文章编号：1003-0077（2018）00-0000-00

注意力的端到端模型生成藏文律诗

色差甲 华果才让 慈祯嘉措 柔特 才让加
（1. 青海师范大学藏文信息处理教育部重点实验室；
（2.青海师范大学藏文信息处理与机器翻译省级重点实验室,青海 西宁 810008）

摘要：文本自动撰写在自然语言处理中是一个重要的研究领域，可通过人工智能的方法来提升文本的生成结果。目前主流的生成方法是基于深度学习法，而该文中提出了一种基于注意力的端到端模型生成藏文律诗法。该方法构建在端到端的基础上，并无需任何人为的特征设置工作。基本框架是一个双向 LSTM 的编码-解码模型，在此基础上逐渐引入了藏文字嵌入、注意力机制和多任务学习法。实验结果表明,该文提出的方法在藏文律诗生成结果中其 BLEU 值和 ROUGE 值分别能达到 59.27% 、62.34%。

关键词：藏文律诗生成；字嵌入；注意力机制；编码-解码器

中图分类号：TP391　　　**文献标识码：**A

Generating Tibetan Poems with Attention Encoder-Decoder Model

SeChaJia, HuaGuoCaiRang, CeZhenJiaCuo, RouRe, CaiRangJia
(1. Qinghai Normal University Tibetan Information Processing Key Laboratory of Ministry of Education,
2. Provincial Key Laboratory of Qinghai Normal University of Tibetan Information Processing with Machine Translation. Xining, Qinghai, 810008)

Abstract : The automatic writing of text is an important research field in natural language processing. The text generation result can be improved by Artificial Intelligence. At present, the mainstream generation method is based on the deep learning method. In this paper, an end-to-end model based on attention is proposed to generate Tibetan poems. The method is built on an end-to-end basis and does not require any artificial feature set work. The basic framework is a BiLSTM encode-decode model. Based on this, Tibetan word embedding, attention mechanism and multi-task learning are gradually introduced. The experimental results show that the BLEU and ROUGE values of the method proposed in this paper can reach 59.27% and 62.34% respectively.

Key words: generating tibetan poems; embedding; attentiong; encoder-decoder

0　引言

文本生成在自然语言处理中是一项重要的研究内容，并具有三种生成类型，分别是：图像到文本、数据到文本以及文本到文本的生成。图像到文本的生成是计算机通过自动分析图像特征后生成相应的描述；数据到文本的生成是计算机通过自动分析数据特征后生成相应的说明；文本到文本的生成是计算机通过自动分析文本特征后生成相应的新文本。然而文本到文本的生成可按任务来区分多个生成类型，其中比较主流的有：机器翻译[1-3]、对话生成[4,5]、律诗生成[6,7]等，本文着重讨论藏文律诗生成。

在律诗自动生成的发展中，相关研究者使用过基于规则法、统计法以及深度学习法等，其中前两者方法的性能会受限于特征的选择和提取。这类方法在律诗自动生成中语法(必须遵守语法规则并且可读)、意义(每句的表达与主题有密切相关)和诗意(律诗必须具有诗意的特征，如节奏，音韵等)等[8]律诗标准的泛化能力相对于较弱。已使用过的方法有词语沙拉法、模板法[9]、

收稿收稿日期：2018-08-08；**定稿日期**：2018-08-08
基金项目：国家自然科学基金（61063033, 61662061）；教育部重点实验室项目（教技函[2010]52 号）；教育部“创新团队发展计划”滚动支持计划（IRT_15R40）；青海省重点实验室项目（2013-Z-Y17、2014-Z-Y32、2015-Z-Y03）；青海省科技厅项目（2015-SF-520）,国家社科基金（14BYY132）

