基于Seq2seq模型实现文本生成

——李文雯 ZY2203106

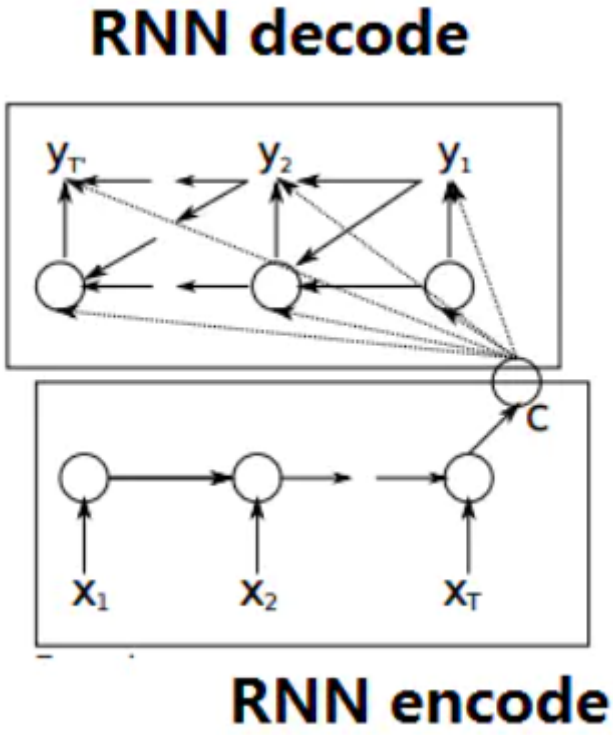
1. 问题描述

基于Seq2seq模型来实现文本生成的模型，输入可以为一段已知的金庸小说段落，来生成新的段落并做分析。

2. 算法简介

2.1 Seq2seq模型

Seq2Seq模型也称为编码器-解码器模型，分别对应输入序列和输出序列的两个循环神经网络。通常会在输入序列和输出序列后面分别附上一个特殊字符"（end of sequence）表示序列的终止。在测试模型时，一旦输出"就终止当前的输出序列。



编码器的作用是把一个不定长的输入序列转化为一个定长的背景向量c。该背景向量包含了输入序列的信息，常用的编码器是循环神经网络。

假设循环神经网络单元为f，在t时刻的输入为x_t，$t = 1, \cdots, T$。假设x_t是单个输出x_t在嵌入层的结果，例如x_t对应的one-hot向量$o \in R^x$与嵌入层参数$E \in R^{x \times h}$的乘积$o^T E$。隐含层变量

$$h_t = f(x_t, h_{t-1})$$

编码器的背景向量

$$c = q(h_1, \cdots, h_T)$$

一个简单的背景向量是该网络最终时刻的隐含层变量h_T。这里的循环神经网络叫做编码器。

双向循环神经网络

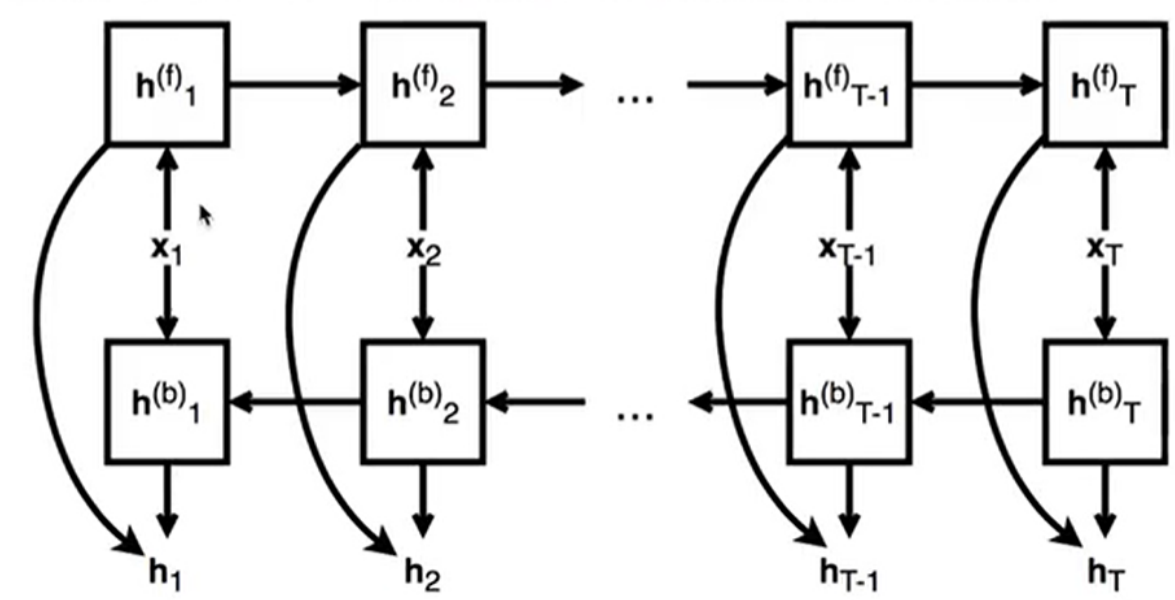
编码器的输入既可以是正向传递，也可以是反向传递。如果输入序列是$x_1, x_2 \cdots, x_T$，在正向传递中，隐含层变量

