AiDA Lab.
Artificial Intelligence & Data Analysis Laboratory

2021 인공지능 소수전공

91차시: BERT

2021.08.13 17:30~18:15

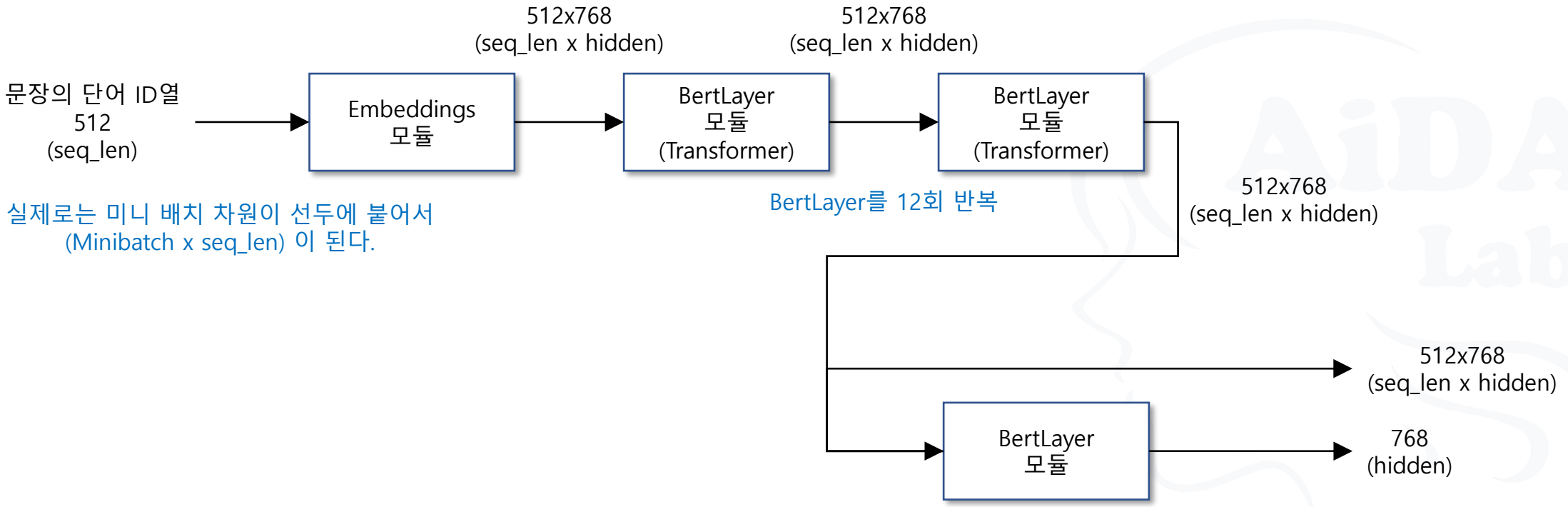
Seokhwan Yang

BERT

- BERT (Bidirectional Encoder Representation Transformers)
  - 2018년 하반기 Google이 발표한 자연어 처리를 위한 딥 러닝 모델
  - 일부 성능평가에서는 인간보다 높은 정확도를 보임
  - 영상인식 계열에 비하여 발전이 늦은 언어처리 딥 러닝의 한계를 돌파하는 계기가 될 것으로 기대 받음

BERT 모델

- **BERT-Base 모델 기준으로 학습**
- **BERT 모델 구조**



BERT 모델 구조

- BERT 모델 프로세스
  - 문장을 단어 ID로 하는 ID열(길이는 seq_len=512) → 입력 → Embeddings 모듈 전달

  - Embeddings 모듈은 ID 열을 단어의 특징량 벡터로 변환
    → 단어와 특징량 벡터의 위치 정보를 나타내는 Positional Embedding 추가
    : BERT-Base에서 사용하는 특징량 벡터의 차원 수는 768
      → (모델 구조도에서 hidden으로 표시)

BERT 모델 구조

- Embedding 모듈의 출력 텐서인 (seq_len x hidden) = 512 x 768 → BertLayer로 전달

- BertLayer 모듈: Self-Attention을 이용하여 특징량 변환 수행 → 모듈을 총 12회 반복: 출력 텐서 크기는 입력 텐서와 같은 512 x 768

- 12회 반복된 BertLayer 모듈의 출력 텐서(512 x 768(에서 첫 단어의 특징량 (1 x 768)을 → BertPooler 모듈에 입력

BERT 모델 구조

- **출력 텐서의 첫 단어를 [CLS]로 설정 → 문장의 클래스 분류 등에 사용하기 위한 입력 문장 전체의 특징량을 가지는 부분으로 활용**

- **선두 단어의 특징량을 BertPooler 모듈로 변환**

- **최종 출력 텐서 (2가지)**
  - 12회의 BertLayer 모듈에서 출력된 (seq_len x hidden)=(512x768) 텐서
  - 선두 단어 [CLS]의 특징량(BertPooler 모듈의 출력)인 크기 768의 텐서

BERT 모델의 사전 작업

- BERT는 네트워크 모델을 두 종류의 언어 작업으로 사전 학습함
  - MLM (Masked Language Model)
  - NSP (Next Sentence Prediction)