遗传算法[10]和统计机器翻译法[11]等。目前，深度学习在自然语言处理中的各个领域中都备受关注，尤其是在神经网络机器翻译中取得优异的结果，同时在律诗自动生成中也逐渐取得理想的成绩。由中央电视台和中国科学院共同主办、中央电视台综合频道和长江文化联合制作的人工智能现象级节目《机智过人》中，清华大学的九歌自动生成的汉语诗歌震撼了所有嘉宾和观众。九歌的模型[7]建立在神经网络机器翻译模型的基础之上的，这类方法可以学习更长的诗句，同时在一定的程度上确保了前后语义的连贯性。

为了通过人工智能的方式让机器更好地理解和生成藏文律诗，该文结合了浩如烟海的藏文经典律诗和不受语种局限的深度学习法来生成全新的藏文律诗。基本框架是一个双向 LSTM 的编码-解码模型，在此基础上逐渐引入了藏文字嵌入、注意力机制和多任务学习。其中多任务学习是指基使用三个相同模块去承担不同的生成任务，第一个模块的任务是由藏文主题词来生成藏文律诗的第一句，第二个模块的任务是由第一句生成第二句，第三个模块的任务是由第一和第二句来生成第三句或者是由第二和第三句来生成第四句。结合三个模块后能生成更加流利的藏文律诗。

该文的后续部分分为：第一部分介绍了端到端模型的基础知识；第二部分重点阐述该文所使用的模型；第三部分给出了详细的实验结果及分析，并对研究语料的整体情况作了介绍；文章最后对整个工作做了总结并介绍下一步的研究计划。

1 背景知识

1.1 双向 LSTM 模型

1997 年 Schuster 等[12]人提出了双向 RNN(BiRNN)模型，目的是为了解决单向 RNN 无法处理后文信息的问题，其基本思路是每个训练序列的前向和后向分别是两个 RNN，两者模型的输出经过某种运算后得出最后的输出。

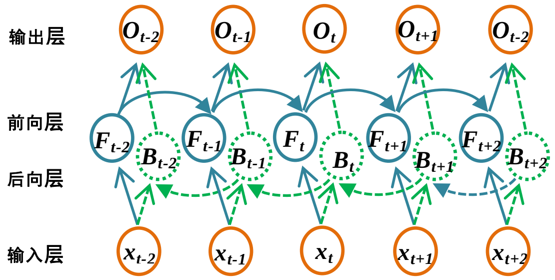

图 1 BiRNN 的结构图

图 1 中，隐藏层和输出层的计算公式如下：

$$F_t = f_F(W_F \cdot F_{t-1} + U_F \cdot x_t + b_F)$$
$$B_t = f_B(W_B \cdot B_{t+1} + U_B \cdot x_t + b_B)$$
$$o_t = f_o(V_F \cdot F_t + V_F \cdot B_t + b_o)$$

其中 W、U 和 V 是权重，b 是偏值，f 是激活函数。

双向 LSTM(BiLSTM)模型是结合 BiRNN 和 LSTM 的各个优点组成的新模型，可视为模型中的 RNN 单元替换成 LSTM 单元。BiLSTM 被广泛应用到自然语言处理的各项任务中后，都获得更为出色的结果，比如语音识别[13]、词性标注[14]和句法分析[15]等。

1.2 编码-解码模型

上述的 RNN 和 LSTM 都是一个将输入序列映射到一个等长的输出序列，但是将一个输入序列映射到一个不等长的输出序列的应用场景特别多，比如机器翻译、语音识别和问答系统等，其中输入序列和输出序列的长度不一定相同。Bahdanau 等人在 2014 年针对这个问题提出一个可变长度序列映射到另一个可变长度序列的架构的模型[16]，后来研究者们把这种构架的模型称之为编码-解码（Encoder-Decoder）模型或者序列到序列（sequence to sequence，简记为 seq2seq）模型。图 2 是编码-解码模型的展示图。