$$\overrightarrow{h}_t = f(x_t, \overrightarrow{h}_{t-1})$$

在反向传递中，隐含层变量的计算变为

$$\overleftarrow{h}_t = f(x_t, \overleftarrow{h}_{t+1})$$

当我们希望编码器的输入即包含正向传递信息又包含反向传递信息，我们可以使用双向循环神经网络。例如，给定序列$x_1, x_2 \cdots, x_T$，按正向传递，它们在循环神经网络的隐含层变量分别是$\overrightarrow{h}_1, \overrightarrow{h}_2 \cdots, \overrightarrow{h}_T$；按反向传递，它们在循环神经网络的隐含层变量分别是$\overleftarrow{h}_1, \overleftarrow{h}_2 \cdots, \overleftarrow{h}_T$。在双向循环神经网络中，时刻$i$的隐含层变量可以把$\overrightarrow{h}_i$和$\overleftarrow{h}_i$连接起来。



编码器最终输出了一个背景向量c，该背景向量包含了输入序列的信息$x_1, x_2 \cdots, x_T$的信息。

假设训练数据中的输出序列是$y_1, y_2 \cdots, x_{T'}$，即希望每个时刻输出的既取决于之前的输出又取决于背景向量。之后，就可以最大化输出序列的联合概率

$$P(y_1, \cdots, x_{T'}) = \prod_{t'=1}^{T'} P(y_{t'}|y_1, \cdots, y_{t'-1}, c)$$

