

基于神经网络的集句诗自动生成¹

梁健楠², 孙茂松, 矣晓沅, 杨成, 陈慧敏, 刘正皓

(清华大学计算机科学与技术系, 清华大学人工智能研究院, 清华大学智能技术与系统国家重点实验室,
北京, 100084)

liangjn16@mails.tsinghua.edu.cn

摘要: 集句诗是中国古典诗歌的一种特殊体裁, 是从前人的诗篇中选取已有诗句, 再将其巧妙组合形成一首新诗, 是一种艺术的再创造形式。集句诗的生成要求集辑而成的诗不仅合辙押韵, 而且有完整的内容、连贯的上下文和新颖的主旨意境, 对创作者的知识储备和诗词鉴赏能力有极高的要求。本文基于计算机的海量存储和快速检索能力, 以及神经网络模型对文本语义较强的表示和理解能力, 提出了一种新颖的集句诗自动生成模型。我们的模型以数十万首古人诗作为基础, 利用循环神经网络(RNN)自动学习古诗句的语义表示, 并设计了多种方法自动计算两句诗句的上下文关联性。根据用户输入的首句, 我们的模型能够自动计算选取上下文语义最相关连贯的诗句进行集辑, 从而形成一首完整的集句诗。自动评测和人工评测的实验结果都表明, 我们的模型能够生成质量较好的集句诗, 远远超过基线模型的效果。

关键词 神经网络 中国古典诗歌 自动诗歌生成

中图分类号: TP391 **文献标识码:** A

Neural Network-Based Automatic Jiju Poetry Generation

Jiannan Liang, Maosong Sun, Xiaoyuan Yi, Cheng Yang,
Huimin Chen, Zhenghao Liu

(State Key Laboratory of Intelligent Technology and Systems, Institute of Artificial Intelligence,
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

Abstract: Jiju poetry is a special kind of Chinese classical poetry, which means the poet select different lines from poems generated by others to form a new poem and is a form of art recreation. The reformed poem should not only obey the structural and phonological constraints, but also have complete content, coherent context and original theme, which requires that the creator has abundant poetry knowledge and strong ability of poetry comprehension. Based on mass storage faster retrieval ability, and the strong semantic representation ability of neural networks, we propose a novel automatic Jiju poetry generation model. We store hundreds of thousands of human-authored poems and utilize Recurrent Neural Network (RNN) to learn the vector representation of each poetry line. Then we design different methods to measure the context coherence of two lines. Given a first line, our model can automatically calculate and select coherent lines to create a Jijue poem. Both automatic and human evaluation results show that our model can generate high-quality Jiju poems, greatly outperforming the baseline models.

Keywords: neural network; Chinese classical poetry; automatic poetry generation

1 引言

随着中华文化的复兴, 越来越多的人开始关注、学习中国古典文化, 而古诗词作为中国优秀古典文化的结晶, 在数千年的历史长河中对中华民族的发展有着深远的影响。

近年来, 中国古典诗歌自动生成的研究逐渐成为热点, 吸引了国内外越来越多研究者的注意。研究者们对中国古典绝句[1][2]和宋词[3][4]的自动生成任务提出了很多不同的模型和方法。这些研究大都集中在绝句和宋词的自动生成。本文着重关注一种特殊的中国古典诗歌形式——集句诗。集句作为一种别出机杼的诗歌再创造方式, 是对不同时期、不同诗人创作的不同诗篇中的已有句子进行选取并重新组合形成的新诗。诗句虽然是前人所写且各有其出处, 但、过渡流畅, 具备完整的内容和新颖的主旨意境, 浑然天成且无强拼硬凑之感。图 1 给出了一首明代著名文学家汤显祖所作的集句诗示例。

¹ **基金项目:** 国家社会科学基金 (13&ZD190)

² **作者简介:** 作者梁健楠 (1991-), 男, 硕士生, 主要研究领域为自然语言处理。

也曾飞絮谢家庭，(李山甫《柳十首》)
欲化西园蝶未成。(张泌《春夕言怀》)
无限春愁莫相问，(赵嘏《寄远》)
绿阴终借暂时行。(张祜《扬川法云寺双桧》)

图 1: 汤显祖《牡丹亭》中所作七言集句诗

集句诗的历史可以上溯到西晋而兴盛于宋代。集句创作需要创作者博闻强识，有大量的诗歌储备作为再创作素材。同时要求创作者有很强的诗词理解和鉴赏能力，只有充分理解前人不同作品中每一诗句表达的内容与意境，才能在对诗句进行有机的重新组合的同时而不破坏集成的整首诗的连贯完整性。在中国古代众多诗人中，只有王安石、苏轼、辛弃疾等诗文大家才有能力创作出高质量的集句诗。

随着硬件技术的发展，现代计算机具备了海量存储和快速检索能力，计算机能够“记忆”的诗作数量远远高于人类。近年来神经网络，尤其是基于循环神经网络 (Recurrent Neural Network, RNN) 的模型得到了极大的发展，并且在图像识别、语音识别、机器翻译等任务上取得了显著突破。RNN 能够自动从大规模语料库中学习到质量较高的句子语义表示。前人的工作已经表明，RNN 能够用于英文诗歌[5][6]和中国古典诗歌[2][4][7]的自动生成，并且能取得不错的效果。