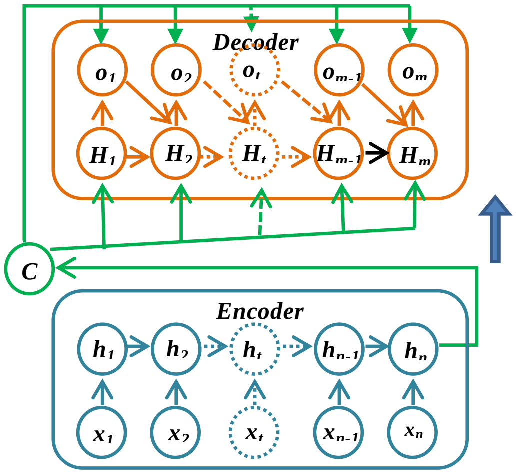

图 2 编码-解码模型的结构图

图 2 中，编码器和解码器是两个不同的 RNN 或者是两个不同的 LSTM。在编码器中也可用 BiRNN 或 BiLSTM，但解码器中就不能使用双向的模型，因为该模型的任务是通过前 t 个时刻的信息来预测 $t+1$ 刻的信息，所以无法使用 t 刻之后

的信息。把输入序列 $X=(X_1,X_2,...,X_k,...,X_{n-1},X_n)$ 编码成一个向量 C，即用编码器把所有输入序列的信息压缩成一个低维的向量。这个向量就是输入序列的量化表示，然后解码器利用该向量来生成最终的输出标签序列。解码器中隐藏层和输出序列的概率分别为：

$$H_t = f_H(W_H \cdot H_{t-1} + U_H \cdot o_{t-1} + b_H)$$
$$P(o_t \mid o_1,...,o_{t-1},c) = g(H_t, o_{t-1}, c)$$

结合上式和图 2 可知，编码器中每时刻的隐藏层跟前一刻的隐藏层和前一刻的输出值有关，而每时刻的输出跟当前的隐藏层、前一刻的输出值以及量化向量 C 有关。

1.3 标注意力机制

上述的编码-解码模型中，无法保证在 C 内包含所有输入序列的信息，因此会丢失一些输入数据的重要信息。针对这种问题，2014 年 Bahdanau 等人在编码-解码模型中引入了注意力机制（attention）来处理机器翻译[17]，其结果有很大的提升。基本思路是：从输入序列中捕捉显著与当前输出信息密切相关的信息，从而提高输出信息的质量，即相关性较高时赋较大的权重，反则赋较小的权重。注意力机制的最终目的是帮助编码-解码模型更好的学习输入序列和输出序列之间的相互关系，从而更好的表示输出序列的信息。在编码-解码模型中引入注意力机制后的展示图如下所示：

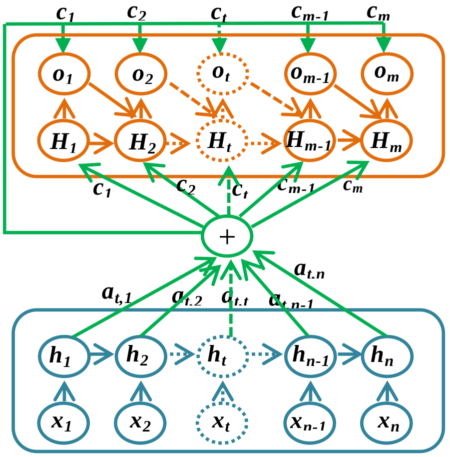

图 3 注意力机制的结构图

其计算公式如下所示：

$$P(o_t \mid o_1,...,o_{t-1},X) = g(o_{t-1},H_t,c_t)$$

$$H_t = f(H_{t-1},o_{t-1},c_t)$$

$$c_t = \sum_{j=1}^{T_x} a_{ij} \cdot h_j$$

$$a_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T_x} exp(e_{ik})}$$

$$e_{tj} = \omega(H_{t-1},h_j) = \upsilon_{\omega}^{T} \cdot tanh(w_{\omega} \cdot H_{t-1} + U_{\omega} \cdot h_j)$$