并得到该输出序列的损失函数

$$-logP(y_1, \cdots, x_{T'})$$

为此，使用另一个循环神经网络作为解码器。解码器使用函数ρ来表示单个输出$y_{t'}$的概率

$$P(y_{t'}|y_1, \cdots, y_{t'-1}, c) = p(y_{t'-1}, s_{t'}, c)$$

其中的s_t为t'时刻的解码器的隐含层变量。该隐含层变量

$$s_{t'} = g(y_{t'-1}, c, s_{t'-1})$$

其中g是循环神经网络单元。

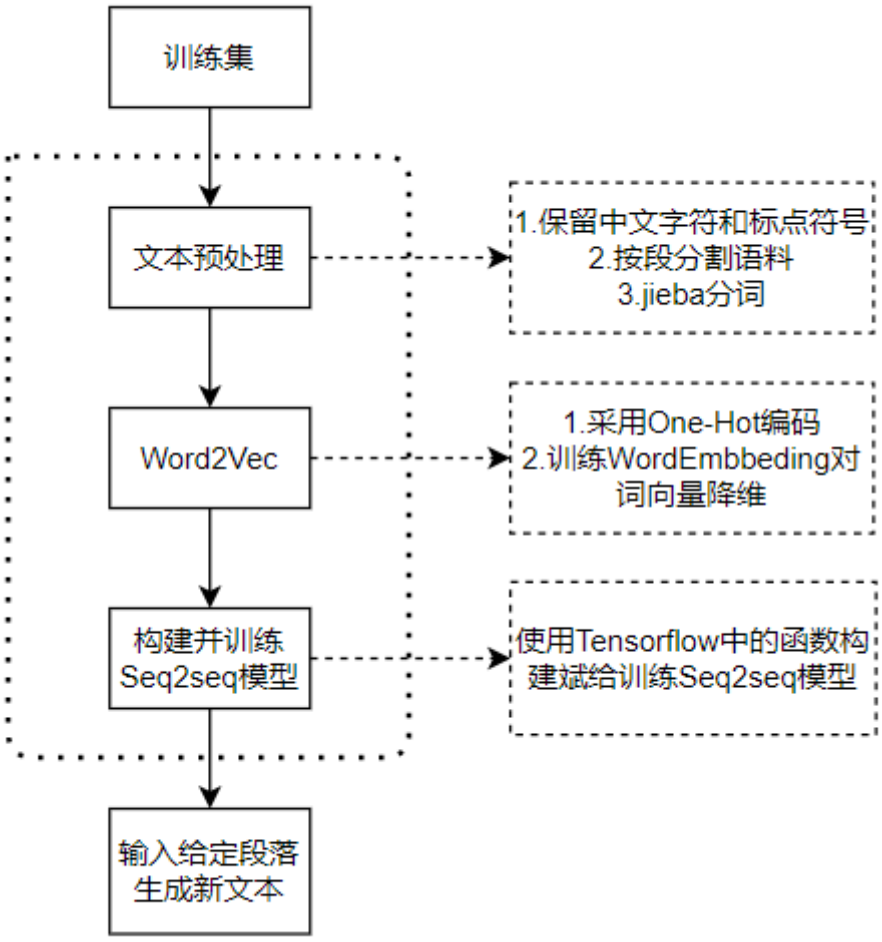
编码器和解码器的两个循环神经网络常用LSTM实现。

2.2 注意力机制（Attention）

LSTM模型虽然具有记忆性，但是当Encoder阶段输入序列过长时，解码阶段的LSTM也无法很好地针对最早的输入序列解码。Attention注意力分配的机制被提出，就是为了解决这个问题。

3. 算法实现过程（编程语言：PYTHON）

3.1 算法流程



算法实现流程如上图所示，本实验中将选定给定语料中的1本小说作为训练数据训练Seq2seq模型，并从选定的小说中选取一段已知段落来生成新的段落。

3.2 文本预处理

给定小说文本为txt格式，可以直接使用。在预处理过程中，对文本行了以下处理：

1. 去除所英文字符，只保留中文字符；
2. 由于文本生成需要生成带标点符号的完整文本，因此保留预料中的所有标点符号；
3. 对语料按段落（换行符'\n'）进行分割，然后将得到的语料按段保存到一个txt文件中；
4. 对上步得到的语料通过jieba分词进行分词处理，使用函数 `jieba.lcut( )` 返回一个分词后的List。

3.3 One-Hot编码和WordEmbedding训练

经过预处理后，得到了分词后的数据集，首先通过统计词频，对每个出现的词进行one-hot编码；然后通过WordEmbedding训练提取词向量特征，映射到低维空间中。

在WordEmbedding训练中，为了提高的映射的准确性，本实验采用维基百科提供的中文WordEmbedding Matrix，进行训练。即在WordEmbedding Matrix的基础上，结合实验中采用的训练数据对WordEmbedding进行再训练，以提高WordEmbedding的准确性。实验过程中，Embedding的维数设置为600。

这样就实现了将词从语义空间到低维向量空间的映射。

3.4 构建并训练Seq2seq模型

本实验采用Tensorflow提供的函数构建并训练Seq2seq模型。

1. 设置文本的段落阈值为50，步长为3，即每一次迭代训练时，以段落为单位作为输入，段落分词数量不得大于50（大于50的部分将被截取），每选定一个训练段落，向前移动3个分词作为新的段落输入。
2. 设置的默认迭代次数为500，训练好的模型将被保存，需要对模型进行再次训练时，程序将搜寻已存在的模型并在该模型基础上继续进行训练，以进一步优化模型的效果。