本文提出了一种新颖的结合计算机的索引能力和神经网络的语义表示能力的自动集句诗生成模型。模型采取逐句生成的方式，根据用户输入的一句首句，依次选取古诗库中上下文语义、意境关联性最强的诗句作为当前生成句并将生成句作为输入诗句，以此类推，逐步生成一首完整的集句诗。首先我们设计了一套格律和韵脚检测方法，剔除诗库中不符合格律押韵要求的诗句，从而减小模型的搜索空间。然后基于整个古诗库，根据上下文的大小不同，我们训练了两个带 Attention 机制的 Neural Sequence-to-Sequence[8]模型，用以生成诗句的向量表示，通过句向量来衡量集句质量最重要的标准是整首诗的上下文连贯性和关联性。基于此，我们设计了三种不同的方法来计算一句诗句和已生成的上文多个诗句之间的关联性，用于当前候选诗句的选取。我们采用信息检索中常用的 MRR (Mean Reciprocal Rank) 指标用于评价集句诗生成模型。实验结果表明，我们的模型相对基于统计互信息的自动集句生成模型有较大提升。

综上，文本的贡献如下：

1. 我们首先提出集句诗的自动生成任务，并设计了相应的自动评价指标；
2. 基于神经网络，我们设计了一种基于 Sequence-to-Sequence 神经网络的自动生成集句诗模型。模型利用句向量自动学习诗句语义，测量语义相似性；
3. 自动评测和人工评测结果都表明，给定一句用户输入的首句，我们的模型能生成质量较高的集句诗。相对于基于统计互信息的自动集句生成模型有很大提升。

2 相关工作

诗歌生成是研究计算机自动分析、理解和使用人类语言的一个重要切入点。国内外在这一领域的研究已经持续了数十年。从方法上，相关的研究工作可以分为三个阶段：

早期的诗歌生成模型都是基于规则和模板的。例如，ASPERA 诗歌生成系统[9]和日文俳句生成系统[10]。

第二个阶段大约上世纪 90 年代开始。在这一阶段，统计机器学习方法陆续被用到诗歌生成上。Levy[11]和 Manurung[12]先后探索利用遗传算法来进行诗歌生成。与此同时，一些研究者也开始利用其他类型的文本生成模型。Yan 等人提出利用自动文本摘要的方式进行诗歌生成，将诗歌的生成过程看作是从整个古诗库进行摘要的过程[13]。另一个在诗歌生成任务上提升显著的模型是统计机器翻译(Statistical Machine Translation, SMT)模型。Jiang 和 Zhou 首先将 SMT 模型用于中文对联的生成[14]，取得了良好的效果。随后，He 进一步将此模型应用于中文绝句的生成[15]。

随着人工神经网络的发展，这一领域迈进了第三个阶段——基于神经网络的诗歌生成。大部分研究者都采用 RNN 或者其变体进行诗歌生成。Zhang 和 Lapata 最早将 RNN 应用到了中文绝句生成任务上[16]。他们使用一个卷积语句模型将生成的每句诗向量化然后压缩到一个上下文向量中，用以捕捉诗歌的上下文信息。但是他们的模型较为简单，为了提升诗歌的连贯性和通顺度，他们的模型还需要和两个额外的 SMT 特征以及一个基于 n-gram 的语言模型相结合。随后，Yan 提出了 polish 模型[17]，使用 RNN 进行绝句的生成。不同之处在于，Yan 的模型会对一首诗进行多次生成，每次生成时，会考虑上一次生成的诗歌，以模拟人类在创作诗歌时，不断修改润色的过程，从而提升诗歌的质量。Yi 等人第一次提出用结合了注意力机制的 Sequence-to-Sequence 模型[18]来进行诗歌生成[19]。他们针对绝句中不同的位置训练了不同的模型，以此来提升诗歌的上下文关联性。随后，在绝句生成任务上，Wang 等人提出了 Planning 模型[20]，该模型预先规划一首绝句里四个句子每一句对应的关键词，然后用对应的模型生成每一句。以往的模型生成的诗歌大多缺乏新颖性，因此 Zhang 等人提出了利用记忆网络来进行诗歌生成[21]。为了提升生成诗歌的新颖性，他们将数百首古人创作的诗歌存入一个外部的 Memory，以提供更丰富的外部知识来引导模型生成。同时，通过控制 Memory 中存储的诗歌风格，他们的模型也能实现对生成的诗歌风格的简单控制。Yi 等人同样提出利用 Memory 模型[22]。他们使用三个不同的 Memory 来模拟人类进行诗歌创作时的写作和记忆习惯。同时，他们提出了一种体裁向量，能够将诗歌的格律和结构信息与诗歌内容在一定程度上分离，从而使一个训练好的模型能生成不同体裁的诗歌，如绝句、宋词、现代诗等等。神经信息检索也为我们提供了另外的思路用以诗句相关程度的评判[25]。