其中 g、f 和 ω 都是激活函数；X 表示所有的输入序列；c_t 表示所有输入序列对解码器中第 t 刻输出值的相关权重；e_{tj} 表示编码器的隐藏层中第 j 刻隐藏状态和解码器的隐藏层中第 $t-1$ 刻隐藏状态对解码器的隐藏层中第 t 刻输出值的相关权重。从中可以看出注意力机制的每个权重 c_t 由解码器中的前一刻隐藏状态和编码器的各个隐藏状态共同决定的；a_{tj} 表示编码器的隐藏层中第 j 刻隐藏状态对第 t 刻输出值的综合影响，并需要用 softmax 函数进行概率化。

2 注意力的端到端模型生成藏文律诗法

在这部分，将介绍如何用注意力机制和多任务学习法来进行端到端的联合建模。在藏文律诗生成中，虽然每个诗句的长度一样，但在该文中考虑了诗句间的语义连贯度，所以每个注意力的端到端模型中，输入和输出序列的长度需要设置不同长度。

2.1 字嵌入

在现有的基于神经网络的律诗生成中，输入向量（词向量或字向量）一般是从高斯分布中随机抽取的。该向量可视为一些符号化表示，因此几乎没有包含任何语义信息。训练模型时优化其它参数的同时优化该输入向量，而语义信息对于律诗生成而言是非常重要。

分布式表示法是针对独热表示法（one-hot）的缺点而提出的。首先需要将某种语言可视为一个固定维度的几何空间；然后通过训练把每个词映射成该空间中的一个点；最后通过两点之间的距离来判断每个词之间的语义相似性。通常分布式表示又称为字嵌入（embedding），或者称为字向量(word vecter)。Mikolov 等人详细介绍字向量的原理，并开源训练工具 Word2Vec[19]后，字向量在自然语言处理中运用很广泛，而且具有丰富的语义信息。该工具中有两种训练方式分别是：词袋模型（CBOW）和跳词模型（Skip-Gram）。该

文中将使用训练好的藏文音节向量作为输入特征训练藏文律诗生成模型。

2.2 单任务学习

藏文律诗生成的任务类似于机器翻译任务，是通过源句来自动生成目标句，即通过第一句来生成第二句，以此类推。但不同的是在机器翻译中源句和目标句的语种是不同的，比如藏汉翻译中源句是藏语，目标句是汉语，显然在律诗生成中源句和目标句是同语种。该文中借鉴基于神经网络的机器翻译（可简称 NMT）模型，因此特意构建了基于注意力的端到端模型来生成藏文诗句。该模型的主要任务是通过当前诗句来生成下一诗句，其展示图如下所示：

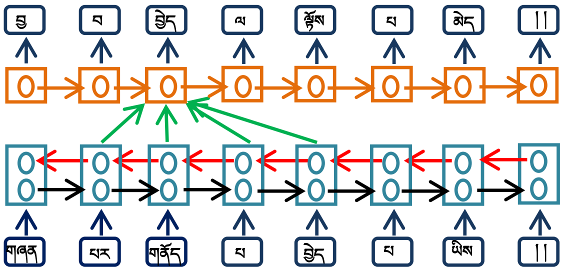

图 4 注意力的端到端模型的结构图

从图 4 中可以看出，该模型中输入和出入序列的长度一样，并且在编码器和解码器之间使用了局部注意力机制（luong attention）[18]，即不需要在全局的上下文信息中计算相关权重，而在局部的上下文信息中计算即可。同时在编码器中使用了双向的 LSTM，可在解码器中只能使用单向的 LSTM。输入的藏文音节向量是预先训练好的向量。该模型虽然可以通过三次循环后得到一个藏文律诗，但无法由主题词来生成第一句，并生成到第三句和第四句时会出现与主题漂移的现象，因此该模型只适合用于单任务，比如值适合用于第一句来生成第二句。