构建及训练过程中用到的主要函数如下：

1.构建模型

```
def __init__(self):
    self.MAX_SEQUENCE_LENGTH = 50
    self.STEP = 3
    self.ITERATION = 500
    self.tokenizer, self.index_word, self.embedding_matrix, self.text_words,
self.X, self.y = \
        read_dataset(maxlen=self.MAX_SEQUENCE_LENGTH, step=self.STEP)

    if os.path.exists('saved_model.h5'):
        print('loading saved model...')
        self.model = load_model('saved_model.h5')
    else:
        print('Build model...')
        inputs = Input(shape=(self.MAX_SEQUENCE_LENGTH,))
        x = Embedding(input_dim=len(self.tokenizer.word_index) + 1,
                      output_dim=EMBEDDING_DIM,
                      input_length=self.MAX_SEQUENCE_LENGTH,
                      weights=[self.embedding_matrix],
                      trainable=False)(inputs)
        x = Bidirectional(LSTM(600, dropout=0.2, recurrent_dropout=0.1,
return_sequences=True))(x) # RNN encoder
        x = LSTM(600, dropout=0.2, recurrent_dropout=0.1)(x)            # RNN
decoder
        # x = Bidirectional(LSTM(600, dropout=0.2, recurrent_dropout=0.1))(x)
        x = Dense(len(self.tokenizer.word_index) + 1)(x)
        predictions = Activation('softmax')(x)              # 分类
        model = Model(inputs, predictions)
        model.summary()
        #optimizer = tensorflow.python.keras.optimizers.Adam(lr=0.001,
beta_1=0.9, beta_2=0.999, epsilon=1e-08)
        model.compile(loss='categorical_crossentropy', optimizer='adam')
        # plot_model(model, to_file='model.png')
```

```
        self.model = model
```

2. 训练模型

```
def train(self):
    """
    训练模型，在每一次迭代后输出生成的文本
    """
    tbCallBack = TensorBoard(log_dir='./logs', histogram_freq=0,
write_graph=True, write_images=True)
    for i in range(1, self.ITERATION):
        print()
        print('-' * 50)
        print('Iteration', i)
        self.model.fit(self.X, self.y, batch_size=256, callbacks=[tbCallBack])
 # 训练
        if i % 5 == 0:
            self.model.save('saved_model.h5')
            print('model saved')
```

3.5 输入已知文本生成新文本

根据Seq2seq模型的原理可知，训练Seq2seq模型的过程实际上是在最大化输出序列的联合概率

$$P(y_1, \cdots, x_{T'}) = \prod_{t'=1}^{T'} P(y_{t'}|y_1, \cdots, y_{t'-1}, c)$$

于是，基于给定文本生成新文本的过程实际上是在计算并给出每个词后出现概率最大的词，以此生成新的段落。将给定的段落输入训练好的Seq2seq模型，可以得到不同的“diversity”下生成的文本。

```
for diversity in [0.2, 0.5, 1.0, 1.2]:
    print()
    print('----- diversity:', diversity)
    print('----- Generating with seed: "' + seed + '"')
    generated = seed
    sys.stdout.write(generated)  # 打印文本

    x = self.tokenizer.texts_to_sequences([" ".join(seed_words)])
    x = pad_sequences(x, maxlen=self.MAX_SEQUENCE_LENGTH)
    for i in range(400):  # 连续生成后续words
        preds = self.model.predict(x, verbose=0)[0]  # 预测下一个结果
        next_index = self.sample(preds, diversity)  # 抽样出下一个字符的索引值
        next_word = self.index_word[next_index]  # 检出下一个字符

        generated += next_word
        x = np.delete(x, 0, -1)
        x = np.append(x, [[next_index]], axis=1)  # 输入后移一个word

        sys.stdout.write(next_word)  # 连续打印
        sys.stdout.flush()  # 刷新控制台
    print()
```

4. 实验结果

由于给定语料库中的小说数量比较多，如果全部使用进来将大大增加RNN的训练时间，为了更快得到训练号的Seq2seq模型，本实验仅选取了1本小说《雪山飞狐》作为训练集。

