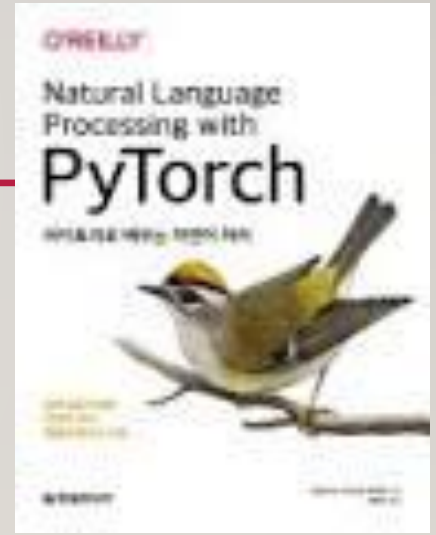
파이토치로 배우는 자연어 처리



7장. 자연어 처리를 위한 시퀀스 모델링 - 중급

8장. 자연어 처리를 위한 시퀀스 모델링 - 고급

AI융합학부 20193124 고경빈

목차

- 7. 자연어 처리를 위한 시퀀스 모델링 – 중급
  - 엘만 RNN의 문제점
  - 엘만 RNN의 문제 해결책: 게이팅
  - 예제: 문자 RNN으로 성씨 생성하기
  - 시퀀스 모델 훈련 노하우
- 8. 자연어 처리를 위한 시퀀스 모델링 – 고급
  - S2S, 인코더-디코더, 조건부 생성
  - 강력한 시퀀스 모델링: 양방향 순환 모델
  - 강력한 시퀀스 모델링: 어텐션
  - 시퀀스 생성 모델 평가
  - 예제: 신경망 기계 번역

시퀀스 예측(시퀀스 레이블링)

- 시퀀스의 각 항목에 레이블을 할당해야 함
- 자연어 모델링: 각 타임 스텝에서 주어진 단어 시퀀스를 기반으로 다음 단어를 예측
- 품사 태깅: 단어의 문법 품사를 예측
- 개체명 인식: 단어가 사람, 위치, 제품, 회사 같은 개체명에 속하는지 예측

엘만 RNN의 문제점

- 멀리 떨어진 정보를 유지하기 어려움
  - RNN이 선택적으로 업데이트를 결정하거나 업데이트할 때 상태 벡터의 어느 부분을 얼마만큼 업데이트할지 판단할 방법이 필요
- 그레디언트가 통제되지 않고 0이나 무한대를 만드는 경향이 있음
  - ReLU 함수 사용, 그레디언트 클리핑, 적절한 가중치 초기화, **게이팅 (Gating)**

게이팅

- $a + \lambda b$ 에서 λ는 덧셈에 b가 포함되는 양을 조절하는 게이트라고 할 수 있음
- 이전 타임 스텝의 은닉 상태가 h_{t-1}이고 현재 입력이 x_t인 엘만 RNN
  - $h_t = h_{t-1} + F(h_{t-1}, x_t)$
  - 조건 없는 덧셈 => 앞에서 설명한 이슈 발생
- 게이팅 함수를 사용한 RNN 업데이트 공식
  - $h_t = h_{t-1} + \lambda(h_{t-1}, x_t)F(h_{t-1}, x_t)$
- LSTM(Long Short-Term Memory) 신경망
  - 조건에 따라 업데이트하는 것뿐만 아니라 이전 은닉 상태의 값을 의도적으로 지움
  - $h_t = \mu(h_{t-1}, x_t)h_{t-1} + \lambda(h_{t-1}, x_t)F(h_{t-1}, x_t)$
- 업데이트 과정을 제어할 뿐만 아니라 그레디언트 이슈를 억제하고 훈련을 쉽게 만들어 줌

문자 RNN으로 성씨 생성하기(I)

- 조건이 없는 모델: GRU가 어떤 국적에도 편향된 계산을 수행하지 않음
  - 문자 인덱스를 임베딩하여 GRU로 상태를 순서대로 계산하고 Linear 층을 사용해 토큰의 예측 확률을 계산
  - 성씨가 몇 몇 형태소 패턴을 따르는 것 같지만 한 국적의 이름처럼 보이지 않음
  - 일반적인 성씨 모델을 학습하면 여러 국적 간의 문자 분포를 혼동하게 만들 수 있음

```
----------------
Hoten
Nannom
Coirhov
Dhovanov
Nagn
Agalawhe
Tatir
Reskahi
Psotch
Valsain
```