2.3 多任务学习

前面已简述了只用单个注意力的端到端模型时会出现的问题，因此针对这些问题需要引入多任务学习方法，也就是通过使用多个模型来承担不同的生成任务。该文中使用了三个注意力的端到端模型来承担三个生成任务，第一个模型的任务是由主题词来生成藏文律诗的第一句，该模型称之为诗字模型（Word Poems Module，简称 WPM）；第二个模型的任务是由第一句来生成第二句，该模型称之为诗句模型（Sentence Poems Module，简称 SPM）；第三个模型的任务是由第一、二句来生成第三句，或者是由第二、三句来生成第四句，该模型称之为诗块模型（Context Poems Module，简称 CPM）；由 WPM、SPM 和 CPM 组成的模型在该文中称之为藏文律诗生成模型（Generating Tibetan Poems Module，简称 GTPM）。

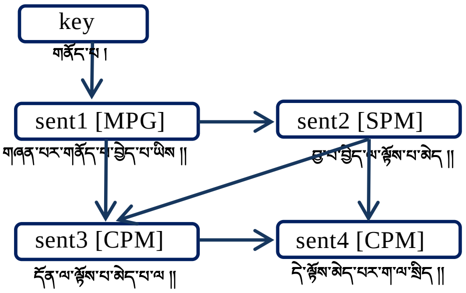

图 5 GTPM 的结构图

图 5 是由主题词”གཏོང་བ །”来生成完整的藏文律诗”གཞན་པར་གཏོང་པ་བྱེད་པ་ཡིས ॥ བྱ་བ་བྱེད་ལ་ལྟོས་པ་མེད ॥ དོན་ལ་ལྟོས་པ་མེད་པ་ལ ॥ དེ་ལྟོས་མེད་པར་ག་ལ་སྲིད ॥”的过程，而每首藏文律诗可以用集合<key, sent1, sent2, sent3, sent4>表示。从而得知训练 WPM、SPM 和 CPM 时所使用的训练数据不相同，其中 WPM 的训练数据的格式为<key, sent1>, SPM 的格式为<sent1, sent2>, CPM 的格式为<sent1+sent2, sent3>或者<sent2+snet3, sent4>。由于藏文律诗中不是每个律诗都有主题词，所以每个主题词是通过关键词抽取算法 textank1[1]来抽取的。

3 实验

在本实验中选用的评价指标有两种，分别为：常用于机器翻译的 BLEU 值和自动文档摘要的 ROUGE 值[20]。两者都是计算 n 元词组的共同出现的概率，呈现句子的词汇充分性和流利度。前者是基于精确地评价指标，后者是基于召回率的评价指标。由于藏文律诗中不会出现音节个数太多的词，所以其计算过程中语言模型被设为二元模型。

3.1 实验数据及其规模

训练神经网络模型时语料规模是一个很重要的因素，规模越大越能反映神经网络的计算性能。比如，神经网络机器翻译模型是通过上千万个句对来训练得出的，所以目前的翻译质量很流

---

[1] http://github.com/TB-SeChaJia/textrank

畅。而藏文律诗的获取方式有两种：第一是通过网络爬虫技术从藏文网站中可获取；第二是通过解析电子书籍来获取。从多个藏文网站和电子书籍等中收集了经典藏文著作的纯本文后，可通过藏文律诗抽取算法来获取其中的藏文律诗。该抽取算法如下图 6 所示：

```
抽取算法
输入：藏文文本
输出：藏文律诗
Text ← 读入文本
对 Text 的特殊音节后添加垂直符
Sents ← 用垂直符切分 Text
S1 ← Φ, S2 ← Φ
FOR  sent ∈ Sents  DO
    if  S1 ← Φ 且 sent 有两个垂直符
    then  S1 ← S1 ∪ sent
    else
        if  sent 的音节个数=S1 的音节个数，
            且 sent 有两个垂直符
        then  S1 ← S1 ∪ sent
        else
            if  S1 中的句子是否是 4 的倍数
            then  S2 ← S2 ∪ S1
            else  S1 ← Φ
Return S2
```

