

## CS 224n : 과제 # 4

이 과제는 두 섹션으로 나뉩니다. *RNN을 사용한 신경 기계 번역*과 *NMT 시스템 분석*. 첫 번째는 주로 코딩과 구현에 중점을 두는 반면 두 번째는 완전히 서면 분석 질문으로 구성됩니다. 첫 번째 섹션에 갇혀 있으면 두 섹션이 서로 독립적이므로 항상 두 번째 섹션에서 작업 할 수 있습니다. NMT 시스템은 이전에 이 클래스에서 구성한 신경망보다 더 복잡하며 GPU에서 훈련하는 데 4 시간이 걸립니다. 따라서 이 과제를 일찍 시작하는 것이 좋습니다. 마지막으로 NMT 시스템의 표기법과 구현은 약간 까다롭기 때문에 도중에 갇히면 TA가 지원할 수 있도록 Office Hours에 오십시오.

### 1. RNN을 사용한 신경 기계 번역 (45 점)

기계 번역에서 우리의 목표는 *출처 언어* (예 : 스페인어)를

*표적 언어* (예 : 영어). 이 과제에서는 신경망 기계 번역 (NMT) 시스템을 구축하기 위해주의를 기울여 시퀀스-시퀀스 (Seq2Seq) 네트워크를 구현합니다. 이 섹션에서는 훈련 절차 Bidirectional LSTM Encoder와 Unidirectional LSTM Decoder를 사용하는 제한 된 NMT 시스템을 위해.

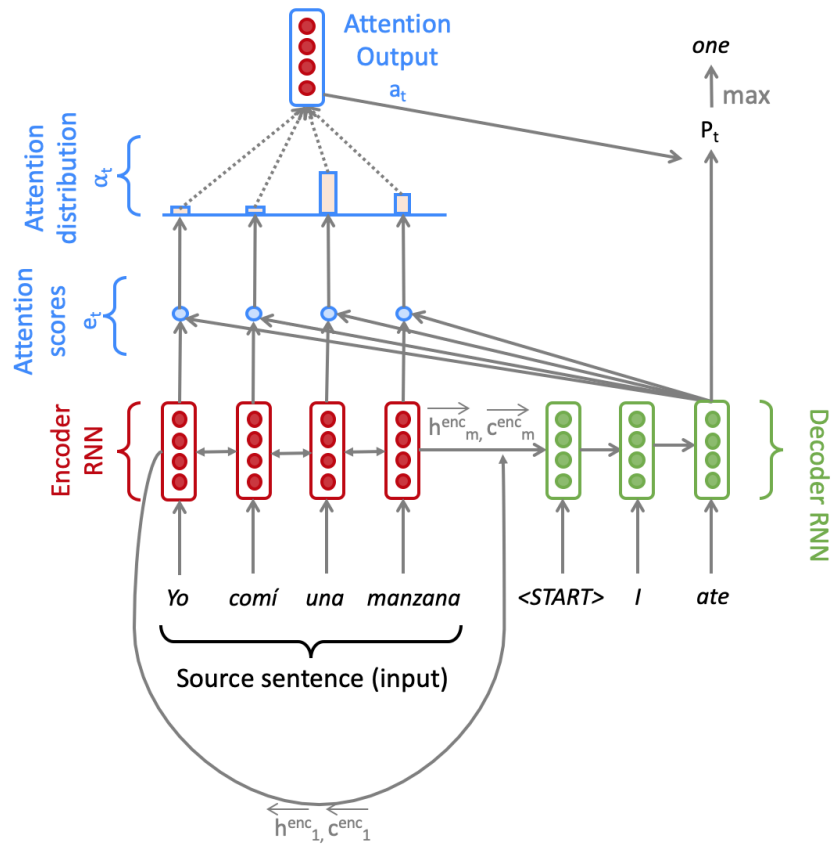


그림 1 : 디코더의 세 번째 단계에 표시된 곱셈주의가있는 Seq2Seq 모델. 숨겨진 상태  $h_{enc}$  및 세포 상태  $c_{enc}$  다음 페이지에서 정의됩니다.

## 모델 설명 (훈련 절차)

소스 언어로 된 문장이 주어지면 임베딩 행렬에서 임베딩이라는 단어를 찾습니다.

