



中国科学院大学
University of Chinese Academy of Sciences

课程作业报告

基于双向 GRU 的自动写诗

作者姓名: _____ 陈子轩 2019E8015061013

学科专业: _____ 软件工程

所在单位: _____ 中国科学院大学计算机科学与技术学院

2020 年 5 月

基于双向 GRU 的自动写诗

陈子轩

摘 要

本实验属于自然语言生成 (Natural Language Generation, NLG) 任务, 具体过程为: 输入一个字 (或一段话), 输出由模型根据输入及诗的不同类型生成的诗句。在实验中, 设计了双向 GRU 结构的网络, 并对原始语料库进行筛选, 对不同的输入长度进行训练。在生成阶段, 设计不同规则的参数组合, 可以支持初始输入为藏头 hide、首字 first 及首句 sen, 并支持诗句数为 4 或 8 的五言或七言诗。在普通生成的基础上, 添加了平仄规则, 使得诗句更加上口通顺。最后分析了本实验的不足之处: 没有增加语义及情感因素, 使得生成的内容显得无意义或难以解释; 在平仄规则中对轻音的设计不合理。

1 介绍

- (1) 本实验要解决的问题为自动写诗, 属于自然语言生成 NLG 任务, 具体过程为: 输入一个字 (或一段话), 输出由模型根据输入及诗的不同类型生成的诗句。
- (2) 语料库: 原始数据为 poetry.txt, 其中包括 43030 首诗。
- (3) 使用双向 GRU 进行训练, 在预测阶段设计不同的诗的生成类型。

2 深度学习平台

TensorFlow == 2.0.0 + keras == 2.2.4 + plaidml == 0.7.0。

关于 plaidml 介绍见[Machine Learning/AI on macOS Catalina with Metal GPU Support](#)。这是一个能够使 A 卡也能用 GPU 加速的方法。使用时加入如下代码:

```
import plaidml.keras
plaidml.keras.install_backend()
import os
os.environ["KERAS_BACKEND"] = "plaidml.keras.backend"
import keras
```

3 实验过程与结果分析

3.1 语料库分析

原始数据为 poetry.txt 文件，其中包括 43030 首诗。经过筛选，形成文件 poetry4min.txt 和 poetry6min.txt。筛选规则为：删掉句子中出现不能正确表示文字（如某些古文现在文字编码中已被删去）的诗；或某些诗句因流传问题，部分文字已丢失，这样的诗亦被删掉。

- poetry4min.txt：每一行为一首五言诗，共 23220 首；
- poetry6min.txt：每一行为一首七言诗，共 16539 首。

实际训练及生成均使用 poetry4min.txt 和 poetry6min.txt。由于诗句的特殊性，不使用 jieba 模块、预训练词向量模型和停用词表，但删去出现频率低于 3 的字。最终分别形成 num2word 为 5336 对和 4544 对。

3.2 双向 GRU

GRU (Gated Recurrent Unit) [1] 是长短时记忆网络 (Long Short-Term Memory, LSTM) [2] 的一种变体，和 LSTM 同属于循环神经网络 (Recurrent Neural Network, RNN) 的改进模型。由于 RNN 在处理序列时具有严重的梯度消失问题，即越靠后的节点对于前面的节点感知能力越低。为了解决梯度消失问题，Hinton 等人提出了长短时记忆网络 LSTM。而 GRU 作为 LSTM 的变体，对序列数据处理同样非常适合，也是通过“门机制”来记忆前面节点的信息，依次解决梯度消失问题。

与 LSTM 不同的是，GRU 模型只有两个门，分别为更新门和重置门，即图 1 中的 z_t 和 r_t 。更新门用于控制前一时刻的状态信息被传入到当前状态中的程度，更新门的值越大说明前一时刻信息传入就越多。重置门用于控制遗忘前一时刻的状态信息的程度，重置门的值越小说明遗忘的越多。