上述模型的诗歌生成都集中在绝句和宋词这两种体裁上。这些模型由于要同时考虑诗歌的通顺性、上下文关联性、语义丰富性等等，难以做到兼顾所有方面，往往会顾此失彼。在这篇论文中，我们提出了一种全新的诗歌生成任务——集句诗自动生成。由于诗句是诗库中古人创作的现有句子，这些句子本身有较好的通顺性和语义丰富度。因此，集句诗自动生成模型可以忽略这些方面，而专注于把握不同语句之间的上下文语义、主题、内容的连贯性和一致性，这也是对诗歌这种的文学性文本最为关键的评价指标之一。

3 模型设计

3.1 任务及模型概述

如图 2，我们首先将集句诗自动生成的任务形式化描述如下：

设一首诗中，上文已经挑选出的诗句为 L_1, L_2, \dots, L_{i-1} ，简写为 $L_{1:i-1}$ ，设古诗库为 $\{S_n\}_{n=1}^N$ ，共包含 N 句古人创作的诗句。我们的目标是从 $\{S_n\}_{n=1}^N$ 中自动选取一句符合对应的言数、平仄、韵律且和 $L_{1:i-1}$ 在语义、主题和意境上连贯一致的诗句。根据用户输入的首句，不断重复该选取过程，直到选出特定数目的句子，则生成一首完整的集句诗。

我们定义一个评分函数 $R(L_{1:i-1}, S_n)$ ，用以计算诗库中的每一候选句和上文的连贯性得分。则集句诗的第 i 句选取为：

$$L_i = S_{n^*}, n^* = \operatorname{argmax}_n R(L_{1:i-1}, S_n) \quad (1)$$

从上述过程可以看出，平仄和韵脚可以通过规则限制进行筛选，一首集句诗的好坏则取决于式(1)中定义的评分函数的效果。我们希望找到合适的评分函数，使得生成的集句效果最佳。

以下将介绍本文设计的三种评分函数。

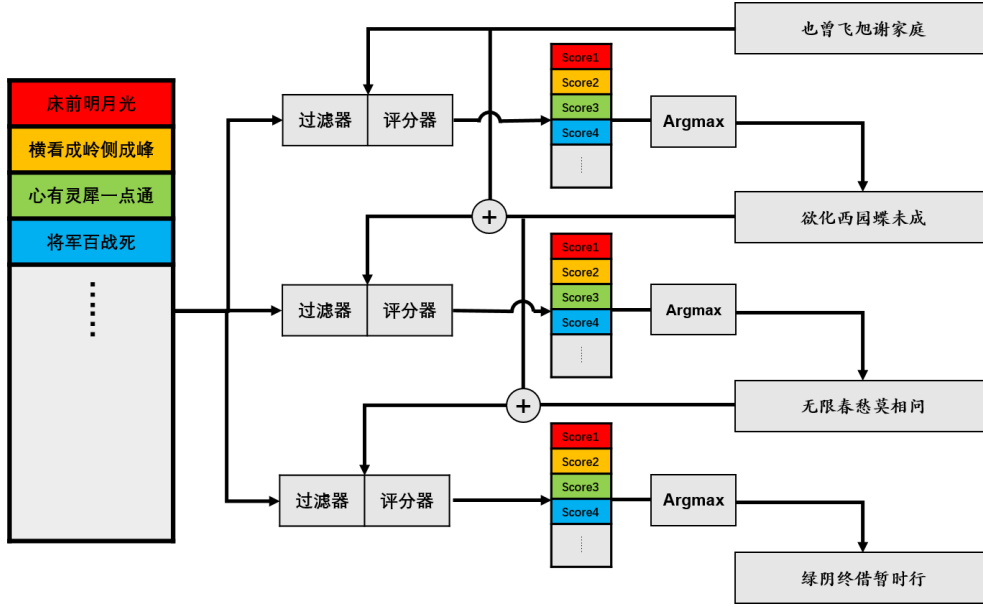


图 2：模型结构图。“+”为句子相加，即两个句子首尾相连拼接起来。

3.2 基于生成结果语义相似性的评分函数

不同于其他体裁(如绝句、宋词等)的生成，集句诗生成要求选取与上文连贯的古人诗句，而非依靠模型生成全新的诗句。前人的工作表明，带注意力机制的序列到序列模型 (Sequence-to-Sequence with attention mechanism) [8] 生成的诗句有不错的上下文关联性[7][18][20]。因此我们的第一种评分函数的思路是，第一步使用神经网络生成模型根据上文生成一句全新的当前诗句，第二步使用特定的方法从诗库中检索一句与该生成句语义最相似的古人诗句。