BERT 모델의 사전 작업

- MLM
  - CBOW 모델의 확장판 작업
  - CBOW 모델: 문장 중 한 단어를 마스크하여 알 수 없게 하고, 마스크 단어의 앞뒤(약 5단어씩) 정보로 마스크된 단어를 추정하는 모델
  - 입력의 512 단어 중 여러 단어를 마스크하여 마스크된 단어 앞뒤 몇 단어를 지정하지 않고 마스크되지 않은 단어 모두를 사용하여 마스크된 단어를 추정함으로써 해당 단어의 특징량 벡터를 획득하는 작업

BERT 모델의 사전 작업

- **NSP**
  - **BERT 모델은 사전 학습에서 두 개의 텍스트 데이터를 입력함 (512 단어로 두 문장 구성)**
  - **두 문장은 [SEP]로 구분되며 지도 데이터 내에서 두 개의 패턴으로 준비됨**
    - 연속적으로 존재하며 의미 있고 관계가 깊은 문장
    - 전혀 관계가 없고 문맥의 연결이 없는 두 문장
  - **BertPooler 모듈에서 출력된 선두 단어 [CLS]의 특징량으로 입력된 두 개의 문장이 어떤 패턴인지 추론**

BERT 모델의 사전 작업

- **BERT 모델의 두 종류의 사전 학습의 예**
  - **MLM**

    일부 단어가 마스크 된 문장을 입력하고 해당 단어가 Vocabulary의 어떤 낱말인지 맞히기

(문장) [CLS] I accessed the 마스크 account [SEP] We play soccer at the bank of the 마스크 [SEP]

답: bank

답: river

  - **NSP**

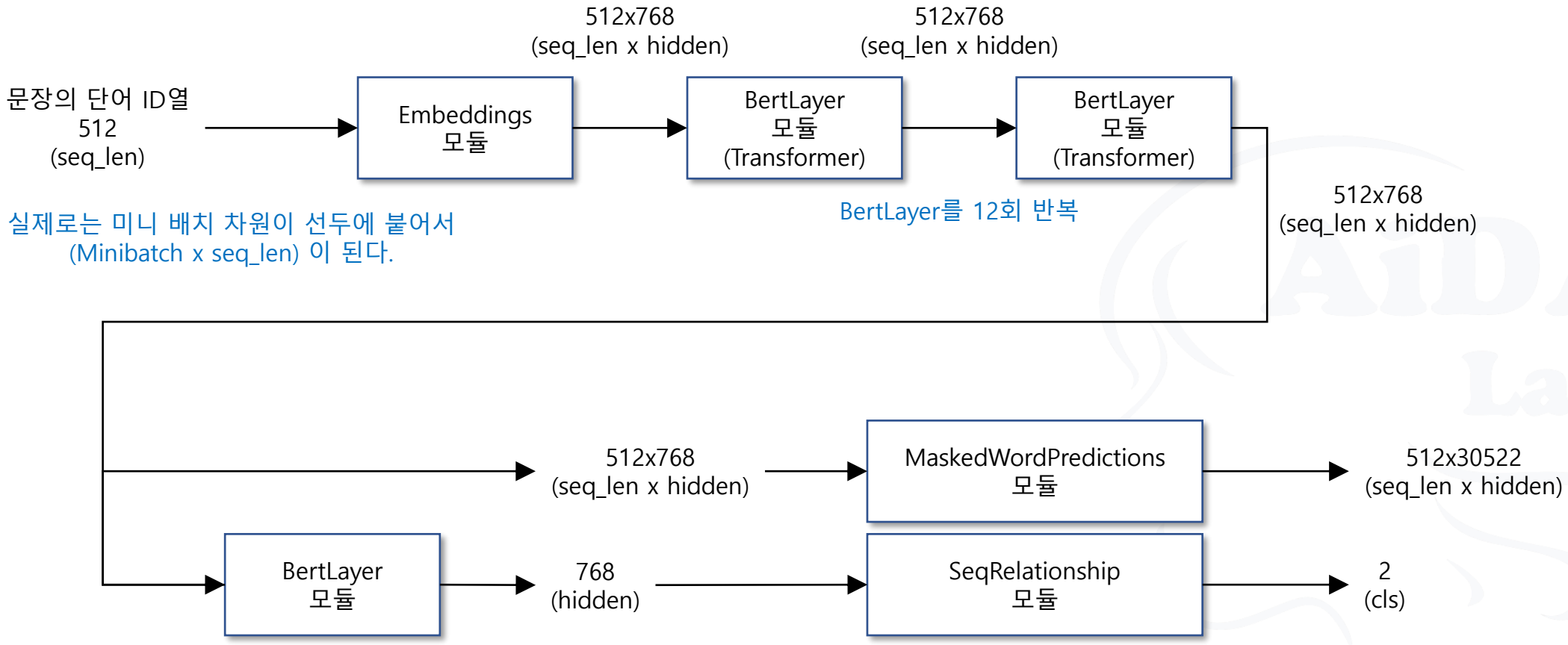
    두 문장을 입력하고 두 문장의 의미가 연결되어 있는지 맞히기

(문장) [CLS] I accessed the bank account [SEP] We play soccer at the bank of the river. [SEP]

답은 관계성이 없음

BERT 모델의 사전 작업

- 사전 학습을 실시하는 BERT 모델 구조



BERT 모델의 사전 작업

- 두 언어 작업을 해결하기 위하여 모듈 연결
  - 기본 모델에 MaskedWordPredictions 모듈과 SeqRelationship 모듈을 붙여 두 종류의 사전 작업인 MLM과 NSP를 잘 수행할 수 있도록 기본 모델 학습