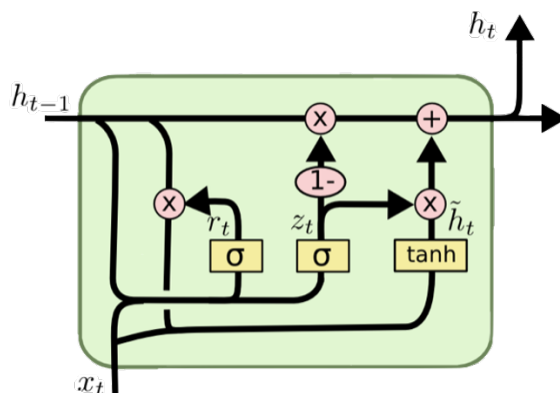


图 1: GRU 模块示意图

x_t 是输入数据， h_t 是 GRU 单元的输出， r_t 是重置门， z_t 是更新门， z_t 和 r_t 共同控制了从 h_{t-1} 隐藏状态到 h_t 隐藏状态的计算，更新门同时控制当前输入数据和先前记忆信息 h_{t-1} ，输出一个在

0~1 之间的数值 z_t , z_t 决定以多大程度将 h_{t-1} 向下一个状态传递。具体门单元计算为:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$

$$\tilde{h}_t = \tanh(W \cdot [r_t \times h_{t-1}, x_t])$$

$$h_t = (1 - z_t) \times h_{t-1} + z_t \times \tilde{h}_t$$

公式中 σ 是 Sigmoid 函数, W_z, W_r, W 分别为更新门, 重置门以及候选隐藏状态的权重矩阵。重置门控制 h_{t-1} 对结果 h_t 的重要程度, 当先前记忆 h_{t-1} 和新的记忆完全相关性较大时, 可以通过重置门发挥作用, 提升先前记忆的影响。根据重置门、更新门和隐含状态的计算结果可以通过公式得到当前时刻的输出 h_t 。

由于 GRU 只能得到前向的上下文信息, 忽略了后向的上下文信息。仿照双向长短时记忆网络 (Bidirectional LSTM, BiLSTM) [3], 将 GRU 进行双向组合。从前后向上同时获取上下文信息, 以提高特征提取的准确率。

3.3 网络设计及训练过程

整个网络模型结构如图 2 所示。其中 Embedding 层直接使用字的 one-hot 形式, 输出维度为 300; GRU 中 units = 128; 输出层使用 softmax 激活; 损失函数为交叉熵; 优化器选择 Adam(lr=0.005)。

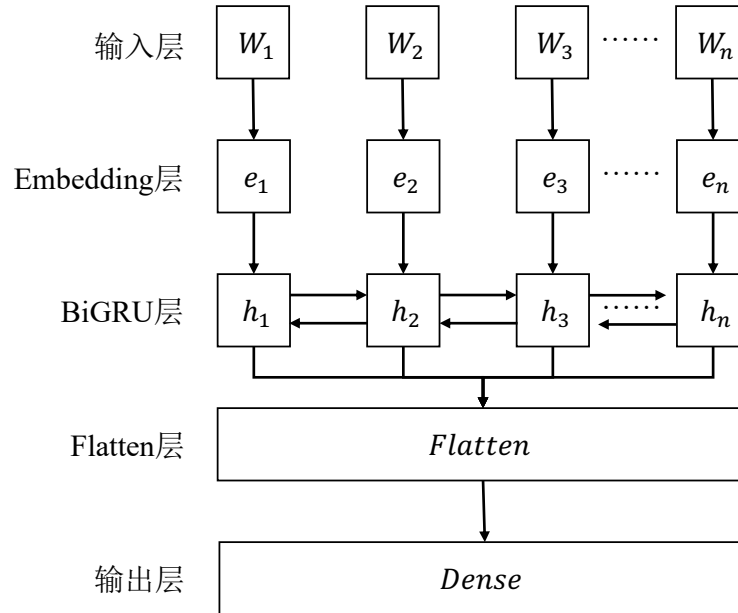


图 2: 网络模型结构

分别使用 poetry4min.txt 和 poetry6min.txt 作为语料库进行训练。为将无监督学习变为有监督学习, 设计为采用滑动窗口机制, 输入模型前 max_len 个字, 让模型预测第 max_len + 1 个字, 与诗句