문자 RNN으로 성씨 생성하기(2)

- 조건이 있는 모델: 성씨를 생성할 때 국적 고려
  - 모델이 특정 성씨에 상대적으로 편향될 수 있는 메커니즘이 존재
  - 은닉 상태 크기의 벡터로 국적을 임베딩하여 RNN의 초기 은닉 상태를 만듦
  - 모델 파라미터가 수정될 때 임베딩 행렬의 값도 조정됨
    - 성씨의 국적과 규칙에 더 민감하게 예측하게 됨
  - 모델이 성씨 철자에 있는 어떤 패턴을 따름을 알 수 있음

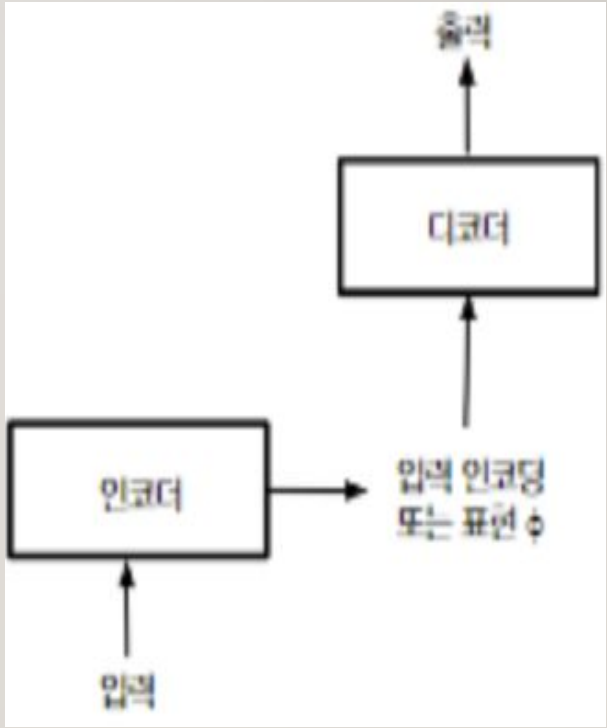
```
Arabic 샘플:
-   Saroem
-   Sorir
-   Gilla
Chinese 샘플:
-   Tho
-   Leuw
-   Dan
Czech 샘플:
-   Tchili
-   Derson
-   Sselon
Dutch 샘플:
-   Kuitt
-   Shajtein
-   Tirnons
```

시퀀스 모델 훈련 노하우

- 가능하면 게이트가 있는 셀을 사용
  - 게이트 구조는 수치 안정성과 관련된 여러 문제를 해결해 훈련을 쉽게 만듦
- 가능하면 LSTM보다 GRU 사용
  - GRU는 LSTM과 거이 비슷한 성능을 제공하면서 파라미터가 훨씬 적고 계산 자원도 덜 사용
- Adam 옵티마이저 사용
  - 안정적이고 다른 옵티마이저보다 빠르게 수렴
- 그레디언트 클리핑 사용
  - 범위를 가늠한 후 이상치를 클리핑하면 훈련 과정을 안정시킬 수 있음
- 조기종료 사용
  - 시퀀스 모델은 과대적합되기 쉬움. 검증 세트에서 측정한 오차가 상승하기 시작하면 조기종료 권장