BERT 모델의 사전 작업

- MaskedWordPredictions 모듈
  - BertLayer 출력(seq_len x hidden)=(512x768)을 입력하고 (seq_len x vocab_size) = (512x30,522) 출력
  - vocab_size (30,522)는 BERT의 vocabulary 전체의 단어 수(영어의 경우)
  - 입력된 512 단어가 전체 vocabulary 단어의 어느 것인지 (512x30522)에 대하여 소프트맥스 함수를 계산하여 도출
  - 실제로 추정하는 것은 입력 단어 512개 전체가 아닌 마스크된 알 수 없는 단어 뿐

BERT 모델의 사전 작업

- **SeqRelationship 모듈**
  - **BertPooler 모듈에서 출력된 선두 단어 [CLS]의 특징량 벡터를 전결합층에 입력하여 클래스의 수가 2인 분류를 실행**
  - **전 결합층의 출력 크기 2 → 아래의 2 패턴 중 어느 쪽인지 판정하기 위함**
    - 연속적으로 존재하며 의미 있고 관계가 깊은 문장
    - 전혀 관계가 없고 문맥의 연결이 없는 두 문장

BERT의 특징

- BERT의 세가지 특징
  - 문맥에 의존한 단어 벡터 표현을 만들 수 있게 되었다.
  - 자연어 처리 작업에서 파인 튜닝이 가능해 졌다.
  - Attention에 의한 설명과 시각화가 간편해 졌다.

BERT의 특징

- **문맥에 의존한 단어 벡터 표현을 만들 수 있게 되었다.**
  - **어떤 언어라도 단어의 의미가 단 하나인 경우는 적음**
    - (예) bank: 은행, 강변 이라는 의미가 있음
  - **다양한 의미를 가진 각 단어들은 문맥에 따라 단어의 의미가 바뀜**
  - **BERT는 문맥에 맞는 단어의 벡터 표현이 가능함**

BERT의 특징

- **BERT는 12단 Transformer를 사용**
  - Embedding 모듈에서 단어ID를 단어 벡터로 변환할 때는 은행의 bank와 강변의 bank는 동일한 길이(768)의 단어 벡터
  - 12단의 Transformer를 거치는 동안 단어 bank의 위치에 있는 특징량 벡터는 변화함
  - 변화 결과, 12단 째의 출력인 단어, bank의 위치에 있는 특징량 벡터는 최종적으로 은행 bank와 강변 bank가 서로 다른 벡터가 됨
  - 여기서 말하는 특징량 벡터란, 사전 학습의 MLM이 풀어놓은 특징량 벡터
  - 문장 중의 단어 bank와 그 주변 단어와의 관계성을 바탕으로 하여 Transformer의 Self-Attention 처리로 작성됨
  - 동일한 단어도 주변 단어와의 관계성에 따라 문맥에 맞는 단어 벡터가 생성됨

BERT의 특징

- **자연어 처리 작업에서 파인 튜닝이 가능해 졌다.**
  - **BERT를 기반으로 다양한 자연어 처리를 수행하려면**
    - 두 언어 작업에서 사전 학습한 가중치 파라미터를 BERT 모델의 가중치로 설정
    - BERT 모델 구조 그림에서 나타낸 (seq_len x hidden)=(512x768) 텐서와 (hidden)=(768)의 두 텐서를 출력
    - 두 텐서를 실행하고 싶은 자연어 처리 작업에 맞춘 어댑터 모듈에 투입
    - 작업에 따른 출력 획득
    - (예) 긍정적/부정적 감정 분석의 경우, 어댑터 모듈로 하나의 전결합층을 추가하는 것 만으로 문자의 판정이 가능해짐

BERT의 특징

- 모델을 학습할 때, 기반이 되는 BERT와 어댑터 모듈의 전결합층 양쪽 모두를 파인 튜닝으로 학습
- BERT의 출력에 어댑터 모듈을 연결하여 다양한 자연어 처리 작업 수행 가능
- Object Detection을 위한 SSD 모델, 자세 추정을 위한 OpenPose 모델에서 사용된 기반 네트워크인 VGG와 같은 역할을 BERT가 수행
- 적은 문서 데이터로도 성능 좋은 모델의 작성이 가능함
- 자연어 처리 작업도 화상 작업처럼 전이학습 및 파인 튜닝을 적용할 수 있게 된 것이 BERT가 주목받은 요인의 하나임

BERT의 특징

- **BERT는 어떻게 화상 작업의 기본 모델인 VGG와 같은 전이학습 및 파인 튜닝의 기반 역할을 수행할 수 있을까?**
  - VGG 모델과 같이 화상 처리에서 화상 분류가 가능한 네트워크는 물체 감지나 시맨틱 분할에도 유효함
  - BERT도 사전작업 MLM을 풀 수 있는 **단어를 문맥에 맞는 특징량 벡터로 변환할 수 있는 능력**이 단어의 의미를 정확하게 파악할 수 있게 함
  - 사전작업 NSP로 **문장이 의미 있게 연결되었는지 여부를 판정할 수 있는 능력**이 문장의 의미를 이해할 수 있게 함
  - 단어와 문장의 의미를 이해할 수 있도록 사전학습을 하고 있으므로 자연어 처리 작업인 감정 분석 등에도 응용이 가능해짐