中真实的第 $\text{max_len} + 1$ 个字做对比形成损失值。具体而言，以 `poetry6min.txt` 为例，滑动窗口长度为 7， $\text{max_len} = 7$ ，向模型输入前 6 个字，使模型预测第 7 个字，并与真实值为滑动窗口中第 7 个字对比形成损失值，训练过程如图 3 所示。

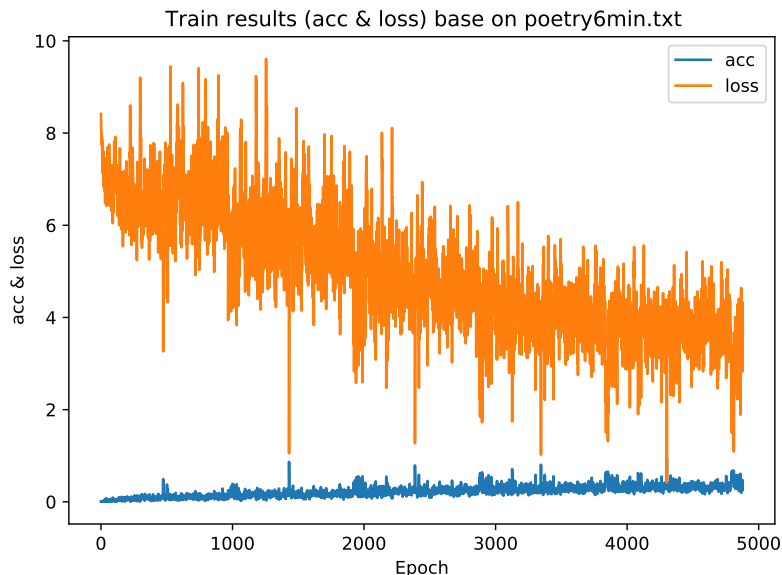


图 3: 训练过程图

3.4 生成诗句设计

本实验中生成诗句所需参数条件如表 1 所示。接下来分别进行说明。

表 1: 生成诗句所需参数条件

参数类型	诗句数 num_of_sentence	n 言诗 type	初始输入示例 text / char	描述
hide	4	5	测试一下	根据输入的四个字或八个字生成藏头诗
		7		
	8	5	测试一下自动写诗	
		7		
first	4	5	山	将输入的一个字作为诗的首个字，生成一首诗
		7		
	8	5		
		7		
sen	4	5	春风起秋云	将输入的一句话作为诗的第一句，生成一首诗
		7	春风不知何处月	
	8	5	春风起秋云	
		7	春风不知何处月	

- **hide**: 根据输入的 4 或 8 个字生成藏头诗。因模型训练时的输入为 max_len 个字，于是将初始输入中的第一个字 i^1 随机与语料库中的一句诗的后 $\text{max_len} - 1$ 个字 $r_1, \dots, r_{\text{max_len}-1}$ 拼接，形成第一个滑动窗口，也即第一次预测的输入为 $r_1, \dots, r_{\text{max_len}-1}, i^1$ ，经过预测得到第一个字 o_1^1 ；第二个滑动窗口为 $r_2, \dots, r_{\text{max_len}-1}, i^1, o_1^1$ ，经过预测得到第二个字 o_2^1 ；以此类推直至完成第一句的全部预测 $i^1, o_1^1, \dots, o_{\text{type}-1}^1$ 。剩下的 $\text{num_of_sentence} - 1$ 句诗，将前一句的后 $\text{max_len} - 1$ 个字拼接后一句的头字作为模型输入进行预测。最终形成的藏头诗为

$$\begin{bmatrix} i^1 & o_1^1 & \dots & o_{\text{type}-1}^1 \\ i^2 & o_1^2 & \dots & o_{\text{type}-1}^2 \\ \vdots & \vdots & \dots & \vdots \\ i_{\text{num_of_sentence}}^1 & o_1^{\text{num_of_sentence}} & \dots & o_{\text{type}-1}^{\text{num_of_sentence}} \end{bmatrix}_{\text{num_of_sentence} \times \text{type}}$$