具体地，我们使用基于双向长短时记忆 (Long Short-Term Memory, LSTM) [23][24]，带 Attention 的编码-解码器 (Encoder-Decoder)。定义 X 为输入到 Encoder 的序列， $X = x_1 x_2 \dots x_{T_x}$ ； Y 为 Decoder 端生成的序列， $Y = y_1 y_2 \dots y_{T_y}$ ，其中 T_x 和 T_y 分别为 Encoder 和 Decoder 端的序列长度。给定 $L_{1:i-1}$ ，则 Decoder 端生成的诗句 G_i 计算如下：

$$s_t = LSTM(s_{t-1}, e(x_{t-1})) \quad (2)$$

$$h_t = LSTM(h_{t-1}, e(y_{t-1}), c_t) \quad (3)$$

$$p(y_t | y_{1:t-1}, L_{1:i-1}) = \text{softmax}(f(h_t, c_t, e(y_{t-1}))) \quad (4)$$

$$G_i = \hat{Y} = \text{argmax}_Y \prod_{t=1}^{T_y} p(y_t | y_{1:t-1}, L_{1:i-1}) \quad (5)$$

其中 s_t 和 h_t 分别为第 t 个时间步的 Encoder 和 Decoder 端的隐状态向量， $e(y_{t-1})$ 为对应的 word embedding， f 为线性变换层。 c_t 为 Attention 机制中的 attention 向量，具体计算参考[8]。

值得一提的是，上述公式(2)中，我们把 $L_{1:i-1}$ 拼接为一个长序列输入到 Encoder 中。参照[7]中的方法，当上文包含不同数目的诗句时，我们分别训练不同的生成模型来进行生成。一般，我们使用 1 句生成 1 句来产生诗歌的第二句，2 句生成 1 句来生成集句的诗歌的其他句子。

得到 Sequence-to-Sequence 模型生成的句子 G_i 之后，我们使用两种方法来度量诗库中的古人诗句 S_n 和 G_i 的语义相似性。

基于最大公共子串 (Longest Common Subsequence, LCS) 的相似性度量。第一种方法，我们假设两个句子之间匹配的子串越大，两个句子的语义相似性就越高。即：

$$R(L_{1:i-1}, S_n) = \text{LCS}(G_i, S_n) \quad (9)$$

在这里，我们考虑了实字和虚字对语义贡献度的不同，在匹配过程中，虚字匹配给予比实字低的分数。

基于句向量余弦距离的相似性度量。LCS 仅考虑了局部的子串匹配，从而可能丢失句子整体语义，同时无法处理近义词、同义词的匹配。LSTM 能够学习到较好的诗句向量表示[7]，因此我们通过两个诗句的句向量之间的余弦距离来度量其语义相似性。对一句诗句 S ，我们将其输入式(2)表示的 LSTM Encoder 中，然后其句向量计算为： $v(S) = \frac{1}{T} \sum_{t=1}^T s_t$ ，其中 T 为句长。则：

$$R(L_{1:i-1}, S_n) = \cos(v(G_i), v(S_n)) \quad (10)$$

3.3 基于上下文生成概率的评分函数

3.2 节中方法采取了首先生成新诗句 G_i ，然后再和古人诗句匹配的方式。在实际中，式(5)的概率最大值很难取到，因为无法遍历整个搜索空间。通常采用 greedy search 或者 beam search 进行搜索，但这无法保证生成的诗句 G_i 是在给定上下文的情况下最优的。因此，我们可以利用 Sequence-to-Sequence 模型直接对古人诗句和上文诗句的关联性进行建模。具体如下：

$$R(L_{1:i-1}, S_n) = P(S_n | L_{1:i-1}) = \prod_{t=1}^{T_n} p(c_t | c_{1:t-1}, L_{1:i-1}) \quad (11)$$

其中， T_n 为 S_n 的长度， c_t 为 S_n 中的每个字。

在式(11)中，我们实际度量的是，给定上文 $L_{1:i-1}$ ，神经网络模型生成某句古人诗句 S_n 的概率。对训练充分的 Sequence-to-Sequence 模型，我们在一定程度上可以认为，有更大概率生成的句子，其与上文的语义关联就更紧密。