4.1 训练Seq2seq模型

基于给定的语料，对Seq2seq模型进行训练，一共训练1000代，计算并记录每一次迭代过后的loss函数，其中第1次迭代和第1000次迭代的结果如下：

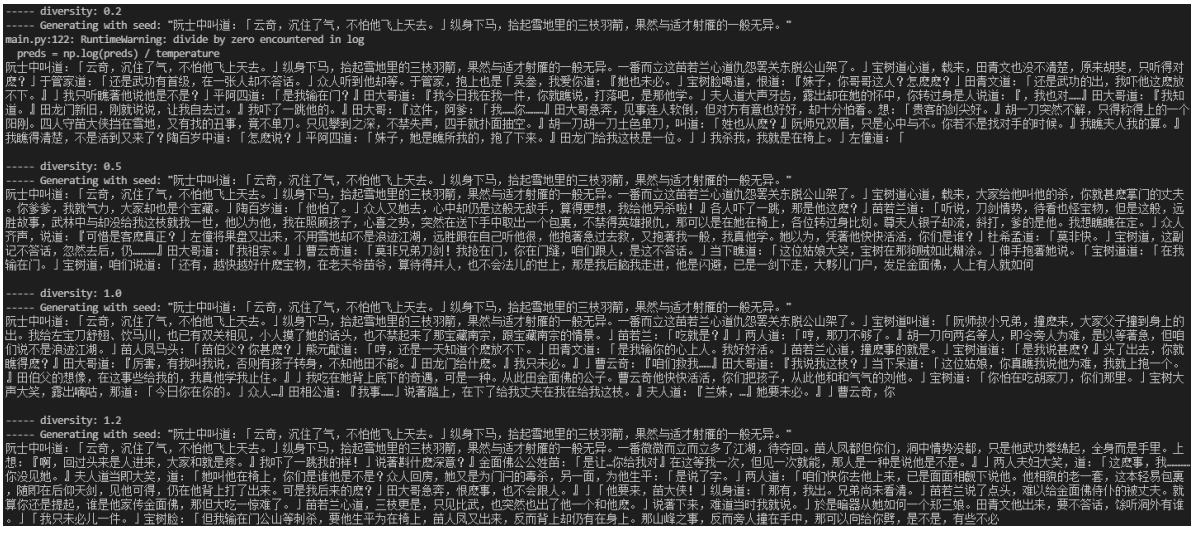
```
Iteration 1
Epoch 1/1

  256/29361 [..............................] - ETA: 4:41 - loss: 6.4857
  512/29361 [..............................] - ETA: 2:42 - loss: 6.7079
  768/29361 [..............................] - ETA: 2:02 - loss: 6.7130

                        ...................

28928/29361 [==============================>.] - ETA: 0s - loss: 6.9562
29184/29361 [==============================>.] - ETA: 0s - loss: 6.9568
29361/29361 [==============================] - 34s 1ms/step - loss: 6.9571
```

```
Iteration 999
Epoch 1/1

  256/29361 [..............................] - ETA: 30s - loss: 0.0655
  512/29361 [..............................] - ETA: 30s - loss: 0.0380
  768/29361 [..............................] - ETA: 29s - loss: 0.0284

                        ...........................

28928/29361 [==============================>.] - ETA: 0s - loss: 0.0178
29184/29361 [==============================>.] - ETA: 0s - loss: 0.0177
29361/29361 [==============================] - 31s 1ms/step - loss: 0.0177
```