BERT의 특징

- **선구적인 범용 언어 모델의 사례를 만듦**
  - 단어와 문장의 의미를 제대로 파악해야 하는 사전 작업의 수행
  - 사전 작업으로 학습한 가중치를 기반으로, 어댑터를 자연어 처리 작업에 맞게 교체하여 파인 튜닝을 수행

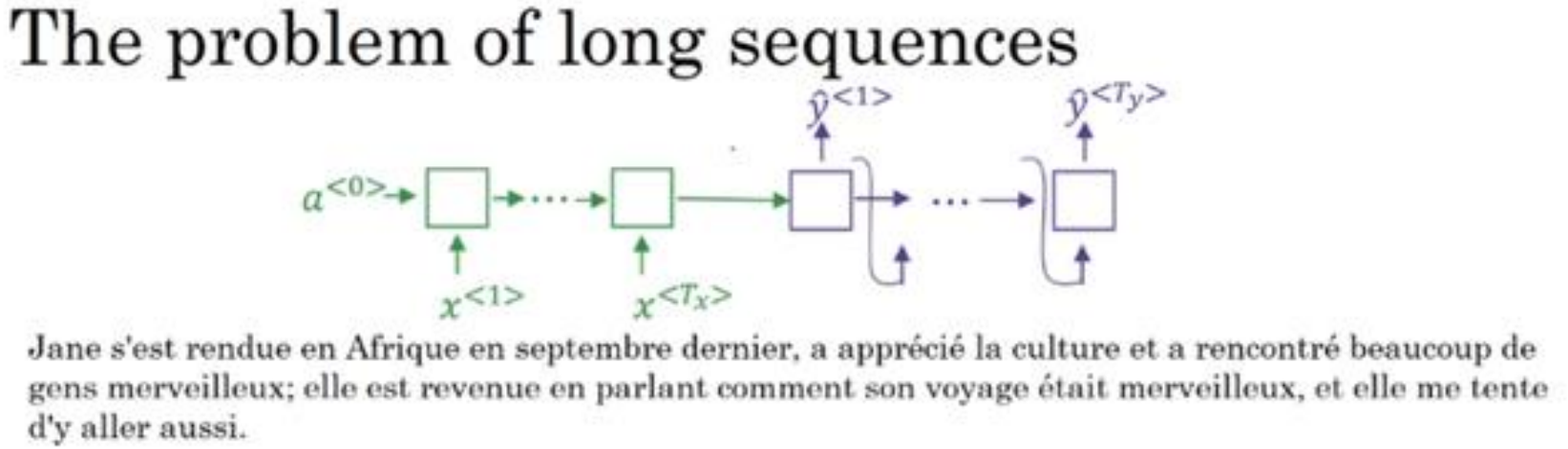
→ 이러한 처리의 흐름이 자연어 처리에서의 하나의 표준이 될 것으로 기대됨

BERT의 특징

- Attention에 의한 설명과 시각화가 간편해 졌다.
  - Attention: 결과에 영향을 준 단어의 위치정보
  - Attention을 시각화 함으로써 인간이 추론 결과를 설명하기가 쉬워짐