3.4 多方法结合的评分函数

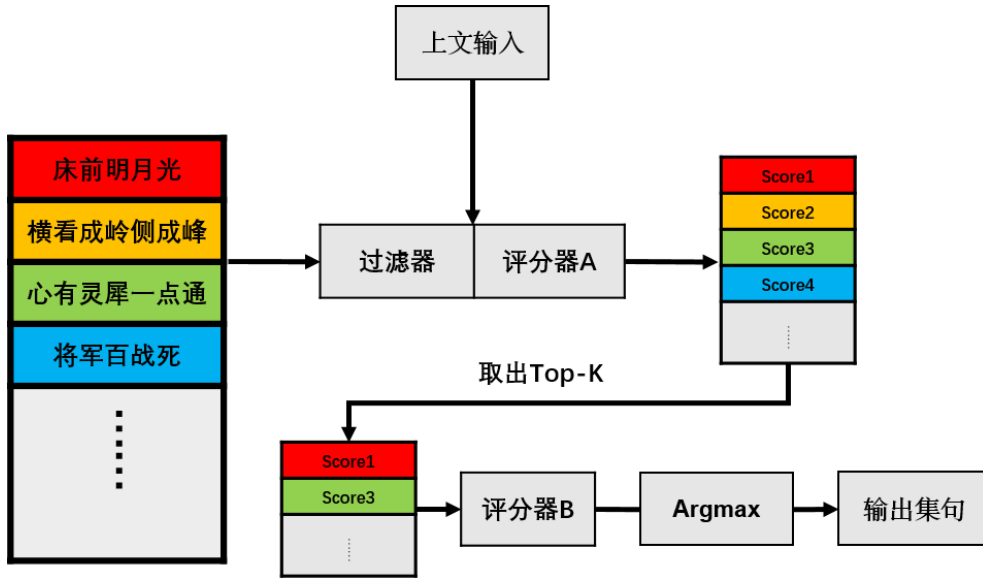


图 3：多方法结合评分结构图

在实际实验中，即使预先根据格律对诗库进行筛选，初步得到的候选句的规模 N 非常大，通常在数万的量级。使用式(11)计算时，由于 LSTM 隐状态的计算以及 Softmax 等操作耗时极大，在实际应用中难以接受。基于生成结果的语义相似性评分的方法计算速度较快，LCS 和句向量的余弦值的计算耗时较少，但是上下文关联性上表现欠佳。

因此，我们结合 3.2 和 3.3 的两种方法，先使用生成结果的语义相似性评分作为初步筛选，从古诗库中筛选出评分较高的候选子集。在候选子集中，再使用基于上下文生成概率的评分函数进行重排序，然后选取得分最高的一句。在第 4 节我们的实验结果表明，这种结合能取得生成速度和生成效果二者很好的平衡。

3.5 格律和押韵控制

在集句诗自动生成中，对古诗库诗句的格律和韵脚识别是非常重要的部分工作，识别错误将导致集句效果大打折扣。难点在于一句诗句格律不明显和一字多韵的情况。

格律的识别是以整首诗的格律来确定诗中某一句的格律，这样可以一定程度上容错，但不影响集句效果。对于一字多韵的情况，同样根据一整首诗的整体确定一句话的韵脚。此外，我们做了单句的韵脚识别器，使用句尾词统计方法来确定一句诗的韵脚——考虑句尾两个字或三个字的情况下，在古诗库中的出现次数，选取出现次数较多的韵作为韵脚。

4 实验

4.1 实验数据和设置

本次实验数据收集了唐代到近现代的五言和七言诗，共 389857 首诗，2537168 句诗句。

从中，随机选择 5000 首诗歌作为验证集，五言和七言各占 2500 首。除去测试集(测试集构造在 4.2.2 中详述)和验证集，其余诗歌作为训练集，用于训练神经网络模型。

本文的实验代码使用 python 和 tensorflow1.4 实现。在带 Attention 的 Sequence to Sequence 模型网络中，hidden size 为 512，word embedding size 为 256，激活函数为 tanh。采用的优化算法是 Adam，初始学习率为 0.001，训练过程中使用的 batch size 为 64。当模型在验证集上的损失函数收敛时停止训练。在我们的数据集上，约训练 10 个 epoch 左右能收敛。

4.2 实验与分析

4.2.1 评分函数的相关性分析

本文在生成句选取最佳匹配句的时候，使用了基于 LCS 和基于句向量的两种评分函数，以下将对两种评分函数的结果求 Spearman 相关系数。

这里选取了 20 个生成句，对生成句取基于 LCS 评分函数排序后的 Top100 候选句和基于句向量评分函数的 Top100 候选句，取这 200 句候选句并集求 Spearman 相关系数。实验得 Spearman 相关系数为-0.104。由此可见，使用基于 LCS 数和基于句向量评分函数的结果相关性并不大。

4.2.2 MRR 评测

MRR(Mean Reciprocal Rank)，即把标准答案在被评价系统给出结果中的名次取倒数后作为该答案的准确度，对所有标准答案的准确度取均值，以此来衡量评价系统的准确度。如下：

$$MRR = \frac{1}{|S|} \sum_{i=1}^{|S|} \frac{1}{rank_i}$$

将在古人创作的绝句中，我们分别选用第一句、第一二句、第一至三句作为输入，第二、第三和第四句作为对应的标准答案，另外以三种不同的筛选方法分别在数据集中选择 999 句同言数的诗句，给每个标准答案匹配对应的三组非标准答案集合，通过计算标准答案在 1000 句诗句中的准确度来评价不同的评分函数的效果。

所使用的三种筛选非标准答案的方法分别为：

1. 在数据集中随机选取同言数的诗句，记为测试集 1；
2. 在数据集中随机选取句同言数、同格律、同韵脚（若押韵）的诗句，记为测试集 2；
3. 在数据集中随机选取同言数、同格律、同韵律、与标准答案匹配两个字以上的诗句，记为测试集 3。

在基于句向量的评分函数中，我们设计了基于 1 句生成 1 句的 Sequence-to-Sequence 模型和基于 2 句生成 1 句的 Sequence-to-Sequence 模型生成，分别用这两个模型的 Encoder 生成诗句向量。基于 1 句生成 1 句的 Sequence-to-Sequence 模型是以古诗库中连续的两句诗句分为作为输入

