Task 3 报告

0 前言

相关脚本如何运行详见repo rnn_jp2en目录下的README.md，环境相见repo rnn_jp2en目录下的requirements.txt。

1 RNN模型

代码详见repo rnn_jp2en/utils目录下的model.py。

EncoderRNN是一个简单的单向LSTM，其中embedding层在训练时会加载提前训练好的word vectors（在rnn_jp2en/train_rnn.py中可以看到加载了rnn_jp2en/embd/jp_embedding.pth）。

DecoderAttnRNN是一个带attention的单向LSTM，embedding层在训练时会加载提前训练好的word vectors（在rnn_jp2en/train_rnn.py中可以看到加载了rnn_jp2en/embd/en_embedding.pth），使用的attention为Luong attention。

在超参数的设置上，EncoderRNN和DecoderAttnRNN中LSTM的hidden state和cell state的长度均为256（config.hidden_size），embedding层中词向量长度为128（config.n_embd），DecoderAttnRNN设置的最长生成长度为72（config.max_len）。

> 注：词向量是在训练集上使用CBOW训练而得的。

2 Train/Validation/Test Set上的PPL, BLEU

数据集	PPL	BLEU
Train	1.94405	0.50636
Validation	6.37031	0.25040
Test	6.73010	0.24026

3 样例生成结果

日文原句：

```
case_1 = "私の名前は愛です"
case_2 = "昨日はお肉を食べません"
case_3 = "いただきますよう"
case_4 = "秋は好きです"
case_5 = "おはようございます"
```

模型英文翻译：

```
case_1 = "My name is my name."
case_2 = "I didn't eat meat yesterday."
case_3 = "You'd better be."
case_4 = "What like color is?"
case_5 = "Good morning."
```

4 分析

4.1 预训练词向量的使用

在最开始训练RNN时因为词向量没有训练好所以没有用训练好的embedding初始化Encoder和Decoder RNN的embedding层，在这时发现生成的句子（且不论正确性）比较割裂，翻译后的句子中经常出现一部分和另一部分像是来自于两个不同的句子一样的情况。

但是使用训练好的embedding之后，句子的连贯性变强了许多，即便翻译得和日文原句意思不同，得到的英语翻译也是一个连贯的句子。可能是因为用CBOW训练embedding时将上下文信息encode到了embedding当中，使得最终在RNN生成翻译时可以更好地考虑上下文，使得语句连贯。

4.2 模型翻译表现

有时模型可以很好地翻译日文原句（样例取自test set）：

日文原句	标准翻译	模型翻译	语法
コーヒーが好きです。	I like coffee.	I like coffee.	<名詞>が<形>です
まだ家にいるの？	Are you still at	Are you still at home?	まだ、〜の？

日文原句	标准翻译	模型翻译	语法
	home?		
トムはオーストラリア出身だと思う。	I think that Tom is from Australia.	I think Tom is from Australia.	～は～だと思う
私は彼の名前を知らない。	I don't know what his name is.	I don't know his name.	<名詞>は<名詞>を<動作>
トムは2013年にボストンを離れました。	Tom left Boston in 2013.	Tom left Boston in 2013.	<名詞>は<時間>に<場所>を<動作>

可以发现模型可以在不同复杂程度的句子上做到比较好的翻译效果，尤其是最后一个日文原句的语构结构很复杂但是模型竟然可以和标准翻译对上。此外，在倒数第二句中可以给出和标准翻译不同的（但是也同样是准确的）翻译，也体现了模型有一定的泛化能力。

但是有时会略微有点偏差：

日文原句	标准翻译	模型翻译	备注	语法
この店は免税店ですか。	Is this a tax-free shop?	Is this store a store shop?	‘免税’没有翻译出来	～は～ですか
ちょっと混乱しています。	I'm a little confused.	I'm a little nervous.	‘混乱’翻译错误	～しています
彼女は無実だ。	She's innocent.	She is a typist.	’無実’没有翻译出来	～は～だ

这些翻译出现略微的偏差但是句式和标准翻译基本保持相似的情况很有可能是因为训练集中这些token的出现次数比较少，比如”免税”在训练集上只出现了2次、”混乱”只出现了6次、”無実”只出现了8次，所以模型没有掌握其语义。

有些时候就是完全和原句的意思相背离：

日文原句	标准翻译	模型翻译
鳥は時たま、飛行機の障害となって事故の原因となることがある。	Birds sometimes cause accidents by getting in the way of airplanes.	The bird's almost run over the birds.

日文原句	标准翻译	模型翻译
私が黙っていたので彼女は余計に腹をたててしまった。	She got all the more angry because I kept silent.	She was angry with me when she was angry.
そろそろ失礼しなくては。	I have to leave now.	I'm sorry to have you done now.

大部分情况下就是因为句子成分复杂或者比较长，模型的翻译会很糟糕，有时也会因为多义或者隐晦的表达翻译出完全不对的结果，比如"失礼する"就是冒犯的意思，而这个词在不同的语境下可能会暗示不同的行为，造成最后翻译结果中对应的动词各不相同。

总而言之，目前这个模型体现出了一定的泛化能力，不过也有很多问题，经过分析我认为扩展一下训练集的数据的丰富程度有利于提高模型能力。