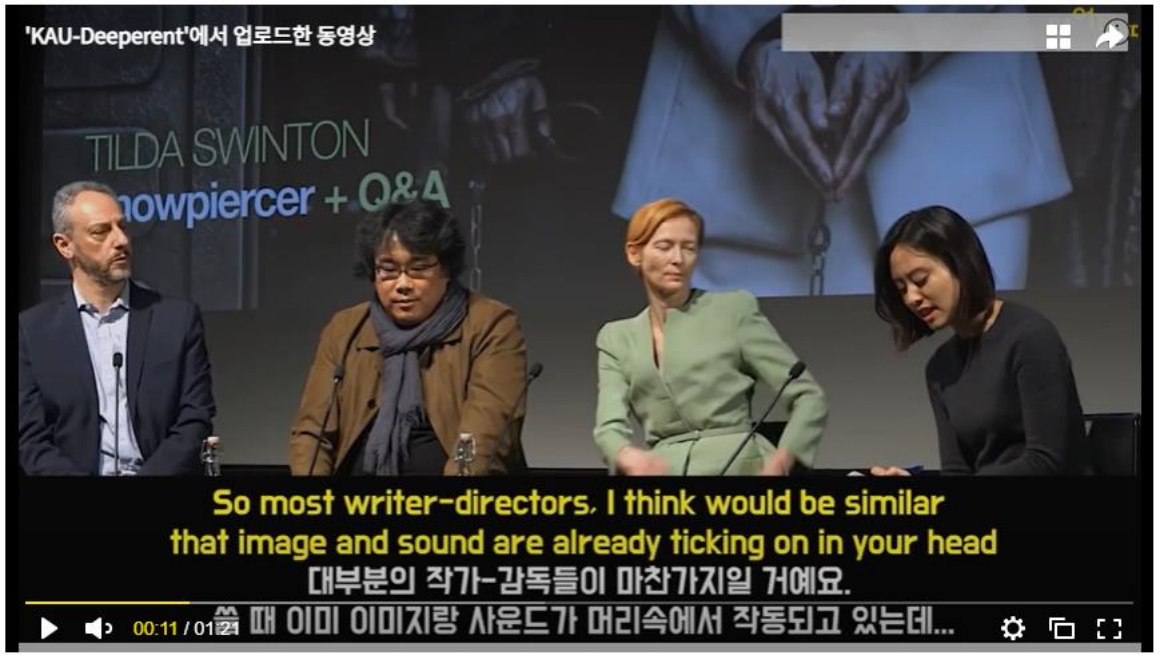
참고: Attention 모델

- 딥러닝 초창기의 기계번역 기술의 주요 방식은 Sequence 방식



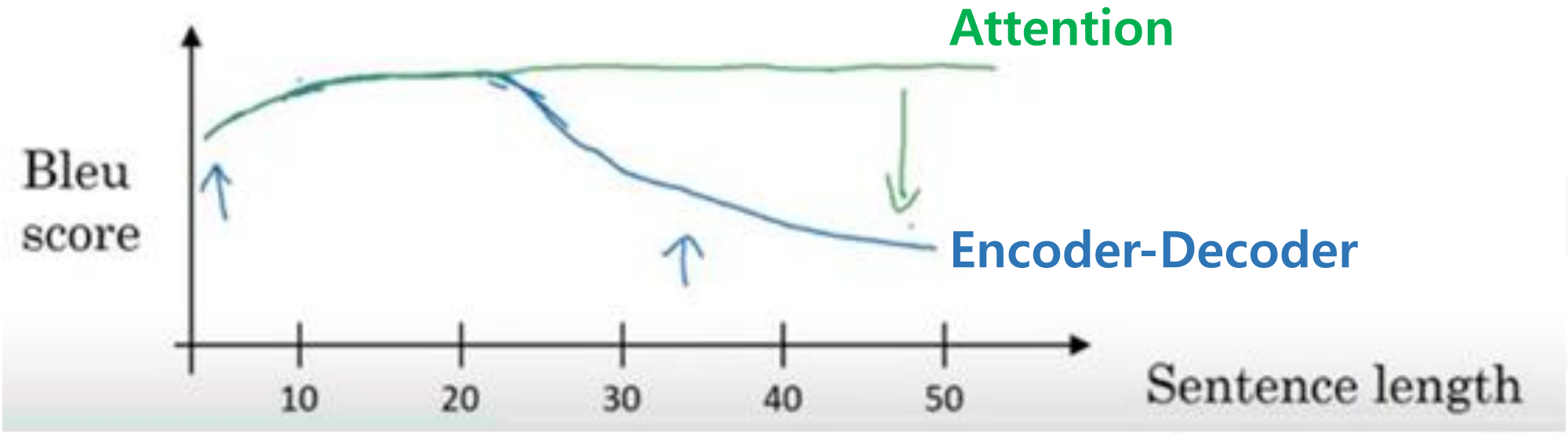
- 데이터를 토큰으로 나눠서 순차적으로 입력 데이터를 준 다음
- 순차적으로 출력을 뽑아내는 방식 → Encoder-Decoder 방식

참고: Attention 모델



- 인간의 경우도 기본적으로 순차 번역을 지향함
- Encoder-Decoder 형 모델
- 그러나 문장이 길어지면(30~40 단어 이상) 성능이 떨어짐

참고: Attention 모델



참고: Attention 모델

- 사람은 번역을 할 때, 모든 문장을 듣지만 각 단어를 번역할 때마다 모든 문장의 정보를 이용하지는 않는다.

- Encoder-Decoder 방식에서는 다음 단어를 번역할 때마다 C 라는 문맥 벡터를 전달하는데, 이 고정된 길이의 벡터에 그 동안 본 모든 단어에 대한 정보가 축약되어 있음 → 문장이 길어지면 효율 저하