급할 수 있는 엑스  $1, \dots, \text{엑스}_{\text{미디엄}}$  (엑스  $n$ 는  $\in$  아르 자형  $\text{이자형} \times 1$ ), 어디  $\text{미디엄}$  소스 문장의 길이이고  $\text{이자형}$  임베딩 크기입니다. 이러한 임베딩을 양방향 인코더에 공급하여 두 가지 모두에 대해 숨겨진 상태와 셀 상태를 생성합니다.

포워드 ( $\rightarrow$ ) 뒤로 ( $\leftarrow$ ) LSTM. 정방향 및 역방향 버전이 연결되어 숨겨진 상태를 제공합니다.  $h_{\text{enc}}$  및 세포 상태  $ss_{\text{enc}}$

$n$ 는  $n$ 는 :

$$h_{\text{enc}} = [h_{\text{ko}} \leftarrow \dots \leftarrow ss_{\text{enc}}] \text{ 어디서 } h_{\text{enc}} \in \text{아르 자형 } 2h \times 1 \leftarrow \dots, h_{\text{enc}} = [h_{\text{ko}} \leftarrow \dots \leftarrow ss_{\text{enc}}] \in \text{아르 자형 } h \times 1 \quad 1 \leq n \leq \text{미디엄} \quad (1)$$

$$ss_{\text{enc}} = [ss_{\text{enc}} \leftarrow \dots \leftarrow ss_{\text{enc}}] \text{ 어디서 } ss_{\text{enc}} \in \text{아르 자형 } 2h \times 1, ss_{\text{enc}} = [ss_{\text{enc}} \leftarrow \dots \leftarrow ss_{\text{enc}}] \in \text{아르 자형 } h \times 1 \quad 1 \leq n \leq \text{미디엄} \quad (2)$$

그런 다음 디코더의 첫 번째 숨겨진 상태를 초기화합니다.  $h_{12}$  및 세포 상태  $ss_{12}$  및 인코더의 최종 은닉 상태와 최종 셀 상태의 선형 투영으로. <sup>1</sup>

$$h_{12} = W_h [h_{\text{enc}} \leftarrow \dots \leftarrow h_{\text{ko}}] \text{ 어디서 } h_{12} \in \text{아르 자형 } h \times 1, W_h \in \text{아르 자형 } h \times 2h \quad (3)$$

$$ss_{12} = W_{ss} [ss_{\text{enc}} \leftarrow \dots \leftarrow ss_{\text{enc}}] \text{ 어디서 } ss_{12} \in \text{아르 자형 } h \times 1, W_{ss} \in \text{아르 자형 } h \times 2h \quad (4)$$

디코더가 초기화되었으므로 이제 대상 문장을 제공해야 합니다. 예  $E/\text{단계}$ , 우리는

임베딩  $E/\text{단계}$  워드, 와이  $E/\text{단계}$  아르 자형  $\text{이자형} \times 1$ . 그런 다음 와이  $E/\text{단계}$  더블어 결합 출력 벡터

영형  $E/\text{단계} - 1 \in$  아르 자형  $h \times 1$  이전 타임 스텝에서 (이 페이지의 첫부분에서 설명하겠습니다!)

와이  $E/\text{단계}$  아르 자형  $(e + h) \times 1$ . 첫 번째 대상 단어 (예 : 시작 토큰)의 경우 영형  $0$  제로 벡터입니다. 그런 다음 우리는 와이  $E/\text{단계}$  디코더에 대한 입력으로.

$$h_{12} \text{ 및 } ss_{12} \text{는 디코더 (와이 } E/\text{단계}, h_{12} \text{ 및 } ss_{12} \text{ 어디서 } h_{12} \in \text{아르 자형 } h \times 1, ss_{12} \in \text{아르 자형 } h \times 1 \quad (5)$$

$$(6)$$

그런 다음  $h_{12}$  및  $ss_{12}$ 를 계산하기 위해  $h_{\text{enc}}$

$$1, \dots, h_{\text{enc}} \text{ 미디엄}$$

$$\text{이자형 } t, i = (h_{12} \text{ 및 } W_{\text{attProj}} h_{\text{enc}} \text{ 어디서 } \text{이자형 } E/\text{단계} \in \text{아르 자형 } \text{미디엄} \times 1, W_{\text{attProj}} \in \text{아르 자형 } h \times 2h \quad 1 \leq n \leq \text{미디엄} \quad (7)$$

$$\alpha_t = \text{엑스} \max (\text{이자형 } E/\text{단계} \text{ 어디서 } \alpha_t \in \text{아르 자형 } \text{미디엄} \times 1 \quad (8)$$

$$\alpha_t = \frac{\alpha_t}{h_{12}} \text{ 어디서 } \alpha_t \in \text{아르 자형 } 2h \times 1 \quad (9)$$