S2S 모델, 인코더-디코더 모델, 조건부 생성

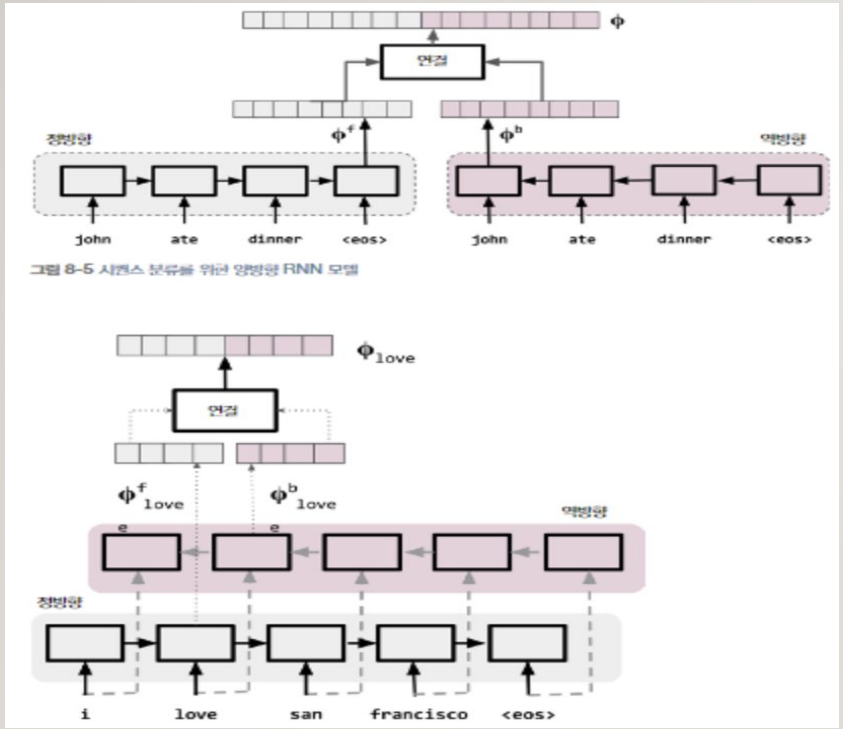
- S2S 모델은 인코더-디코더 모델의 일종
- 인코더는 입력에서 현재 문제와 관련된 중요한 성질을 감지하고, 디코더는 인코딩된 입력을 받아 원하는 출력을 만드는 데 사용
- 인코더와 디코더는 시퀀스 모델이고, 입/출력은 모두 시퀀스
- 두 시퀀스의 길이는 다를 수 있음
- 인코더-디코더 모델은 조건부 생성 모델의 일종
- 입력 표현 대신 일반적인 조건 문맥을 사용해 디코더가 출력을 만듦
- 조건 문맥은 구조적인 데이터 소스에서 올 수 있음
  - 조건 생성 모델이 모두 인코더-디코더 모델은 아님



양방향 순환 모델(I)

- 순환 모델은 시퀀스를 벡터로 인코딩하는 블랙 박스
- 시퀀스 모델링 할 때 지난 단어와 함께 앞으로 나타날 단어를 관찰하면 도움이 됨
- "The man who hunts ducks out on the weekends."
  - L => R 모델과 L ⇦⇨ R 모델은 ducks를 다르게 표현함
  - 과거와 미래의 정보를 합치면 문장에 있는 단어의 의미를 안정적으로 표현할 수 있음

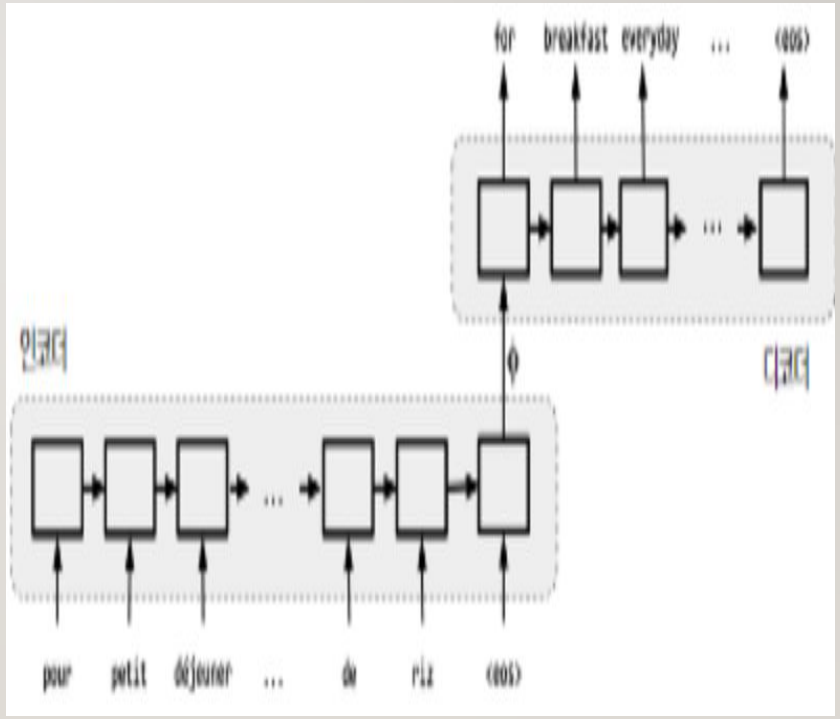
양방향 순환 모델(2)