		L2	L3	L4	Avg
五言	1 生 1 模型	0.143	0.031	0.064	0.079
	2 生 1 模型 (输入 1 句)	0.091	0.024	0.052	0.056
	2 生 1 模型 (重复输入)	0.086	0.041	0.047	0.058
七言	1 生 1 模型	0.049	0.025	0.057	0.044
	2 生 1 模型 (输入 1 句)	0.037	0.012	0.046	0.032
	2 生 1 模型 (重复输入)	0.045	0.037	0.059	0.047

表格 1: 基于句向量的评分函数 MRR 评测结果

和目标训练得到的模型。基于 2 句生成 1 句的 Sequence-to-Sequence 模型是以古诗库中连续的两句作为输入，紧接的第三句作为目标训练得到的模型。下面我们对以上两个模型使用测试集 1 进行 MRR 评测。原则上，2 生 1 模型需要输入 2 句得到句向量，这里我们使用想 2 生 1 模型 Encoder 端输入一句话和同一句话输入两次的方法，求得两种不同的句向量。

从表格 1 可以看出，使用 1 生 1 模型的 Encoder 得到句向量和 2 生 1 模型（重复输入）的 Encoder 得到句向量能有比较好的 MRR 值。

我们对基于句向量的集句模型和基于生成概率的集句模型进行 MMR 评测。此外，我们引入了基于互信息的集句模型作为基线。对于上文 L 和候选句 S_n ，计算它们之间的互信息作为排序的依据。互信息的计算公式为：

$$I(L, S_n) = \frac{1}{|L||S_n|} \sum_{l \in L} \sum_{s \in S_n} \frac{p(l, s)}{p(l)p(s)}$$

其中， l 和 s 分别为上文以及候选句中的单字， $p(l, s)$ 为古诗库中， l, s 同时出现在同一首诗的概率， $p(l)$ 为古诗库中， l 出现的概率。

由于基于 LCS 的集句模型的评分函数是非连续的，计算时无法得到一个准确的名次，因此在本实验中并没有讨论。

模型		MRR@测试集 1				MRR@测试集 2				MRR@测试集 3			
		L2	L3	L4	Avg	L2	L3	L4	Avg	L2	L3	L4	Avg
五言	MI	0.059	0.028	0.063	0.050	0.066	0.027	0.024	0.039	0.03	0.018	0.011	0.020
	Vec	0.143	0.031	0.064	0.079	0.075	0.016	0.053	0.048	0.029	0.011	0.016	0.019
	Prob	0.116	0.068	0.338	0.174	0.107	0.027	0.098	0.077	0.049	0.024	0.049	0.041
七言	MI	0.056	0.041	0.050	0.049	0.029	0.020	0.015	0.021	0.018	0.007	0.010	0.012
	Vec	0.049	0.025	0.057	0.044	0.022	0.013	0.047	0.027	0.017	0.004	0.030	0.017
	Prob	0.058	0.025	0.263	0.115	0.032	0.015	0.074	0.040	0.04	0.009	0.055	0.035

表格 2: MRR 评测结果

由表格 2 可见，基于生成概率(Prob)的集句模型计算得到的 MRR 值明显高于其他模型。基于生成概率(Prob)的集句模型和基于句向量(Vec)的集句模型在五言中的得分明显好于在七言中的得分。而在生成第三句时，我们的两个模型 MRR 值明显有降低。分析原因，是由于在绝句中，一般第三句都是起转折的地方，因此神经网络模型学习的效果会比其他位置的句子稍差。

4.2.3 集句效率分析

为了提高集句诗的生成效率，我们对古诗库的诗歌根据格律和韵进行划分建表，以便快速查找。对于 LCS 的计算，我们引入了倒排索引来大规模快速计算 LCS。对于句向量的计算，我们预先

计算每个句子对应的句向量并保存到内存中以提升计算速度，但由此也带来存储空间占用大的问题。以句向量是由 1024 维的 32 为浮点数组成为例，存储 250 万句诗句对应的句向量存储空间将达到 9.6GB。

根据实验统计，基于生成概率的集句模型平均集一首诗需要 2105.14 秒，基于 LCS 的集句模型平均集一首诗需要 62.83 秒，基于 LCS 和生成概率相结合的模型平均集一首诗需要 93.74 秒，基于句向量的集句模型平均集一首诗需要 4.91 秒，基于句向量和生成概率结合的模型平均集一首诗需要 5.99 秒。

4.2.4 人工评测

模型	五言	七言	总评
<i>Human</i>	3.45	3.55	3.50
<i>Jiju(LCS)</i>	2.40	2.37	2.39
<i>Jiju(LCS&Prob)</i>	2.69	2.65	2.67
<i>Jiju(Vec)</i>	2.44	2.40	2.42
<i>Jiju(Vec&Prob)</i>	2.73	2.67	2.79

表格 3：人工评测结果，平均评价得分

我们收集了 50 首古人创作的集句诗，其中五言绝句 25 首，七言绝句 25 首。对于每一首人创作的集句诗，我们以同样的首句作为输入，分别使用 Jiju(LCS)、Jiju(LCS&Prob)、Jiju(Vec)、Jiju(Vec&Prob) 四种集句模型进行集句生成，得出四首机器自动生成的集句诗。将古人创作的集句诗和自动生成的诗歌放在一起，组成 50 组，每组包含 5 首集句诗(组内顺序随机打乱)的人工测试集。