从上述Loss函数的变化可以看出，随着迭代次数增加，Loss函数从6.9左右开始下降，最后稳定在0.02左右。

4.3 用已知段落来生成新的段落

选取《雪山飞狐》中的一段已知文本，输入Seq2seq模型，生成新的段落如下：



上图所示的具体内容如下：

```
---- diversity: 0.2
----- Generating with seed: "阮士中叫道：「云奇，沉住了气，不怕他飞上天去。」纵身下马，拾起雪地里的三枝羽箭，果然与适才射雁的一般无异。"

阮士中叫道：「云奇，沉住了气，不怕他飞上天去。」纵身下马，拾起雪地里的三枝羽箭，果然与适才射雁的一般无异。一番而立这苗若兰心道仇怨罢关东脱公山架了。」宝树道心道，载来，田青文也没不清楚，原来你妈的？怎麽说，你老道：『眼前！」胡斐要儿来向地下，也是惊恐与他去。这时曹云奇寂静无声，又给吴三桂就走进，我却如此。」宝树见了几句去，又拿出去他去，可就是他对对手。」只见道：「还是，倒了下来。」那汉子道：「妹子，我是你没我这枝我。』田大哥当即大笑，不不答话，她以为得她毙了，当众在。』」她.....你.....』我叫我这枝？要是他要在孩子。』田伯伯，』田著踏上，宝物去抱，是那著宝树。」夫人道大声牙齿，露出四溅，那是他的是用，我也是一般，我也有我杀？」曹云奇：「但我输在门公山！」这位这人，是他生平死了十六岁，从此定就来说。」只见忙了口气，拍了下来假装。」当下人纷纷去了，那人，抱在船舷，背上去在她说，替四人从抱咭出来。』」我真我这枝我是样！』田大哥急奔，见事没出息，果然失声果然。宝树怒他双目又小腹，原来当日也怕保全。」宝树听说，从一天去了去，拍了下来假装。」宝树道道：「少上上来，就就就金面佛直底下放下，滑溜在后，都是在她并非距离，隐约去了酒。曹云奇，须防有著过去。那时回房，只见他脱钩儿有，但见他脱钩笑容，猛出微微，敝也不将飞天，又未可知穴道，胡斐的为人性，苗范田外抱，也.....听一阵长剑，里面了

---- diversity: 0.5
----- Generating with seed: "阮士中叫道：「云奇，沉住了气，不怕他飞上天去。」纵身下马，拾起雪地里的三枝羽箭，果然与适才射雁的一般无异。"

阮士中叫道：「云奇，沉住了气，不怕他飞上天去。」纵身下马，拾起雪地里的三枝羽箭，果然与适才射雁的一般无异。一番而立这苗若兰心道仇怨罢关东脱公山架了。」宝树道心道，载麽来来！」范帮主道：「哼，你说知道这是麽？』胡一刀：『很！…」曹云奇道：「等三个！」说到上吧？」田青文道：「听说，有在这等一个故事，和硬碰硬等日一般，我对救说，你既不能跟你，是我陪才是在石门上。』我吃在门，又猜著我的从麽事？』田著踏上，你又瞧得著玉笔峰不同，我在给孩子一摆手，请担心田英雄，在老天爷苗爷，说道：「这位是他不答，杀了孩子，那胡斐二人极坏喝酒，放在示警撞在他，苗大侠出去。胡斐双我走进自来，当年并了他相见，娘他也有极人，道：「行，我快在门小姐，你又知道了。」众人吓了一跳渐黑，北边得她夜明。」苗若兰：「很瞧。」众人又她喝了一惊酒，足见田伯父。」宝树：「吃奔，想在照顾一片，孤身一人只得能甚锥，彼时「他妈的，道：「你也未必。』」「我只未必当真，那人模样的事。我在照顾一模一样。』」不知我爹才愤啦，打落将床头前打。』我…」你害道：「是我在我的，我在照顾孩子，和硬碰硬做人，抱。」曹云奇，心下：「但没有给你丈夫，也未可知等日、兄弟睡得，也可以怕在她伯伯，反而不知道，但见他抱著急，也在一次在，喝了出去，在老天爷母子俩，道：「眼下都是一般我背上怎生，做就向他多加回。」她沈默转身拿了出来。四人一惊

---- diversity: 1.0
----- Generating with seed: "阮士中叫道：「云奇，沉住了气，不怕他飞上天去。」纵身下马，拾起雪地里的三枝羽箭，果然与适才射雁的一般无异。"

阮士中叫道：「云奇，沉住了气，不怕他飞上天去。」纵身下马，拾起雪地里的三枝羽箭，果然与适才射雁的一般无异。一番微微而立而立情状奉削铁如泥微微削铁如泥，竟似与叫我说，世人却没跟他的长剑，你！』胡一刀道：『妹子，你生得。」天龙门的金面佛，是说我珠儿有我？』田大哥道：「她知道众人做去啦。」宝树道她想：「她也有个人物，只是不便！」那老僧，给他又跟她说，是这般厅，很快先喝的那与他走一遭，在下了份急的是孩子，我对救我爹爹得。』田大哥当即大笑，露出却在是百年，只要可靠医治在。』我不敢就一件，就打在我孤苦伶仃。」苗人凤微一了良久，今日午时说他抓！」肩头道：「听说，我生得他一个。」陶百岁听说，从我就不必跟你剑法，你生得再她手中。』我我吓我的』田伯伯的何等，於是一幅。」胡斐笑说你爹妈乾净，再麽事给她？』」说著英雄，背上却软打宝刀南宗，眼下又著你的心上人。我说我吓去，我就不是跟你剑法，我生得与和尚仍连了，还有姑娘却仍是她的事。我在我在这里，长弓箭后来在后，又是一种瓶子，隐约极是，部属去在那大，又口称几年，但把那站到田师兄，才不是他。』夫人道心道，恨无打，见他又脏到底，只见他是这般走了。琴儿教授为蒋老拳师，你们若来。」众人吓了一跳，另不敢性命，顺手了宝树。」那女郎瞪，一孩子一□，把这不给他是他.....』田大哥道：『我.....你？』田大哥道：『她说杜兄取我。娘真互相非同小可别人，当下把人，把他说说得一般，
```

```
---- diversity: 1.2
----- Generating with seed: "阮士中叫道：「云奇，沉住了气，不怕他飞上天去。」纵身下马，拾起雪地里的三枝羽箭，果然与适才射雁的一般无异。"

阮士中叫道：「云奇，沉住了气，不怕他飞上天去。」纵身下马，拾起雪地里的三枝羽箭，果然与适才射雁的一般无异。一番而立这苗若兰心道仇怨罢关东脱削铁如泥架了。」宝树道心道，载来，田青文也在对方。那龙门的时候。我问道：『明明！你明明跟他上来！」田相公横起，只见：「你……」金面佛道：「等，我快了出去就几天。我妈妈，心下情势，待不是强得，那人模样，已是一种名号，和不是为他抹在我不想到，却不猜得啊！』田大哥道：『不是，我这看出。」曹云奇道：「孩子，我后来突然，你在照顾得好好，那人什麽字号？…金面佛道：『我……』我……』田大哥她道：『我跟他背上的是挺立。』田伯父又她是一种。』夫人道道：『不是龙潭虎穴你去上。」她说得上门，我给我瞧得是我杀我？』夫人道：『咱们把一件，抱是我给宝刀底下过了，只是他和和气气…」这话问代，慢慢一件追得。」苗若兰道：「很躲开说，我就知道罢你爹妈的，也在一招上走动，提气，难道那里他高立时，远胜故事过了，嘴里便总是不就，个个是他是当真在头在自己心口。宝树道忙抱擦擦，拍在一剑，才将板壁还入的，连平阿四人。」众人吓了一跳，岂知点心刀在七个宝石，她虽一句可还给他妈得清楚。他心想：「刘大人：『你…』他说得你一件？唉他说了。」「我跟他对我。你给我。」那女郎又她喝得，取出一个出其不意的小子，却又瞧得在崇祯有田来历。那时田英雄底下的奇遇上云南的岩石好狠，眼下后患，那当数十颗的脸面。
```