- 모델이 어떻게 양방향으로 문장을 읽고 정방향과 역방향 표현을 합친 감성 분류를 위한 표현을 만드는지 보여주는 그림
- ϕ_{love}는 단어 love가 입력되는 타임 스텝에서 신경망의 은닉 상태에 대한 표현(인코딩)
- 입력의 각 단어에 대한 정방향 표현과 역방향 표현을 연결해서 단어의 최종 표현을 만듦

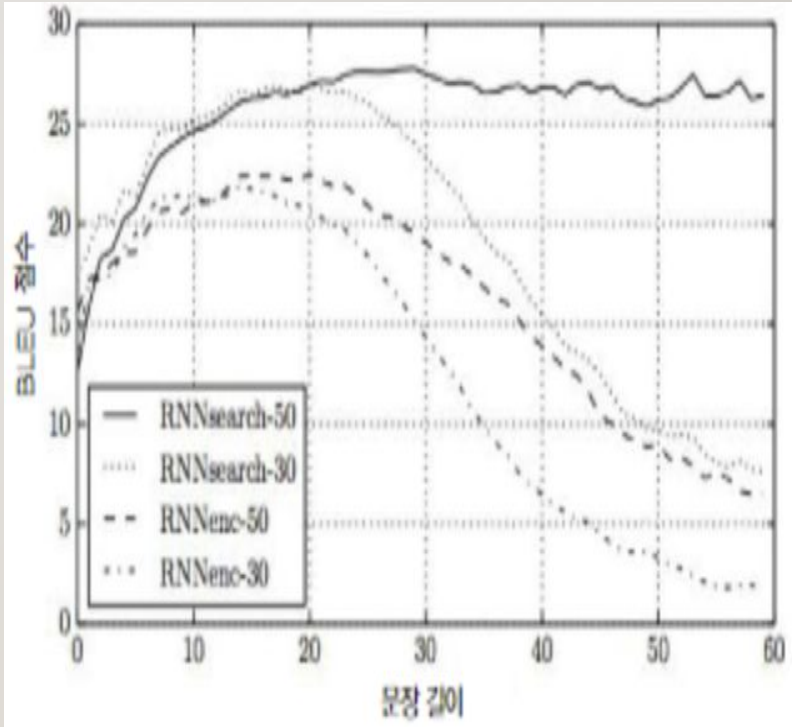
S2S 모델의 문제점

- 전체 입력 문장을 하나의 벡터에 밀어 넣음
- 긴 문장에서는 전체 입력 정보를 감지하지 못함
  - 인코딩에 최종 은닉 상태만 사용하는 제약 때문
- 긴 문장을 역전파할 때 그레디언트가 소실되어 훈련이 어려움



