

12022021 Week2 HW 보고서

Backgrounds

확률론적 관점에서,
기계번역이란 주어진 문장을 조건으로 하는 생성될 문장의 확률을 최대화하는 문장의 값을 구하는 문제이다.
우리는 문장을 토큰화함으로써 입력 단어열을 연속확률공간속의 입력요소로 볼 수 있고,
입력과 출력 사이의 조건부확률을 RNN을 이용하여 모델링함으로써 문장을 생성할 수 있다.
RNN만을 사용하면 이전의 입력에 대한 정보를 잘 반영하지 못한다는 단점이 있다.
이를 해결하기 위해서 Attention이 추가된 Seq2Seq 모델을 사용한다.

구현

라이브러리 관련

주어진 예제 코드를 colab에서 실행하기 위해서는, torchtext legacy를 사용하기 위해 이전 버전을 설치해야 한다.
`!pip install torchtext==0.11` 을 통해 설치할 수 있다.

tokenizer를 사용하기 위해서는

```
import locale
locale.getpreferredencoding = lambda: "UTF-8"
!python -m spacy download en
!python -m spacy download de
```

를 통해 locale 설정이 필요하다.

예제 코드 분석

예시코드는 encoder와, encoder의 overall output을 통해 context vector를 구하는 attention을 구현한 뒤,
decoder에서 attention을 활용하는 정석적인 논문의 구조대로 구현하였다.

그런데 evaluation loop 또는 encoder의 forward를 구현할 때, 정답 label 토큰이 함께 입력되므로 이대로는
inference가 불가능하다.

inference 구현

inference는 입력이 독어 문장임을 가정하고 구현하였다.

모델의 입력 순서를 생각하면, 입력받은 문장을 뒤집는 과정이 먼저 필요하다.

이후로는 입력받은 문자를 토큰화, 벡터(텐서) 화, 모델의 forwarding 후 출력 벡터를 다시 토큰과 문자열로 변환하
는 과정을 구현하였다.

```
def inference(sentence):
    tokenized_sentence = tokenize_de(sentence)
    numericalized_sentence = [SRC.vocab.stoi[t] for t in tokenized_sentence]
    sentence_tensor = torch.LongTensor(numericalized_sentence).unsqueeze(1).to(device)
    pred_probs = model(sentence_tensor, None, 0).squeeze(1)
    translation_tensor = torch.argmax(pred_probs, 1)
    translation = [TRG.vocab.itos[t] for t in translation_tensor][1:]
    return translation
```

그런데 이 경우에는 정답 레이블이 없으므로

seq2seq모델의 forward 시 None 레이블이 입력되면 입력과 같은 크기만큼의 더미 인풋을 생성하여 예외처리 하였다. teacher_forcing_ratio는 마찬가지로 추론시 0이어야 한다.

```
if trg is None:
    trg = torch.zeros((25, src.shape[1])).fill_(2).long().to(src.device)
    assert teacher_forcing_ratio == 0
```

결과 확인

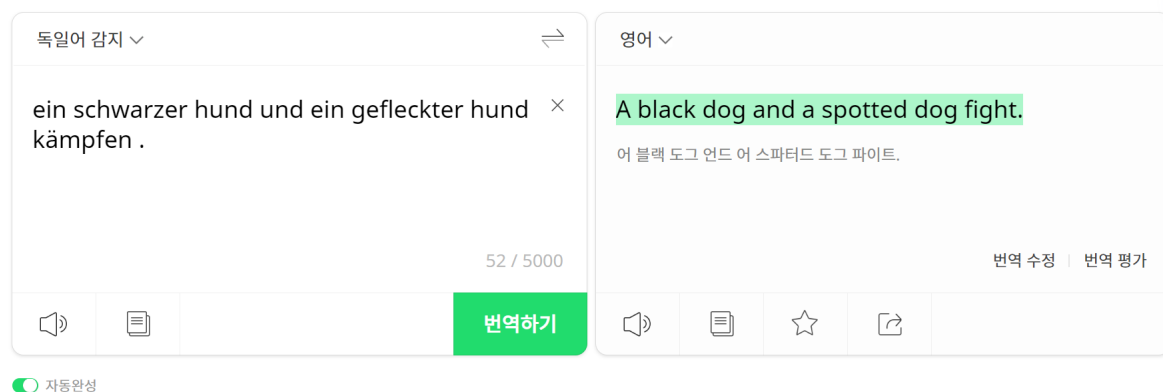
트레인셋 내의 입력 문장

ein schwarzer hund und ein gefleckter hund kämpfen .

a black dog and a spotted dog are fighting

에 대하여

papago는 정확한 출력을 보여준다



그러나 우리의 모델은 트레인셋의 데이터를 사용했음에도 불구하고 점박이라는 단어가 빠진 것을 확인할 수 있다.

```

1 example_data = ' '.join(vars(train_data.examples[12])['src'][:-1])
2 truth = ' '.join(vars(train_data.examples[12])['trg'])
3
4 print(example_data)
5 print(truth)

```

```

➞ ein schwarzer hund und ein gefleckter hund kämpfen .
a black dog and a spotted dog are fighting

```

```

1 inference([example_data])

```

```

['a',
 'black',
 'dog',
 'and',
 'a',
 'dog',
 'and',
 'dog',
 'is',
 'fighting',
 ',',
 ',',
 '<eos>',
 '<eos>',
 '<eos>',
 '<eos>',
 '<eos>',
 '<eos>',
 '<eos>',
 '<eos>',
 '<eos>',
 '<eos>',
 '<eos>']

```

학습에 사용하지 않은 데이터(어휘 셋에 없는) 인 경우 다음과 같이 출력이 정확하지 않은 모습을 보인다.