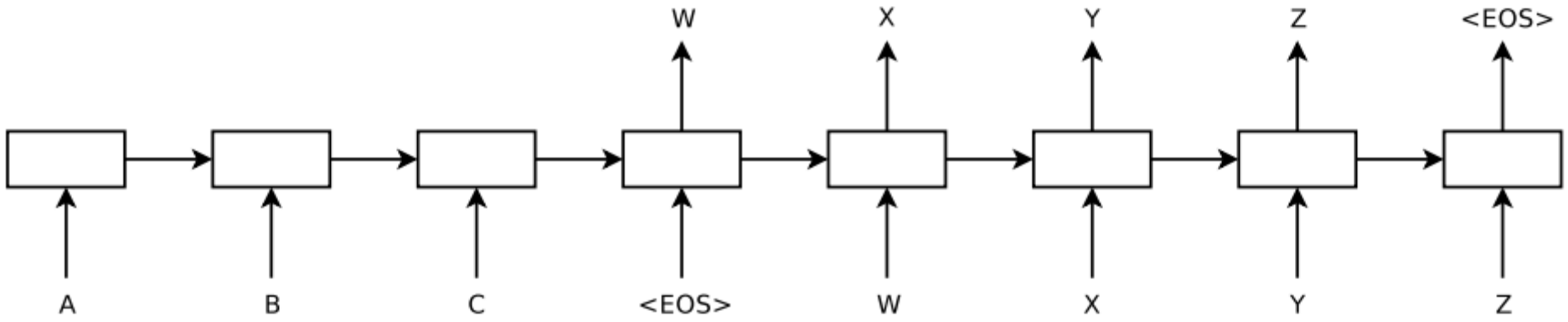
참고: Attention 모델

- 고정된 길이의 C 벡터에 모든 정보를 축약하지 말고, 매 Step마다 필요한 정보를 새로 만들자! → 해결/개선안 제시

→ Attention 모델

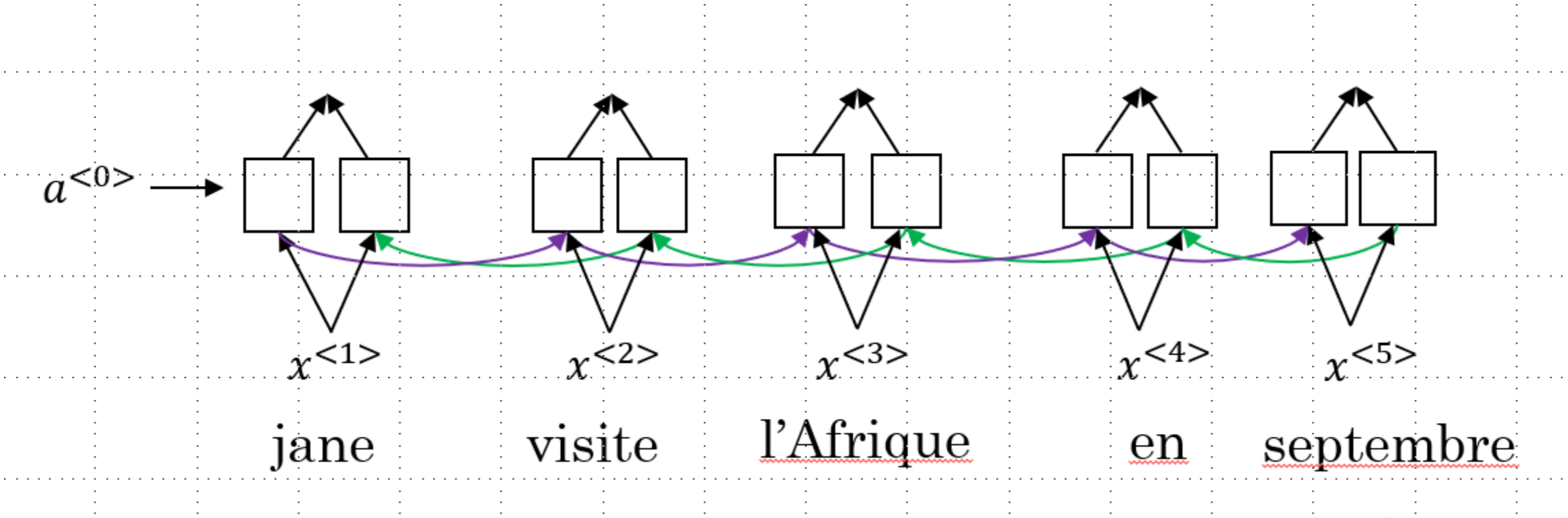
참고: Attention 모델

- RNN 모델에 대한 Attention 기법 적용
  - 기존의 encode는 word-for-word translation. 단어를 하나 번역하면 그 단어를 넘기는 방식