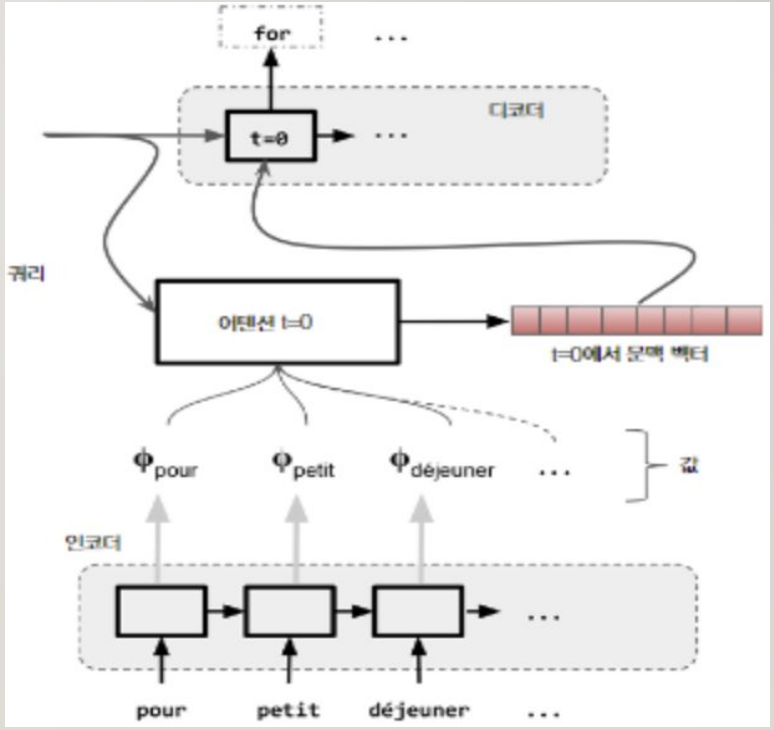
어텐션

- 출력을 생성할 때 관련된 입력 부분에 초점을 맞추는 현상
- 적은 기억 용량으로 여러 작업을 성공적으로 수행할 수 있음
- 어텐션 메커니즘
  - 전체 입력의 최종 요약이 아니라 입력의 여러 부분에 어텐션을 부여하는 시퀀스 생성 모델
- 입출력이 복잡한 딥러닝 모델의 성능을 높이는 데 유용함



심층 신경망의 어텐션

- 어텐션을 사용할 때 인코더의 최종 은닉 상태 뿐만 아니라 중간 타임 스텝의 은닉 상태도 고려함
- 값(value) or 키(key): 인코더의 은닉 상태
- 쿼리(Query): 디코더의 이전 은닉 상태
- 어텐션 벡터: 주의를 기울이려는 값의 개수와 차원이 같은 벡터
- 글림스: 어텐션 가중치가 값과 연결되어 생성하는 문맥 벡터
- 전체 문장의 인코딩 대신 문맥 벡터가 디코더의 입력이 됨
- 다음 타임 스텝의 어텐션 벡터는 호환성 함수를 사용해 업데이트 됨



어텐션을 구현하는 방법

- 콘텐츠 인식 어텐션
- 쿼리 벡터와 키만 사용하는 위치 기반 어텐션
- 소프트 어텐션: 어텐션 가중치가 0과 1 사이의 실수
- 하드 어텐션: 어텐션 가중치가 0 아니면 1인 이진 벡터를 학습
- 전역 어텐션: 입력의 모든 타임 스텝에 대해 인코더의 상태를 사용
- 지역 어텐션: 현재 타임 스텝 주위에 있는 입력에만 의존
- 지도 어텐션: 동시에 훈련되는 별도의 신경망을 사용해 어텐션 함수를 학습
- 멀티헤드 어텐션: 트랜스포머 네트워크를 위한 어텐션
- 멀티모달 어텐션: 이미지와 음성처럼 입력의 형태가 다양할 때 사용

시퀀스 생성 모델 평가(I)

- 분류 지표는 정답이 여럿이 모델이 도움이 안됨
  - 시퀀스 모델은 참조 출력으로 평가
- 사람 평가
  - 한 명 이상의 사람이 모델 출력에 '좋음' 또는 '나쁨 ' 을 표시하거나 번역을 고치는 방법
  - 사람이 작업할 때와 유사하게 시스템 출력의 최종 목표와 비슷한 간단한 에러율을 만듦
  - 평가 속도가 느리고 비용이 많이 들며 구하기 어려워 자주 사용하지 않음
  - 사람들 간의 평가가 다를 수 있음 => 평가자 간이 일치율을 함께 사용함
    - HTER: 추가, 삭제, 이동 횟수를 헤어려 계산한 가중치가 적용된 편집 거리

Judge Sentence

You have already judged 14 of 3064 sentences, taking 86.4 seconds per sentence.