이제주의 출력을 연결합니다.  $\alpha_t$  디코더 숨겨진 상태  $h_{12}$  및 선형 레이어, tanh 및 드롭 아웃을 통과하여 결합된 출력 벡터 영형  $E/\text{단계}$

$$y_t = f(\alpha_t, h_{12}) \text{ 어디서 } y_t \in \text{아르 자형 } h \times 1 \quad (10)$$

$$V_t = W_V y_t \text{ 어디서 } V_t \in \text{아르 자형 } h \times 1, W_V \in \text{아르 자형 } h \times h \quad (11)$$

$$\text{영형 } t = \text{드롭 아웃} (\tanh(V_t)) \text{ 어디서 } \text{영형 } t \in \text{아르 자형 } h \times 1 \quad (12)$$

그런 다음 확률 분포를 생성합니다. 피  $E/\text{단계}$  대상 단어 이상  $E/\text{단계}$  타임 스텝 :

<sup>1</sup> 명확하지 않다면 우리가  $[1, h_{\text{미디엄}}$

$h_{\text{enc}} \leftarrow \dots \leftarrow h_{\text{enc}}$  인코더의 최종 숨겨진 상태로.

$$\text{피 } l = \text{소프트 맥스 (W 어휘 영형 } e_l) \text{ 어디 피 } e_l \in \text{아르 자형 } V_{\text{rel}} \times 1, \text{W 어휘 } \in \text{아르 자형 } V_{\text{rel}} \times h \quad (13)$$

여기,  $V_{e_l}$  목표 어휘의 크기입니다. 마지막으로 네트워크를 훈련시키기 위해 다음 사이의 소프트 맥스 교차 엔트로피 손실을 계산합니다. 피  $e_l$ 과 지  $e_l$  어디 지  $e_l$  시간 단계에서 대상 단어의 원-핫 벡터입니다.  $E_l$ :

$$\text{제 } O_{l|e_l}(\theta) = \text{CrossEntropy}(\text{피 } e_l, \text{지 } e_l) \quad (14)$$

여기,  $\theta$  모델의 모든 매개 변수를 나타내고  $\text{제 } O_{l|e_l}(\theta)$  단계에서의 손실입니다  $E_l$  디코더의. 모델에 대해 설명 했으므로 이제 스페인어에서 영어로 번역 할 수 있도록 구현해 보겠습니다!

## 가상 머신 설정

의 지침을 따르십시오. [CS224n Azure 가이드](#) (링크는 웹 사이트 및 Piazza에서도 제공됨) VM 인스턴스를 만들 수 있습니다. 약 45 분이 소요됩니다. 모델을 학습하려면 GPU가 필요하지만 VM에서 학습을 시도하기 전에 먼저 로컬에서 코드를 개발하고 실행되는지 확인하는 것이 좋습니다. GPU 시간은 비싸고 제한적입니다. 대략 걸립니다 4 시간 NMT 시스템을 훈련합니다. 모델을 훈련하고 평가하는 것보다 모델 디버깅에 실제로 GPU 시간을 모두 사용하는 것을 원하지 않습니다. 드디어, VM을 사용하지 않을 때마다 VM이 꺼져 있는지 확인하십시오.

Azure 구독 비용이 부족하면 VM이 일시적으로 잠기고 액세스 할 수 없게됩니다. 이 경우 Piazza에 이름, Azure 및 SUNetID에 사용 된 이메일로 비공개 게시물을 작성하여 더 많은 크레딧을 요청하세요.

모델 코드를 실행하려면 현지 다음 명령을 실행하여 적절한 가상 환경을 만드십시오.

```
conda env create --file local env.yml
```

이 가상 환경은 ~하지 않을 것이다 VM에 필요합니다.

## 구현 및 서면 질문

(a) (2 포인트) (코딩) 텐서 연산을 적용하기 위해 우리는

주어진 배치의 길이가 같습니다. 따라서 배치에서 가장 긴 문장을 식별하고 다른 문장을 동일한 길이로 채워야합니다. 구현 패드 전송 기능 `utils.py`, 패딩 된 문장을 생성합니다.