我们请了 5 中国古典诗词的专家对该人工测试集进行打分，根据集句的上下文语义的连贯性、主题和意境的是否鲜明进行打分，每首诗最高分 5 分，最低分 1 分。每位专家独立评分，相互之间不允许交流。同时我们要求每位专家在打分时集中在诗歌质量上，忽略“诗歌是机器还是人类创作”这一因素。

由表 3 可见，使用联合生成概率的集句模型平均比非联合生成概率的集句模型得分明显要高。而基于句向量的集句模型会比基于 LCS 的集句模型略好，但无显著差距。

4.2.5 个例分析

我们随机选取了一些不同模型生成的集句诗(表格 4)。对生成的七言集句诗，LCS 生成的结果中，上下文有明显的矛盾。第三句“春在江山无限好”写春景，第四句却写秋色，从而造成了季节上的矛盾。这是因为 LCS 仅仅句子考虑了局部子串的匹配。如第四句，生成模型生成的原句是“夜深风雨不胜愁”，此句与上文并无矛盾之处。LCS 进行匹配时，匹配了“夜深风雨”四个字，但因为仅仅考虑本句的局部，忽略了上文，所以匹配句中带来了“秋”这一矛盾。

对比之下，通过句向量相似度生成的集句诗整体质量上提升明显。第二句“不识花前人醉否”，点明了季节为春季。同时，第三句中“东风”紧扣“春”，“零落”也很好地承接了前一句的“花前”。第四句“拟于何处说来由”在语义上收束前三句，写出了诗人心中愁苦，于春花前借酒浇愁，但是这份苦痛却不知应该从何道起。

LCS&Prob 模型对诗句整体的一致性和连贯性有所提升。第二句“芳草青青”点明春季，第四句“对花”也紧扣“春”。由于生成概率本身对上下文关联性起到了约束作用，因此模型在基于 lcs 考虑局部语义相似时，也能在一定程度上避免上下文矛盾。此外，Vec&Prob 模型生成的集句诗比起单纯用句向量求近似的 Vec 模型，在整体的意境也有所提升。第一句描写闲适的春日景

	<i>Jiju(LCS)</i>	<i>Jiju(LCS&Prob)</i>	<i>Jiju(Vec)</i>	<i>Jiju(Vec&Prob)</i>
七言	闲云潭影日悠悠 醉倒不知何处接 春在江山无限好 夜深风雨总成秋	闲云潭影日悠悠 芳草青青古渡头 载酒有人来问字 对花何必更登楼	闲云潭影日悠悠 不识花前人醉否 多少东风零落恨 拟于何处说来由	闲云潭影日悠悠 底事东风不解愁 流水落花无限恨 可堪重倚仲宣楼
五言	花寒未聚蝶 密处少藏鸦 老去心犹壮 愁来不可遮	花寒未聚蝶 柳暗欲藏鸦 野径无人到 溪桥逐水斜	花寒未聚蝶 树密不遮亭 坐久无人问 唯寻相鹤经	花寒未聚蝶 柳暗欲藏鸦 野鸟啼空谷 江云散晚霞

表格 4：集句实例

色。在这样的景致中，第二句里诗人却感叹，春风是不能理解自己的愁绪的。第三句借落花凋零，流水匆匆形象地展现了自己内心的愁苦，第四句借用王粲的典故，收束全篇，点出自己愁苦的原因——怀才不遇。这首诗完整描述了一个郁郁不得志的诗人形象，比起 Vec 模型的结果，Vec&Prob 模型选择的句子意境更丰富，整首诗句子间的过渡也更好。

对生成的五言诗，Vec 模型的结果中，前三句关联性较好，但是第四句“唯寻相鹤经”和上文内容无关。Vec&Prob 模型则在上下文一致性上有明显提升。“花”、“蝶”、“柳”、“鸟啼”、“晚霞”，整首诗描绘了一副微寒的春晨，静谧野外的景致。使用 LCS 模型的结果中，前两句写景，后两句抒情，但景和情之间缺少联系，转折过于突兀，从而破坏了连贯性。LCS&Prob 的结果和 Vec&Prob 的结果类似，都是描绘了春季景色。区别在于，Vec&Prob 的结果中，第四句“晚霞”点明了时间是清晨，很好地映照了第一句“花寒”的“寒”字，交代了花寒的原因。但是 LCS&Prob 的结果没有体现出这一点，因此稍逊一筹。

从上述分析可以较为直观地看出，生成模型概率能紧密约束上下文关联性，对局部字串匹配带来的信息丢失也能起到一定的缓解作用。

5 结论与未来工作

本文探究了深度神经网络在集句任务中的应用。基于神经网络自动学习诗句的语义表示，我们设计了一种结合生成结果语义和上下文生成概率的评分方法，提出了一种高效率、高质量的自动集句生成模型。