Source: les deux pays constituent plutôt un laboratoire nécessaire au fonctionnement interne de l ' ue .

Reference: rather , the two countries form a laboratory needed for the internal working of the eu .

Translation	Adequacy	Fluency
both countries are rather a necessary laboratory the internal operation of the eu .	1 2 3 4 5	1 2 3 4 5
both countries are a necessary laboratory at internal functioning of the eu .	1 2 3 4 5	1 2 3 4 5
the two countries are rather a laboratory necessary for the internal workings of the eu .	1 2 3 4 5	1 2 3 4 5
the two countries are rather a laboratory for the internal workings of the eu .	1 2 3 4 5	1 2 3 4 5
the two countries are rather a necessary laboratory internal workings of the eu .	1 2 3 4 5	1 2 3 4 5
Annotator: Philipp Koehn **Task:** WMT06 French-English		Annotate
Instructions	5= All Meaning 4= Most Meaning 3= Much Meaning 2= Little Meaning 1= None	5= Flawless English 4= Good English 3= Non-native English 2= Disfluent English 1= Incomprehensible

시퀀스 생성 모델 평가(2)

- 자동 평가
  - 실행하기 쉽고 빠름
  - N-그램 중복 기반 지표: 참조와 출력이 얼마나 가까운지 n-그램 중복 통계로 점수 계산
  - 혼란도: 정보 이론에 기반한 자동 평가 지표(출력 시퀀스의 확률을 측정할 수 있다면 적용 가능)
    - $Perplexity(x) = 2^{-P(x)\log P(x)}$
  - 따로 떼어 놓은 데이터셋에서 모델의 혼란도를 측정해 여러 시퀀스 생성 모델 비교 가능
  - 계산하기 쉽지만, 시퀀스 생성 모델의 평가에 사용할 때 문제점 발생
    - 혼란도는 과장된 지표: 모델 성능의 작은 차이가 혼란도에서 큰 차이를 만듦
    - 모델의 오차율에 직접 반영되지 않음
    - 혼란도가 향상되더라도 사람이 판단하기에 향상되었다고 느끼지 못할 수 있음

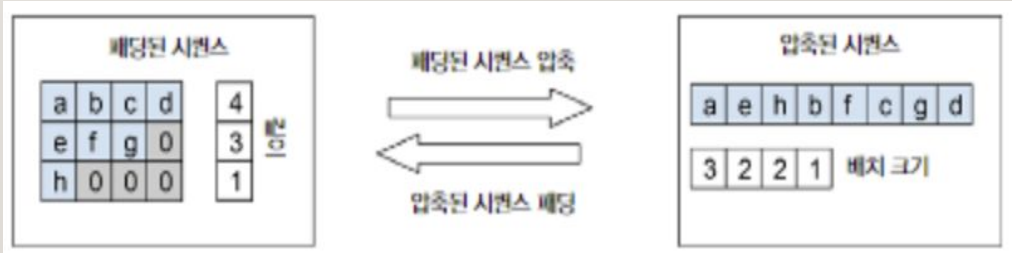
기계 번역 데이터셋

- 모든 문장을 소문자로 만들고 NLTK의 영어, 프랑스어 토큰화를 각 문자 쌍에 적용
- NLTK의 언어에 특화된 단어 토큰화를 적용해 토큰 리스트를 만듦
- 특정 문장 패턴을 지정하여 데이터의 일부분만 선택해 학습 문제를 단순하게 만듦
  - 제한된 문장 패턴으로 데이터 범위를 좁힘
- 훈련하는 동안 모델의 분산을 낮추고 짧은 훈련 시간 안에 높은 성능 달성 가능

NMT를 위한 벡터 파이프라인

- 소스와 타깃을 다루는 방법이 다름
  - 소스 시퀀스는 시작 부분에 **BEGIN-OF-SEQUENCE** 토큰이 추가되고 마지막에 **END-OF-SEQUENCE** 토근이 추가되어 벡터화 됨
  - 타깃 시퀀스는 토큰 하나가 밀린 복사본 두 개로 벡터화 됨