(b) (3 점) (코딩) 초기화 기능 모델 `embeddings.py` 초기화하려면 --

필요한 소스 및 타겟 임베딩.

(c) (4 점) (코딩) 초기화 기능 `nmt model.py` 필요한 초기화 --

모델 임베딩 ( `ModelEmbeddings` 수업에서 모델 `embeddings.py`) NMT 시스템을위한 레이어 (LSTM, 프로젝션 및 드롭 아웃).

(d) (8 점) (코딩) 인코딩 기능 `nmt model.py`. 이 함수는

패딩 된 소스 문장을 텐서로 엑스, 생성  $h_{enc}$   $1, \dots, h_{enc}$  미디엄, 초기 값을 계산합니다. 상태  $h_{12}$  및 초기 셀  $12$  월 디코더를 위해. 다음을 실행하여 포괄적이지 않은 온 전성 검사를 실행할 수 있습니다.

파이썬 `sanity_check.py` 1d

(e) (8 점) (코딩) 풀다 기능 `nmt_model.py`. 이 함수는 -

- 그리고 실행 단계 입력에 대한 모든 시간 단계에 대해 가능합니다. 다음을 실행하여 포괄적이지 않은 온 전성 검사를 실행할 수 있습니다.

파이썬 `sanity_check.py 1e`

(f) (10 점) (코딩) 단계 기능 `nmt_model.py`. 이 기능은 -

단일 시간 단계에 대한 디코더의 LSTM 셀, 대상 단어의 인코딩 계산  $h_{t-1}$  및  $z_{t-1}$  그만큼 주의 점수 이자형  $e_t$ , 주의 분포  $\alpha_{t-1}$ , 주의 출력  $h_t$  그리고 마지막으로 결합 된 출력 영형  $e_t$ . 다음을 실행하여 포괄적이지 않은 온 전성 검사를 실행할 수 있습니다.

파이썬 `sanity_check.py 1f`

(g) (3 점) (작성) 보낸 마스크 생성 () 기능 `nmt_model.py`. 텐서를 생성 -

호출 `enc` 마스크. 모양 (배치 크기, 최대 소스 문장 길이)을 가지고 있으며 입력의 '패드'토큰에 해당하는 위치에 1을 포함하고 패드가 아닌 토큰의 경우 0을 포함합니다. 주의력 계산 중에 마스크가 어떻게 사용되는지 살펴보십시오. 단계() 기능 (라인 295-296).

먼저 마스크가 전체주의 계산에 미치는 영향을 설명합니다 (약 세 문장으로). 그런 다음 이러한 방식으로 마스크를 사용해야 하는 이유를 (한두 문장으로) 설명하십시오.

이제 작업을 실행할 시간입니다! 다음을 실행하여 필요한 어휘 파일을 생성합니다.

`sh run.sh 어휘`

앞서 언급했듯이 개인용 컴퓨터에서 코드를 개발하는 것이 좋습니다. 적절한 `conda` 환경에서 실행 중인지 확인한 후 다음 명령을 실행하여 로컬 컴퓨터에서 모델을 학습시킵니다.

`sh run.sh train_local`

코드가 크래시되지 않는지 확인한 후에는 `iter 10` 또는 `iter 20`), Azure 웹 포털에서 VM의 전원을 켭니다. 그런 다음 *VM에 대한 코드 배포 관리* 우리의 섹션 *VM에 대한 실용 가이드* 코드를 VM에 업로드하는 방법에 대한 지침은 웹 사이트 및 Piazza에서도 제공되는 링크입니다.

다음으로 다음을 실행하여 필요한 패키지를 VM에 설치합니다.

`pip install -r gpu_requirements.txt`

마지막으로 *VM에서 프로세스 관리* 실습 가이드의 섹션을 참조하고 지침에 따라 새 `tmux` 세션을 만듭니다. 구체적으로 다음 명령을 실행하여 `tmux` 세션을 만듭니다.

`nmt`.

`tmux 새로운 -s nmt`

VM이 구성되고 `tmux` 세션에 있으면 다음을 실행합니다.

`sh run.sh 기차`

코드가 제대로 실행되고 있음을 알게되면 세션에서 분리하고 서버에 대한 `ssh` 연결을 닫을 수 있습니다. 세션에서 분리하려면 다음을 실행하십시오.