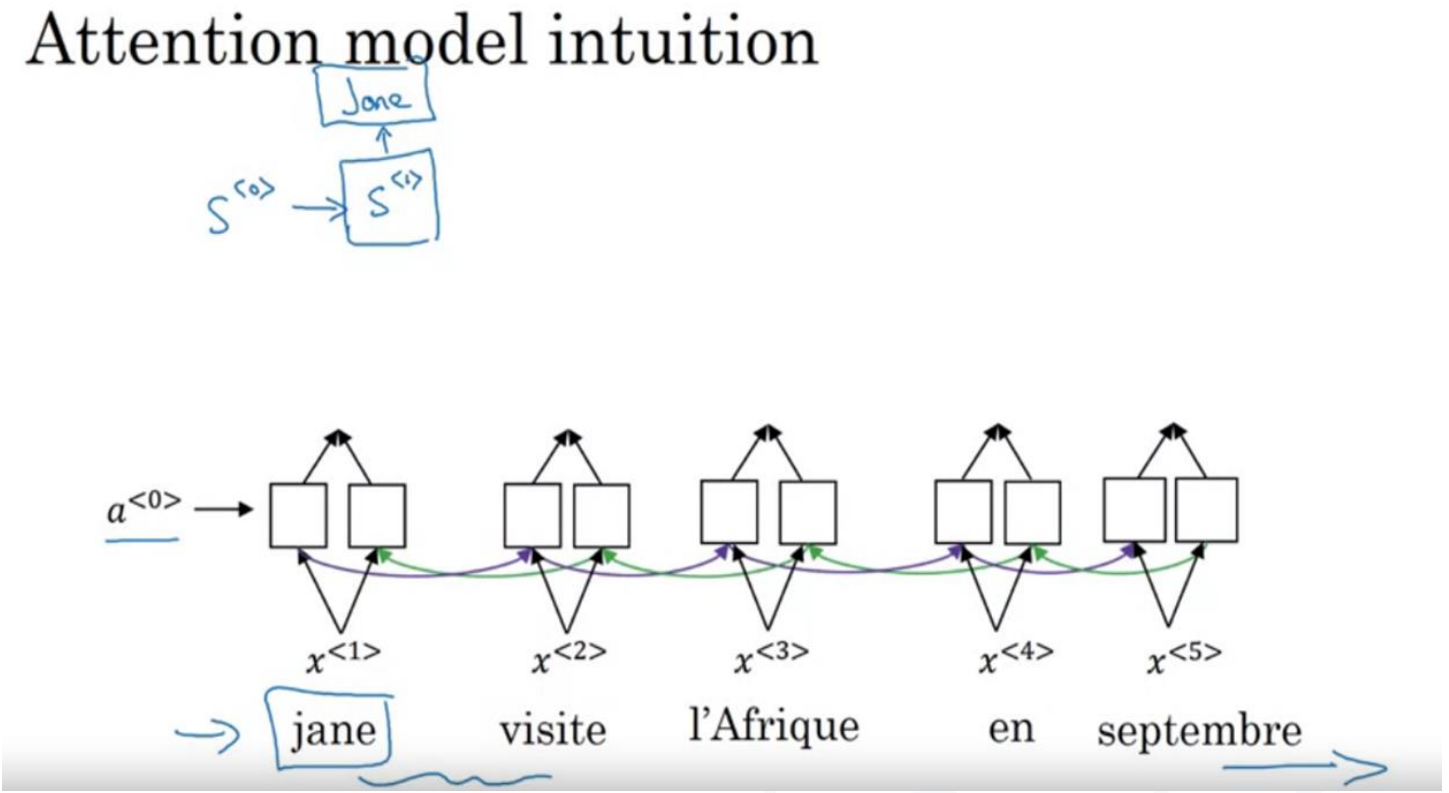
참고: Attention 모델

- RNN 모델에서의 번역



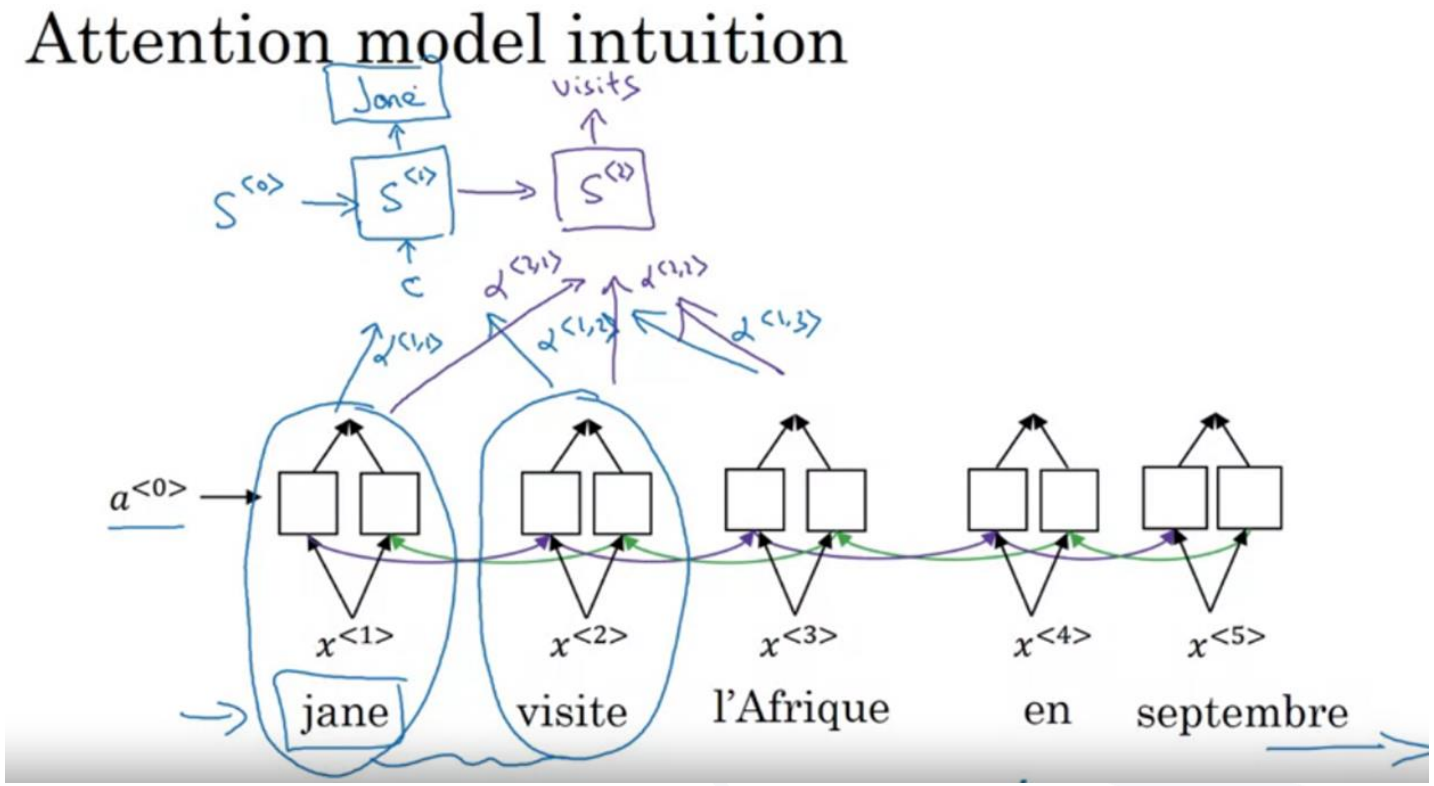
참고: Attention 모델

- **RNN 모델에서의 번역**
  - S: Hidden State
  - 우리는 첫번째 단어가 jane 이 될 것을 기대
  - 목표는 Jane visits to Africa September
  - 여기서 Jane이라는 이름의 결과물을 내려면 몇 개의 프랑스어 단어를 봐야하나?