- **first**: 将输入的一个字作为诗的首个字，生成一首诗。与 **hide** 一样，将初始输入中的第一个字 i^1 随机与语料库中的一句诗的后 $\text{max_len} - 1$ 个字 $r_1, \dots, r_{\text{max_len}-1}$ 拼接，形成第一个滑动窗口，也即第一次预测的输入为 $r_1, \dots, r_{\text{max_len}-1}, i^1$ ，经过预测得到第一个字 o_1^1 ；第二个滑动窗口为 $r_2, \dots, r_{\text{max_len}-1}, i^1, o_1^1$ ，经过预测得到第二个字 o_2^1 ；直到完成第一句的全部预测，形成第一句 $i^1, o_1^1, \dots, o_{\text{type}-1}^1$ ；第二句的预测使用第一句的后 max_len 个字为输入，预测得到第二句的第一个字 o_1^2 ；接下来使用第一句后 $\text{max_len} - 1$ 个字与 o_2^1 拼接作为输入，得到第二句的第二个字 o_2^2 ；以此类推，最终形成诗为

$$\begin{bmatrix} i^1 & o_1^1 & \dots & o_{\text{type}-1}^1 \\ o_1^2 & o_2^2 & \dots & o_{\text{type}}^2 \\ \vdots & \vdots & \dots & \vdots \\ o_1^{\text{num_of_sentence}} & o_2^{\text{num_of_sentence}} & \dots & o_{\text{type}}^{\text{num_of_sentence}} \end{bmatrix}_{\text{num_of_sentence} \times \text{type}}$$

- **sen**: 将输入的一句话作为诗的第一句，生成一首诗。因为输入的一句话的长度必然大于 max_len ，所以只需截取输入的后 max_len 个字作为第一次预测输入 $i_2^1, \dots, i_{\text{type}}^1$ ，经过预测得到第二句的第一个字 o_1^2 ，之后仿照 **first** 的预测方式，按照滑动窗口机制直至整首诗预测结束。最终形成诗为

$$\begin{bmatrix} i^1 & i_1^1 & \dots & i_{\text{type}-1}^1 \\ o_1^2 & o_2^2 & \dots & o_{\text{type}}^2 \\ \vdots & \vdots & \dots & \vdots \\ o_1^{\text{num_of_sentence}} & o_2^{\text{num_of_sentence}} & \dots & o_{\text{type}}^{\text{num_of_sentence}} \end{bmatrix}_{\text{num_of_sentence} \times \text{type}}$$

输出方式均为奇数行后添加逗号，偶数行后添加句号。

3.5 平仄设计

在普通生成诗句的基础上，为达到输出的诗句尽可能地满足汉语语法和表达习惯，加入平仄设计，即按照五言诗和七言诗的平仄规则进行限制生成。首先介绍平仄与声调的对应规则，如表 2 所

示。使用 `pypinyin` 模块获取单个字的声调，然后根据表 2 的映射关系定义该字的平仄，其中轻声直接定义为空平仄，在生成诗句中避免使用空平仄的字。

表 2: 本实验中平仄与声调对应规则

声调	平仄
0 (轻声)	(空)
1 (阴平)	平
2 (阳平)	
3 (上声)	仄
4 (去声)	

接下来介绍不同 `num_of_sentence` 及 `type` 下的平仄规则。该部分参考知乎：古诗词讲求平仄的道理是什么？。具体设计如表 5 所示。在 3.4 小节中的三种生成诗句设计中加入平仄规则，但显然并不是所有初始输入都能匹配到可满足的平仄规则，因此设计为若初始输入可以满足平仄条件，则按照平仄规则进行生成，否则按照普通生成。因此 `first` 中必然会按照平仄规则生成，而 `hide` 和 `sen` 会根据初始输入的不同决定是否按照平仄规则生成。需要说明的是，这里的平仄规则定义较为简单，在实际的五言或七言诗中存在许多不满足该平仄规则的情况。

在平仄规则下，生成过程比普通生成过程多了一步过滤。普通生成过程下预测的最后一步可以认为是返回 `top1`，在加入平仄规则后，预测的最后一步则为返回经过过滤的 `top1`。过滤规则为根据输入的 `max_len` 个字得到第 `max_len + 1` 个字应为平或仄，然后在预测值列表中过滤掉不为平或仄的字，最后在剩余的字中选择 `top1` 作为在平仄规则下的预测值。