`tmux 분리`

다음을 실행하여 서버로 다시 `ssh-ing`하고 `tmux` 세션에 연결하여 학습 모델로 돌아갈 수 있습니다.

`tmux a -t nmt`

- (i) (4 점) 모델 학습이 완료되면 ( VM에서 약 4 시간이 소요됩니다),  
다음 명령을 실행하여 모델을 테스트하십시오.

sh run.sh 테스트

모델의 말뭉치 BLEU 점수를보고하십시오. 21보다 커야합니다.

- (j) (3 점) (작문) 수업에서 내적 주의력, 곱셈 적 주의력, 그리고  
부가적인주의. 곱셈주의에 비해 내적주의의 한 가지 장점과 단점을 설명하십시오. 그런 다음 첨가제의 장점과 단점을 설명하십시오.

곱셈 적주의에 비해주의. 다시 말해, 내적 관심은 이자형  $t, i = \text{에스 } \epsilon_l$

$\epsilon_l h_{i-1}$ 는,

곱셈주의는 이자형  $t, i = \text{에스 } \epsilon_l$

$\epsilon_l \square \text{ 나}$ 는, 그리고 부가적인주의는 이자형  $t, i = V \epsilon_l / \tanh (W_1 h_{i-1} + W_2 \text{에스 } \epsilon_l)$ .

## 2. NMT 시스템 분석 (30 점)

- (a) (12 점) 여기서 우리는 출력에서 발견 한 일련의 오류를 제시합니다. <sup>2</sup> NMT 모델 (방금 훈련 한 것과 동일합니다.) 스페인어 소스 문장, 참조 (예 : '금') 영어 번역 및 NMT (예 : '모델') 영어 번역의 각 예에 대해 다음을 수행하십시오.
1. NMT 번역에서 오류를 식별합니다.
  2. 모델이 오류를 만들었을 수 있는 가능한 이유를 제공하십시오 (특정 언어 구조 또는 특정 모델 제한으로 인해).
  3. 관찰 된 오류를 수정하기 위해 NMT 시스템을 변경할 수 있는 한 가지 가능한 방법을 설명합니다. 오류에 대한 가능한 수정 사항이 두 개 이상 있습니다. 예를 들어 숨겨진 레이어의 크기를 조정하거나주의 메커니즘을 변경할 수 있습니다.

아래는 위에서 설명한대로 분석해야 할 번역입니다. 어휘를 벗어난 단어에는 밑줄이 표시됩니다. 이러한 질문에 답하기 위해 스페인어를 몰라도 됩니다. 영어 만 알면됩니다! 이 질문에 나오는 스페인어 단어는 대부분 정렬을 볼 수 있을 정도로 영어와 비슷합니다. 일부 단어에 대해 확실하지 않은 경우 Google 번역과 같은 리소스를 사용하여 찾아보세요.

나는. (2 점) 소스 문장 : *Aquí otro de mis* 가장 좋아하는 "La noche estrellada".

참조 번역 : 그래서 제가 가장 좋아하는 "별이 빛나는 밤"중 하나입니다.

NMT 번역 : 제가 가장 좋아하는 또 다른 작품 인 "별이 빛나는 밤"이 있습니다.

ii. (2 점) 소스 문장 : *Ustedes saben que lo que yo hago es escribir para los niños, y, de hecho, probablemente soy el autor para niños, ms ledo en los EEUU.*

참조 번역 : 아시다시피, 제가하는 일은 아이들을 위해 글을 쓰는 것입니다. 사실 저는 아마도 미국에서 가장 널리 읽히는 아동 작가 일 것입니다.

NMT 번역 : 제가하는 일은 아이들을위한 글입니다. 사실 저는 아마도 아이들을위한 작가 일 것입니다.

iii. (2 점) 소스 문장 : 내가 좋아하는 친구 – Richard Bolingbroke.

참조 번역 : 제 친구가 그렇게했습니다 – Richard Bolingbroke.

NMT 번역 : 내 친구가 행한 - 리처드 <UNK>

iv. (2 점) 소스 문장 : 독주 *tienes que dar vuelta a la manzana para verlo como una epifanía.*

참조 번역 : 당신은 단지 그것을 깨달음으로보기 위해 블록을 돌아 다니기 만하면됩니다.