참고: Attention 모델

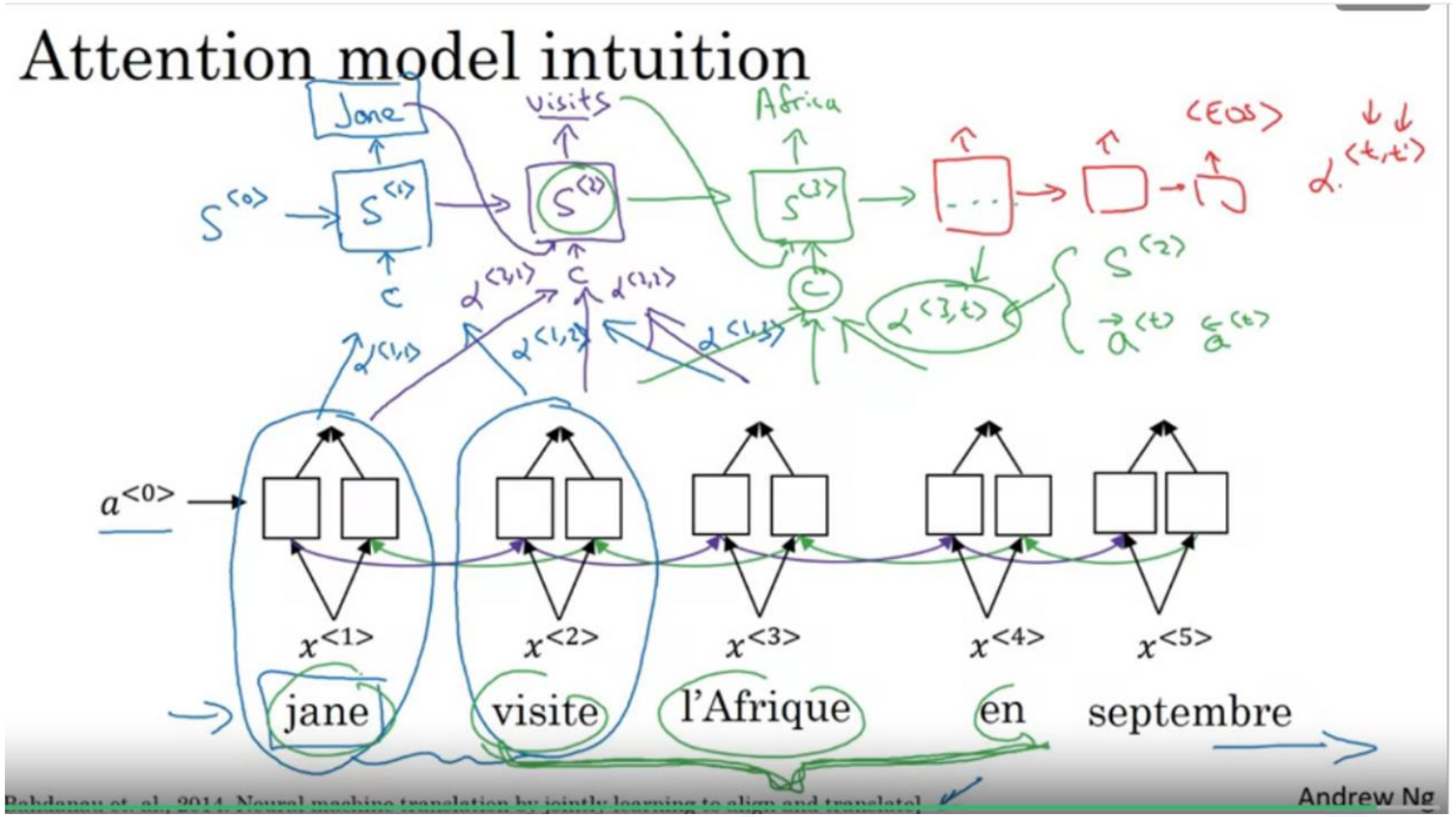
- 문장의 끝까지 볼 필요는 없음
- Attention 모델이 계산해야 할 것은 Attention Weight (A(1,1)로 표기)
- A(1, 1)에서 첫 번째 1은 첫 번째 단어를 생성하는데 얼마나 가중치를 부여할 것인가? 두 번째 1은 첫 번째 정보를 사용하겠다라는 의미



- A(1, 2): 첫번째 단어를 생성하는데 두 번째 정보를 사용하겠다. 의 의미
- 이런 정보들이 합쳐져서 맥락을 나타내는 벡터 C를 계산함

참고: Attention 모델

- 이런 구성을 가짐
- A(3, t)의 경우는,
  S(2), A(2, t), A(4, t)의
  영향을 받음



- A(t, t')는 t번째 단어 번역을 할 때, 원문의 T'번째 단어에 얼마나 Attention을 줄 것이냐?
  를 의미하게 됨 → 전체 원문에서 일부, 즉 Local Window에 Attention을 주겠다는 의미