- **PackedSequence**를 사용하려면 소스 시퀀스의 길이에 따라 각 미니배치를 정렬해야 함
  - 각 시퀀스의 길이를 알아야 하고, 시퀀스의 길이 순서대로 내림차순으로 정렬되어야 함
  - 정렬된 행렬을 만들기 위해 미니배치에 있는 텐서를 시퀀스 길이 순서대로 정렬함



NMT 모델의 인코딩과 디코딩

- 인코더가 먼저 소스 시퀀스를 양방향 GRU로 벡터 상태의 시퀀스로 매핑함
- 디코더가 인코더의 은닉 상태를 초기 은닉 상태로 만들고 어텐션 메커니즘으로 소스 시퀀스에 있는 다양한 정보를 선택해 출력 시퀀스로 만듦
- 일반적으로 인코더는 정수 시퀀스를 입력으로 받아 위치마다 특성 벡터를 만듦

어텐션 자세히 알아보기

- 쿼리 벡터와 키 벡터를 입력으로 받아 값 벡터를 선택하는 일련의 가중치를 계산
- 디코더의 은닉 상태를 쿼리 벡터로 사용하고 인코더 상태 벡터를 키와 값 벡터로 사용
- 디코더의 은닉 상태와 인코더 상태 벡터의 점곱은 인코딩된 시퀀스에 있는 아이템마다 스칼라 값 하나를 만듦
- 소프트맥스 함수를 사용해 이 스칼라 값을 인코더 상태 벡터에 대한 확률 분포로 변환
- 이 확률을 사용해 상태 벡터에 가중치를 적용한 다음 모두 더해서 배치 아이템마다 벡터 하나를 만듦

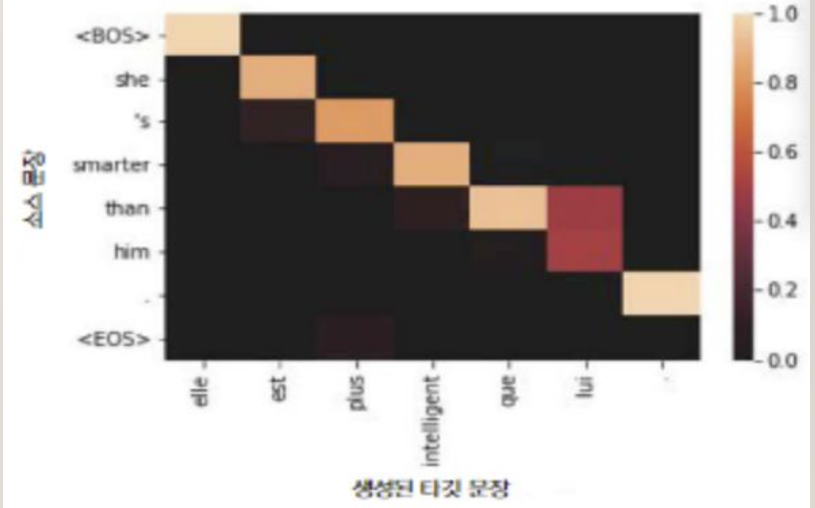
탐색 학습과 스케줄링 된 샘플링

- 모델은 타깃 시퀀스가 제공되고 디코더에서 타임 스텝마다 입력으로 사용된다고 가정함
  - 테스트 시에는 이런 가정이 어긋남 => 모델이 생성할 시퀀스를 알 수 없음
- 훈련하는 동안 자체 예측하는 방법 사용
  - 모델이 스스로 경로를 샘플링하는 기법은 데이터셋의 타깃 시퀀스에서 벗어날 때 확률 분포가 더 나아지도록 모델을 최적화할 수 있음

모델 훈련 결과

- 스케줄링 된 샘플링을 사용하지 않는 모델
  - 제공된 타깃 시퀀스를 타임 스텝마다 디코더의 입력으로 사용함
- 스케줄링 된 샘플링을 사용하는 모델
  - 스케줄링 된 샘플링을 사용해 모델이 자체 예측을 만들어 디코더의 입력으로 사용
  - 모델이 자체적인 오류를 최적화하도록 함

모델	BLEU 점수
스케줄링된 샘플링을 사용하지 않는 모델	46.8
스케줄링된 샘플링을 사용하는 모델	48.1



Q & A