图 6 抽取算法的伪代码

抽取算法是基于藏文律诗的垂直符使用规律和诗句长度一致性来建立的，其中特殊音节后添加垂直符是指，若藏文句子中最后一个音节的后加字为”ག”时，则需要在该音节后添加一个垂直符”།”。目的是为了保证每个句子后面至少有一个垂直符，便于用该符号来切分句子。

通过上述的抽取算法，从已收集的藏文纯文本中共抽取了 381261 首藏文律诗,其中诗句的音节个数为 7 到 9 的律诗占 98%（有 373636 首），因此该实验的训练数据只使用了 373636 首藏文律诗。从中 WPM 的训练句对可抽取为 373636 个，SPM 的训练句对可抽取为 1119898 个，CPM 的训练句对可抽取为 746273 个。另外单独各收集了 500 个藏文律诗句对分别作为验证集和测试集。

3.2 实验参数设置

通过多次试验来优化参数，最终各个参数设置如下：模型训练次数设置为 100000；批量处理个数设置为 200；隐藏层神经单元个数设置为 256；隐藏层的层数设置为 4，由于是双向 LSTM，所以 2 层是正向的，另 2 层是反向的；字嵌入向量维度设置为 512；梯度截断值设置为 5；优化算法设置为随机梯度下降法（Stochastic Gradient Descent,简称 SGD）；注意力机制设置为局部注意力机制；学习率初始化为 0.8，同时被设为逐渐衰减法，即循环每 2000 次时衰减一次；为了防止神经网络过拟合，采用 Dropout 并设置为 0.6，即丢弃率为 0.4。表 1 是其它参数不变，只有模型不同的实验结果，调整每个超参过程中其模型的循环次数都设置为 20000：

表 1 RNN、GRU 和 LSTM 的对比结果

模型	验证集（%）		测试集（%）	
	BLEU	ROUGE	BLEU	ROUGE
RNN	3.60	2.80	3.60	2.80
GRU	5.60	3.90	6.00	4.00
LSTM	13.1.	16.70	15.10	19.20

表 1 中显而易见，在有限的训练次数（即 20000 次）内 LSTM 的 BLEU 值和 ROUGE 值都优于 RNN 和 GRU 的结果，因此在该实验中选用了 LSTM，而且是双向的 LSTM。其它的超参也是通过这种对比法来选取的。

3.3 实验结果及其分析

前面已经介绍了数据的规模及其分布情况，同时也选好了每个超参的取值，因此 GTPM 的最终的实验结果如表 2 所示：

表 2 GTPM 的对比结果

模型	**第一句（%）**		**第二句（%）**		**第三句（%）**		**第四句（%）**		**平均值（%）**	
	BLEU	ROUGE	BLEU	ROUGE	BLEU	ROUGE	BLEU	ROUGE	BLEU	ROUGE
SPM+word2rank	--	--	61.94	62.77	56.26	58.21	50.86	51.03	42.27	43.00
SPM+word2vec	--	--	76.50	79.53	63.84	65.84	54.59	55.75	48.73	50.28
CPM+word2rank	--	--	--	--	67.18	67.42	64.98	64.86	33.04	33.07
CPM+word2vec	--	--	--	--	75.89	76.23	69.54	69.59	36.36	36.46

GTPM+word2rank	27.43	35.29	61.94	62.77	65.06	64.87	52.69	51.70	51.78	53.66
GTPM+word2vec	34.76	42.93	76.50	79.53	65.88	66.58	59.92	60.31	59.27	62.34

表 2 中的’--’表示模型通过前一句来生成当前的句时，由于前一句的信息不足，会导致后续诗句的生成结果很糟糕，即可表示模型不适合生成该诗句的意思。比如，SPM 由主题词来生成第一句时，主题词成了 SPM 的输入数据，从而该词的音节个数远远少于 SPM 原本输入数据的音节个数，因此主题词进行向量化时需要补充很多零向量，或者是需要补充特殊向量（专门用来表示补充的向量）。显然补充后得到的向量矩阵中有很多没意义的信息，所以导致后续的生成结果。