NMT 번역 : 당신은 그것을 깨달음으로보기 위해 사과로 돌아 가야만합니다.

v. (2 점) 소스 문장 : 엘라 살브

ó mi vida al permitirme entrar al baño de la sala de

직업.

참조 번역 : 선생님 휴게실에있는 화장실에 갈 수있게 해주셔서 제 생명을 구했습니다.

NMT 번역 : 그녀는 내가 여자 화장실에있는 화장실에 갈 수있게함으로써 내 생명을 구했습니다.

<sup>2</sup> 데이터는 TED 회담 .

vi. (2 점) 소스 문장 : *Eso es más de 100,000 hectáreas.*

참조 번역 : 그것은 25 만 에이커 이상입니다.

NMT 번역 : 그것은 100,000 에이커가 넘습니다.

(b) (4 점) 이제 훈련 한 모델의 결과물을 살펴볼 시간입니다! 테스트 세트

문제에서 생성 된 모델 번역 1-i 위치해야 합니다 outputs / test outputs.txt.

확인하십시오 2 예 모델이 생성 한 오류의  $\sum$  찾은 두 가지 예는 이전 질문에서 제공된 예와는 다른 오류 유형과 다른 오류 유형이어야 합니다. 각 예에 대해 다음을 수행해야 합니다.

1. 스페인어로 원본 문장을 씁니다. 소스 문장은 en es data / test.es.
2. 참조 영어 번역을 작성합니다. 참조 번역은 en es data / test.en.
3. NMT 모델의 영어 번역을 작성합니다. 모델 번역 된 문장은 outputs / test outputs.txt.
4. NMT 번역에서 오류를 식별합니다.
5. 모델이 오류를 범했을 수 있는 이유를 제공합니다 (특정 언어 구조 또는 특정 모델 제한으로 인해).
6. 관찰 된 오류를 수정하기 위해 NMT 시스템을 변경할 수 있는 한 가지 가능한 방법을 설명합니다.

(c) (14 점) BLEU 점수는 NMT 시스템에서 가장 일반적으로 사용되는 자동 평가 지표입니다.

일반적으로 전체 테스트 세트에서 계산되지만 여기서는 단일 예에 대해 정의 된 BLEU를 고려합니다.

예. 소스 문장이 있다고 가정합니다. 에스, 세트  $\mathcal{C}$ /참조 번역 아르 자형  $t_1, \dots$ , 아르 자형  $\mathcal{C}$ /및 후보 번역  $\mathcal{C}$ . BLEU 점수를 계산하려면  $\mathcal{C}$ , 우리는 먼저 계산 수정  $\mathcal{C}$ -그램

정도  $\mathcal{C}$ /의  $\mathcal{C}$ , 각각  $\sum$  의  $n = 1, (2, 3, 4$ , 여기서  $\mathcal{C}$ 이다  $\mathcal{C}$ 에  $n$ -그램 :

$$p_{\mathcal{C}} = \frac{\prod_{n=1}^4 \frac{\min(\text{count}_{\mathcal{C}}(n\text{-gram}), \text{count}_{\mathcal{C}}(n\text{-gram}))}{\text{count}_{\mathcal{C}}(n\text{-gram})}}{\prod_{n=1}^4 \text{count}_{\mathcal{C}}(n\text{-gram})} \quad (15)$$

여기에서 각  $\mathcal{C}$ -후보 번역에 나타나는  $n$ -그램  $\mathcal{C}$ , 하나의 참조 번역에 표시되는 최대 횟수를 계산하며 표시되는 횟수로 제한됩니다.  $\mathcal{C}$  (이것은 분자입니다). 우리는 이것을 수로 나눕니다.  $\mathcal{C}$ -그램  $\mathcal{C}$  (분모).

다음으로 우리는  $\mathcal{C}$ -그램 BP. 허락하다  $\mathcal{C}$  길이  $\mathcal{C}$  그리고하자  $\mathcal{C}$  ( $r$ ) 될

참조 번역의 길이 { 가장 가까운  $\mathcal{C}$  ( $c$ ) (두 개의 동일하게 가까운 참조의 경우 번역 길이, 선택  $\mathcal{C}$  ( $r$ ) (짧은 것).

$$BP = \begin{cases} 1 & \text{만약 } \mathcal{C} \geq \mathcal{C} (r) \\ \frac{1 - \mathcal{C}}{\mathcal{C} (r)} & \text{그렇지 않으면} \end{cases} \quad (16)$$

마지막으로 후보자에 대한 BLEU 점수  $\mathcal{C}$  존경심으로 ( $\mathcal{C}$  )

$$BLEU = BP \times \prod_{n=1}^4 \mathcal{C} \quad (17)$$

어디  $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4$  합계가 1 인 가중치입니다. 여기서 로그는 자연 로그입니다.