试验结果表明，将生成模型计算候选诗句的生成概率作为集句候选的标准，可以有效地自动选择出效果比较好的集句。计算句向量近似可以在保证质量的前提下，提高集句的生成效率。

我们将继续探究诗句相似性的评价算法。当前神经信息检索方法通过核函数计算句对之间的相似度，我们在今后的工作希望能够使用相关方法更好的计算诗句之间的相似度。

参考文献

- [1] 何晶, 周明, 蒋龙. 基于统计的汉语格律诗生成研究[J]. 中文信息学报, 2010, 24(2).
- [2] Xingxing Zhang and Mirella Lapata. Chinese poetry generation with recurrent neural networks. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pages 670–680, Doha, Qatar, 2014.
- [3] 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).
- [4] Qixin Wang, Tianyi Luo, Dong Wang, and Chao Xing. Chinese song iambics generation with neural attention-based model. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, pages 2943–2949, New York, USA, 2016.
- [5] Marjan Ghazvininejad, Xing Shi, Jay Priyadarshi, and Kevin Knight. Hafez: an interactive poetry generation system. In Proceedings of ACL 2017, System Demonstrations, pages 43–48. Association for Computational Linguistics, 2017.

- [6] Jack Hopkins and Douwe Kiela. Automatically generating rhythmic verse with neural networks. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pages 168–178. Association for Computational Linguistics, 2017.
- [7] Xiaoyuan Yi, Ruoyu Li, and Maosong Sun. 2017. Generating chinese classical poems with rnn encoder-decoder. In Proceedings of the Sixteenth Chinese Computational Linguistics, pages 211–223, Nanjing, China.
- [8] Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In Proceedings of the 2015 International Conference on Learning Representations, San Diego, CA.
- [9] Pablo Gervás. 2001. An Expert System for the Composition of Formal Spanish Poetry. Springer London.
- [10] Xiaofeng Wu, Naoko Tosa, and Ryohei Nakatsu. 2009. New hitch haiku: An interactive renku poem composition supporting tool applied for sightseeing navigation system. In Proceedings of the 8th International Conference on Entertainment Computing, pages 191–196, Paris, France.
- [11] Robert P. Levy. 2001. A computational model of poetic creativity with neural network as measure of adaptive fitness. In Proceedings of the ICCBR-01 Workshop on Creative Systems.
- [12] Hisar Maruli Manurung. 2003. An evolutionary algorithm approach to poetry generation. Ph.D. thesis, University of Edinburgh.
- [13] Rui Yan, Han Jiang, Mirella Lapata, Shou-De Lin, Xueqiang Lv, and Xiaoming Li. 2013. I, poet: automatic chinese poetry composition through a generative summarization framework under constrained optimization. In Proceedings of the 23rd International Joint Conference on Artificial Intelligence, pages 2197–2203, Beijing, China.
- [14] Jing He, Ming Zhou, and Long Jiang. 2012. Generating chinese classical poems with statistical machine translation models. In Proceedings of the 26th AAAI Conference on Artificial Intelligence, pages 1650–1656, Toronto, Canada.
- [15] Jing He, Ming Zhou, and Long Jiang. 2012. Generating chinese classical poems with statistical machine translation models. In Proceedings of the 26th AAAI Conference on Artificial Intelligence, pages 1650–1656, Toronto, Canada.
- [16] Xingxing Zhang and Mirella Lapata. 2014. Chinese poetry generation with recurrent neural networks. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pages 670–680, Doha, Qatar.
- [17] Rui Yan. 2016. i, poet: automatic poetry composition through recurrent neural networks with iterative polishing schema. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, pages 2238–2244, New York, USA.
- [18] Xiaoyuan Yi, Ruoyu Li, and Maosong Sun. 2017. Generating chinese classical poems with rnn encoder-decoder. In Proceedings of the Sixteenth Chinese Computational Linguistics, pages 211–223, Nanjing, China.
- [19] Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In Proceedings of the 2015 International Conference on Learning Representations, San Diego, CA.
- [20] Zhe Wang, Wei He, Hua Wu, and Haiyang Wu, Wei Li, Haifeng Wang, and Enhong Chen. 2016. Chinese poetry generation with planning based neural network. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 1051–1060, Osaka, Japan.
- [21] Jiyuan Zhang, Yang Feng, Dong Wang, Yang Wang, Andrew Abel, Shiyue Zhang, and Andi Zhang. 2017. Flexible and creative chinese poetry generation using neural memory. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1364–1373. Association for Computational Linguistics.
- [22] Xiaoyuan Yi, Maosong Sun, Ruoyu Li, Zonghan Yang. 2018. Chinese Poetry Generation with a Working Memory Model. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence.
- [23] Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.
- [24] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- [25] LIU, Zhenghao, et al. Entity-Duet Neural Ranking: Understanding the Role of Knowledge Graph Semantics in Neural Information Retrieval. arXiv preprint arXiv:1805.07591, 2018..

梁健楠（1991—），男，清华大学硕士生，主要研究领域为自然语言处理。Email:liangjn16@mails.tsinghua.edu.cn