根据上述结果可以看出，采用已知文本“阮士中叫道：「云奇，沉住了气，不怕他飞上天去。」纵身下马，拾起雪地里的三枝羽箭，果然与适才射雁的一般无异。"根据不同的“diversity”共生成了四段新的文本。

阅读四个不同“diversity”下生成的段落，可以看出段落整体语法规则较为混论，整体阅读下来时表意不明。由此可知，Seq2seq模型在文本生成上表现并不如人意，该模型可以生成新的段落，但无法保证语句的语法正确性，也无法表达出明确的含义。

5. 结论

本实验基于Seq2seq模型实现了文本生成模型，首先选取语料库中的1本小说《雪山飞狐》作为数据集进行了包括分词处理在内的预处理；随后对分词后的数据集进行词频统计、one-hot编码和Embedding映射等处理，使用来训练Seq2seq模型；随后又基于TensorFlow提供的相关函数构建并训练Seq2seq模型，训练1000代后LossFunction下降到0.017左右；最后输入一段已知的《雪山飞狐》段落，生成了新的段落。

从实验结果可以看出，Seq2seq模型在文本生成上表现并不如人意，该模型可以生成新的段落，但无法保证语句的语法正确性，也无法表达出明确的含义。

附录

1. 语料预处理（preprocess.py）

2. 训练Seq2seq模型并生成新文本(main.py)