以王贞白的《金陵》和刘禹锡的《酬乐天扬州初逢席上见赠》为原诗（如表 3、表 4 所示）选取部分字句作为初始输入，对比未加入平仄规则和加入平仄规则（如表 6 所示）。可以看到在增加了平仄规则后，所生成的诗句更符合诗句的读音习惯，更加上口通顺。

表 3: 王贞白的《金陵》

《金陵》
六代江山在，繁华古帝都。乱来城不守，战后地多芜。 寒日随潮落，归帆与鸟孤。兴亡多少事，回首一长吁。

表 4: 刘禹锡的《酬乐天扬州初逢席上见赠》

《酬乐天扬州初逢席上见赠》
巴山楚水凄凉地，二十三年弃置身。怀旧空吟闻笛赋，到乡翻似烂柯人。 沉舟侧畔千帆过，病树前头万木春。今日听君歌一曲，暂凭杯酒长精神。

表 5: 不同 num_of_sentence 及 type 下的平仄规则

诗句数 num_of_sentence	n 言诗 type	平仄规则
4	5	仄仄平平仄，平平仄仄平。平平平仄仄，仄仄仄平平。
		仄仄平平仄，平平仄仄平。平平平仄仄，仄仄仄平平。
		仄仄仄平平，平平仄仄平。平平平仄仄，仄仄仄平平。
		平平平仄仄，仄仄仄平平。仄仄平平仄，平平仄仄平。
		平平平仄仄，仄仄仄平平。仄仄平平仄，平平仄仄平。
	8	仄仄平平仄，平平仄仄平。平平平仄仄，仄仄仄平平。
		仄仄平平仄，平平仄仄平。平平平仄仄，仄仄仄平平。
		仄仄仄平平，平平仄仄平。平平平仄仄，仄仄仄平平。
		平平平仄仄，仄仄仄平平。仄仄平平仄，平平仄仄平。
		平平平仄仄，仄仄仄平平。仄仄平平仄，平平仄仄平。
7	5	平平仄仄仄平平，仄仄平平仄仄平。仄仄平平平仄仄，平平仄仄仄平平。
		平平仄仄平平仄，仄仄平平仄仄平。仄仄平平平仄仄，平平仄仄仄平平。
		平平仄仄平平仄，仄仄平平仄仄平。仄仄平平平仄仄，平平仄仄仄平平。
		仄仄平平仄仄平，平平仄仄仄平平。平平仄仄平平仄，仄仄平平仄仄平。
		仄仄平平平仄仄，平平仄仄仄平平。平平仄仄平平仄，仄仄平平仄仄平。
		仄仄平平平仄仄，平平仄仄仄平平。平平仄仄平平仄，仄仄平平仄仄平。
	8	平平仄仄仄平平，仄仄平平仄仄平。仄仄平平平仄仄，平平仄仄仄平平。
		平平仄仄平平仄，仄仄平平仄仄平。仄仄平平平仄仄，平平仄仄仄平平。
		平平仄仄平平仄，仄仄平平仄仄平。仄仄平平平仄仄，平平仄仄仄平平。
		仄仄平平仄仄平，平平仄仄仄平平。平平仄仄平平仄，仄仄平平仄仄平。
		仄仄平平平仄仄，平平仄仄仄平平。平平仄仄平平仄，仄仄平平仄仄平。
		仄仄平平平仄仄，平平仄仄仄平平。平平仄仄平平仄，仄仄平平仄仄平。
		仄仄平平平仄仄，平平仄仄仄平平。平平仄仄平平仄，仄仄平平仄仄平。
		仄仄平平平仄仄，平平仄仄仄平平。平平仄仄平平仄，仄仄平平仄仄平。
		仄仄平平平仄仄，平平仄仄仄平平。平平仄仄平平仄，仄仄平平仄仄平。