$\mathcal{C}$  '오류'는 참조 번역과 일치하지 않는 NMT 번역이 아닙니다. 귀하의 의견으로는 NMT 번역에 문제가있는 것 같습니다.

$\mathcal{C}$  이 문장 수준의 BLEU 점수 정의는 문장 bleu () 기능 nltk Python 패키지. NLTK 함수는 대소 문자를 구분합니다. 이 질문에서 모든 텍스트는 소문자이므로 대소 문자는 관련이 없습니다.

나는. (5 점)이 예를 고려하십시오.

소스 문장  $s$ : el 사랑 todo lo puede

참조 번역 아르 자형<sub>1</sub>: 사랑은 항상 방법을 찾을 수 있습니다

참조 번역 아르 자형<sub>2</sub>: 사랑은 모든 것을 가능하게 한다

NMT 번역 씨<sub>1</sub>: 사랑은 항상 할 수 있다

NMT 번역 씨<sub>2</sub>: 사랑은 모든 것을 가능하게 할 수 있습니다

다음에 대한 BLEU 점수를 계산하십시오. 씨<sub>1</sub>과 씨<sub>2</sub> 허락하다  $\lambda_{n,n} = 0.5$  검색어:  $L_n \in \{1, 2\}$  및  $\lambda_{n,n} = 0$

$L_n \in \{3, 4\}$  ( 이것은 우리가 3 그래프와 4 그래프를 무시한다는 것을 의미합니다. 즉, 계산하지 마십시오  $\pi_{3,3}$  또는  $\pi_{4,4}$ . 언제

BLEU 점수 계산, 작업 표시 (예: 다음에 대한 계산 된 값 표시)  $\pi_{1,1}$ ,  $\pi_{2,2}$ ,  $len(c)$ ,  $len(r)$ 과  $BP$ ). BLEU 점수는 0과 1 사이 또는 0 사이로 표현 될 수 있습니다.

코드는 0에서 100까지의 척도를 사용하는 반면이 질문에서는 0에서 1 규모.

두 NMT 번역 중 BLEU 점수에 따라 더 나은 번역으로 간주되는 것은 무엇입니까? 더 나은 번역이라는 데 동의하십니까?

ii. (5 점) 하드 드라이브가 손상되어 참조 번역이 손실되었습니다. 아르 자형<sub>2</sub> 추천 해주세요

Pute BLEU 점수 씨<sub>1</sub>과 씨<sub>2</sub>. 이번에는 아르 자형<sub>1</sub> 뿐. 이제 두 NMT 번역 중 더 높은 BLEU 점수를 받는 것은 무엇입니까? 더 나은 번역이라는 데 동의하십니까?

iii. (2 점) 데이터 가용성으로 인해 NMT 시스템은 종종 단일 참조 번역과 관련하여 평가됩니다. 이것이 문제가 될 수 있는 이유를 (몇 문장으로) 설명해주시고.

iv. (2 점) 기계 번역의 평가 지표로 사람의 평가와 비교하여 BLEU의 두 가지 장점과 단점을 나열합니다.

## 제출 지침

이 과제를 GradeScope에 두 개의 제출물로 제출해야 합니다. 하나는 "과제 4 [코딩]"이고 다른 하나는 "과제 4 [작성]"입니다.

1. 실행 submit.sh 수집 Azure에서 스크립트를 생성하여 assignment4.zip 파일. 당신 사용할 수 있습니다 [scp](#) Azure와 로컬 컴퓨터간에 파일을 전송합니다.
2. 업로드 assignment4.zip "과제 4 [코딩]"에 대한 GradeScope 파일을 작성하십시오.
3. 서면 솔루션을 GradeScope에 "과제 4 [작성]"에 업로드합니다. 과제를 제출할 때 Gradescope의 제출 지침에 따라 각 문제의 모든 페이지에 태그를 지정해야 합니다. 제출물에 올바르게 태그가 지정되지 않은 경우 포인트가 차감됩니다.