagate 할 수 있는 soft alignment를 계산한다. (즉, 학습될 수 있다.)
- expected annotation 은 previous state와 current annotation의 신경망을 통해 얻는다.

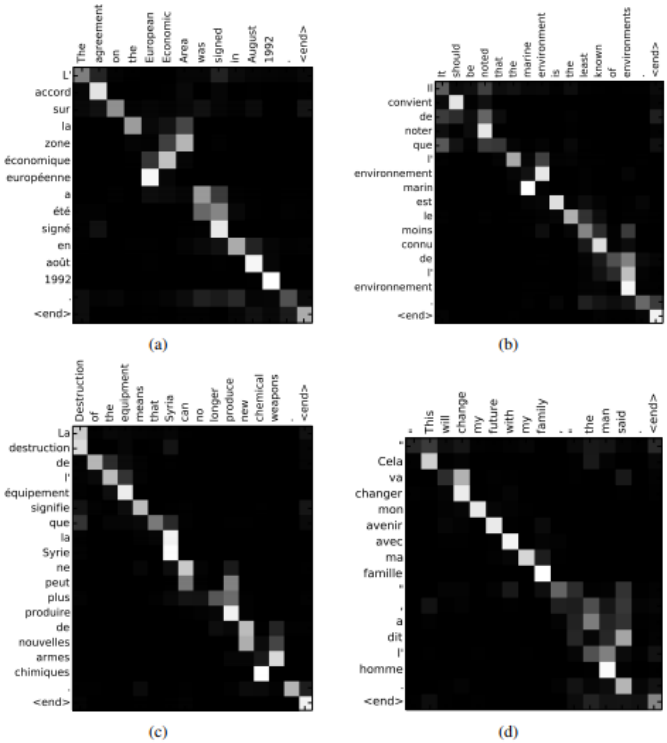
- encoder가 문장의 모든 정보를 고정된 길이의 벡터에 저장할 필요가 없어지며, 정보는 annotations에 퍼지게 되고 decoder는 이를 선택적으로 탐색할 수 있게 된다.

3-2. Encoder

- Bidirectional RNN이 사용된다.
  - annotation을 forward hidden state와 backward hidden state를 통해 얻는다.
  - 이를 통해 특정 input의 주변을 더 강하게 반영할 수 있게 된다.

4. Result

- 기존의 길이가 고정된 context vector를 사용한 모델은 길이가 길어지면 성능이 급격히 떨어지는 반면 attention을 사용한 모델은 길이가 긴 문장에 대해서도 성능이 떨어지지 않았다.
- 훨씬 많은 corpus를 이용해서 번역하는 Moses에 비해도 성능이 좋은데, 굉장한 성과다.



- 번역 문제에 있어서 한 언어의 단어가 다른 언어의 어떤 단어에 대해 매치되는지도 alignment model을 통해 학습하는데, 잘 된다.