经过对比表 2 的实验结果可知：

1) word2vec 和 word2rank 分别表示模型中使用了预先训练好的藏文音节向量和随机生成的音节向量。使用 word2vec 后的结果优于 word2rank 的结果。原因是 word2vec 中具有一定的语义信息，这对藏文律诗生成结果有很大的提升。

2）SPM 的生成结果稍微劣于 CPM 和 GTPM,同时生成到第三和第四句时诗句的流利度不如第二句。因为 SPM 的输入数据只有一个诗句，所以生成到第三或第四句时不仅缺乏上下文信息，而且会出现错误信息被传递的情况。通过主题词也无法有效生成第一句。

3) CPM 生成第三和第四句的结果最好，是因为该模型中使用了更多的上下文信息。通过前两句来生成后一句，比 WPM 和 CPM 所使用的上下文信息更多。同样该模型无法有效生成第一和第二句。

4) 总体来说，GTPM 的生成结果最理想。该模型使用了多任务学习法，即 WPM 负责生成第一句，SPM 负责生成第二句，CPM 负责生成第三和第四句后，藏文律诗的整体生成结果又很大的提升，而且平均 BLEU 值和 ROUGE 值分别能达到 59.27% 和 62.34%。这数据足以说明 GTPM 生成藏文律诗的结果在流畅度和忠诚度上效果很好。

GTPM 生成的部分藏文律诗如下所示：

ཐོས་བསམ།
དམ་པའི་ཆོས་ལ་ཐོས་བསམ་སྒོམ་པའི་ཚེ།།
སངས་རྒྱས་ཆོས་ཀྱི་སྐུ་ལ་མི་བརྟེན་པའི།།
སངས་རྒྱས་རྣམས་ལ་ཕྱག་འཚལ་མཆོད་པ་འབུལ།།
སྐུ་གསུང་ཐུགས་ཀྱི་དངོས་གྲུབ་སྩོལ་དུ་གསོལ།།

གནོད་པ།
གཞན་པར་གནོད་པ་བྱེད་པ་ཡིས།།
བྱ་བ་བྱེད་ལ་སྤོས་པ་མེད།།
དོན་ལ་སྤོས་པ་མེད་པ་ལ།།
དེ་སྤོས་མེད་པར་ག་ལ་སྲིད།།

这两首藏文律诗是 GTPM 通过主题词”ཐོས་བསམ”和”གནོད་པ”来生成的结果。从生成结果中可看出每个诗句的流利度很好，而且读起来会朗朗上口。说明 GTPM 不仅学会了藏文律诗中诗句长度一致性，而且节律也保持的很好。但不足之处是稍微缺乏诗句之间的语义连贯度。

4 总结与展望

该文的工作主要有以下四点：

1）该文提出了从藏文纯文本中提取藏文律诗的抽取算法，并使用该算法共收集了 373636 首藏文经典律诗。

2） 将注意力的端到端模型运用到了藏文律诗生成中，并结合多任务学习法，由三个模块分别承担不同任务来构建了 GTPM 模型。

3）在 GTPM 中引入预先训练好的藏文音节向量后，其生成结果有明显提高。

4）首先通过实验对比法来选择最优的超参，然后训练好 GTPM，最后通过实验结果分析得知，该模型的生成结果中 BLEU 值和 ROUGE 值分别能达到 59.27% 和 62.34%。能说明 GTPM 所生成的藏文律诗在诗句的流利度和忠诚度上效果很好。

该文中目前存在的问题有以下两点：

1）语料的精度上有一些瑕疵，比如部分藏文音节的部件出现了多录、少录和误录等现象。还有语料种类分布不均匀，比如，已收集的藏文律诗多数偏向于佛教文和民间谚语，缺乏其它类型的生成。

2）分析 GTPM 的生成结果可知，从句子层面来说生成结果很好，但从句子间的连贯度的方面来说，目前还欠缺了一点，仍存在可提升的空间。

下一步将计划使用更好的深度学习模型，如生成对抗网络（GAN）或者自注意力机制（self-attention）等，并需要收集更多的语料，进一步研究特定藏文律诗风格的生成法。

参考文献

[1] Wu Y, Schuster M, Chen Z, et al. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation[J]. arXiv preprint arXiv:1609.08144, 2016.

[2] Gehring J, Auli M, Grangier D, et al. Convolutional Sequence to Sequence Learning [J].1705.03122v2,2017.

[3] Vaswani A, Shazeer N, Parmar N, et al. Attention Is All You Need[J]:1706.03762, 2017a.

[4] Layla El Asri, Jing He, and Kaheer Suleman. A sequence-to-sequence model for user simulation in spoken dialogue systems. In Interspeech 2016, pages 1151–1155. ISCA. 2016.

[5] Ondřej Dušek, Filip Jurčíček. Sequence-to-Sequence Generation for Spoken Dialogue via Deep Syntax Trees and Strings[J]. arXiv:1606.05491. To appear in Proceedings of ACL. 2016:45-51.

[6] Zhang X, Lapata M. Chinese Poetry Generation with Recurrent Neural Networks[C]// Conference on Empirical Methods in Natural Language Processing. 2014:670-680.

[7] Yi X, Li R, Sun M. Generating Chinese Classical Poems with RNN Encoder-Decoder[J]. arXiv preprint arXiv:1604.01537,2017.

[8] Manurung H M. An Evolutionary Algorithm Approach to Poetry Generation[J]. University of Edinburgh, 2004.

[9] Tosa N, Obara H, Minoh M. Hitch Haiku: An Interactive Supporting System for Composing Haiku Poem[M]// Entertainment Computing - ICEC 2008. Springer Berlin Heidelberg, 2009:209-216.

[10] 周昌乐, 游维, 丁晓君. 一种宋词自动生成的遗传算法及其机器实现[J]. 软件学报, 2010, 21(3):427-437.

[11] Jiang L, Zhou M. Generating Chinese Couplets using a Statistical MT Approach.[C]// COLING 2008, International Conference on Computational Linguistics, Proceedings of the Conference, 18-22 August 2008, Manchester, Uk. DBLP, 2008:377-384.

[12] Schuster M, Paliwal K K. Bidirectional recurrent neural networks[M]. Signal Processing, IEEE Transactions on, 1997: 45(11), 2673-2681.

[13] Graves A,Jaitly N,Mohamed A.Hybrid speech recognition with deep bidirectional LSTM [C]//Proceedings of the 2013 IEEE Workshop on Automatic Speech Recognition and Understanding(ASRU).IEEE,2013:273-278.

[14] Huang Z,Xu W,Yu K. Bidirectional LSTM-CRF models for sequence tagging[J].arXiv preprint arXiv:1508.01991,2015.

[15] Kiperwasser E,Goldberg Y. Simple and accurate dependency parsing using bidirectional LSTM feature representations[J].arXiv preprint arXiv:1603.04351,2016.

[16] Cho K, Merrienboer B V, Gulcehre C, et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation[J]. Computer Science, 2014.

[17] Bahdanau D, Cho K, Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate[J]. Computer Science, 2014.

[18] Luong M T, Pham H, Manning C D. Effective Approaches to Attention-based Neural Machine Translation[J]. Computer Science, 2015.

[19] Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space[J]. Computer Science, 2013.

[20] 张瑾,王小磊,许洪波.自动文摘评价方法综述[J].中文信息学报,2008,22(3):81-88.

色差甲（1991—），博士，主要研究领域为藏文自然语言处理。
E-mail：bsichrb@outlook.com

华果才让（1984—），博士，主要研究领域为藏文自然语言处理。
E-mail：365332395@qq.com

才让加（1963—），博士生导师，主要研究领域为藏文自然语言处理，通讯作者。
E-mail：zwxxzx@163.com