表 6: 未加入平仄规则和加入平仄规则后的生成诗对比

生成目标	未加入平仄规则	加入平仄规则
初始输入为 六繁乱战 的藏头诗	六尺三乐举，繁花看花声。 乱雁呈几度，战马光辉日。	六尺情常怀，繁英借问光。 乱古人不曾，战马乐曾是。
	六卷遇津人否泰，繁幸家县主牖竭。 乱吴官作海通津，战论诏数赠君室。	六岁事晚奏变却，繁逼秦旋壮壁战。 乱羨数许战战逸，战酣隅翔泰比浮。
初始输入为 六繁乱战寒归回 的藏头诗	六德味是良，繁疾疾许曾。 乱金凤翼歌，战调沦恩光。 寒日正相酬，归泪沾衣疾。 兴休歌声催，回首看盖残。	六七伯采芝，繁金屋珠带。 乱金金鳞图，战鳞龙妇乐。 寒风忽闻海，归歌声催曲。 兴来饮酒休，回首鱼食身。
	六龙丝索荣何人，繁时归县状静迟。 乱映源可敛易热，战看羨体战灼战。 寒翔舞战后休覆，归却尽云尽望尽。 兴第山念羨筭查，回庆鼓鞞好映误。	六战海举热寝比，繁羨可贵所寝热。 乱索姿惊战媒诏，战映笔映斜泰静。 寒锦绣合斑绿花，归却境岸频暖忆。 兴白酒扫岫映敛，回曙照薄飞荐官。
初始输入为首字 六 的诗	六侯家采舍，翁曾看盖疾。 调迟调调餐，众里客暂休。	六尺鱼曾教，纤纤目是恩。 光垂衣卷绣，帐待髻衣朱。
	六兵奏却曛辞却，息泰关东海大远。 祠曲炉体战场岫，大裁绕战牖赠仞。	六丈竭余汗漫酣，珍洁感所闭关玩。 春直调战相骑羨，柳醉归骑此暮旋。
	六侯家大贤，贤方自乐惟。 有酒乐曾是，吾家疾取化。 丹砂自相许，衣酒厄从此。 地翻土凤凰，歌凤翼集蓬。	六尺乐曾扶，桑宁鬓暂休。 时恩光报饮，苦饮酒休食。 睡日休垂泪，沾衣取教衣。 疾疾休奏带，酒酒待楼台。
	六匹时极恨至逸，海泰信六匹覆舟。 迹暮逼岫没坟泰，羨曙热匹漠旌帐。 暮改泰扫盛却曙，取取去日徐里丽。 辞休嗽笔宾钩唱，琴首所查暮赠岫。	六绿曛呼故起骑，陂尘断却久恩冥。 失桥路壁绝忧骨，苑战没匡满草窟。 润六白直白岸倒，纷结祖宿照天接。 逼竭帐取合歇我，可索吴逼荐却通。
初始输入为首句 六代江山在 的诗	六代江山在，干小凤兵冠。 奏待酬凤凰，带金貂蝉干。	六代江山在，新开众里移。 花逐红凤畏，日看带衣冠。
	六代江山在，干小凤兵冠。 奏待酬凤凰，带金貂蝉干。 金夷乐武侯，家看黄金谷。 初带酒千旦，醉春衣酒看。	六代江山在，新开众里移。 花逐红凤畏，日看带衣冠。 重千金炉瑞，微挥漾泛金。 鳞轻忽忆此，语醉语蓬蒿。
初始输入为首句 巴山楚水凄凉地 的诗	巴山楚水凄凉地，岫郡田衣帝岫历。 战晴战后时钟战，灼死覆秦滑川岫。	巴山楚水凄凉地，郡峡深秋岫暮至。 声热冷电倚匹汝，竹丛意迹战体幽。
	巴山楚水凄凉地，岫郡田衣帝岫历。 战晴战后时钟战，灼死覆秦滑川岫。 翔索化体骑神旋，织字香岸帅洲白。 却倚却高飞县逸，海师惜尔学乌宦。	巴山楚水凄凉地，郡峡深秋岫暮至。 声热冷电倚匹汝，竹丛意迹战体幽。 羨战器六达敛戟，丽县愁眉录数毛。 苍苍苍苍梧烟波，眉释羁啸语白边。

4 结论

作为一个无标签的 NLG 任务，其评价标准可以是多样化的。由于时间及个人能力限制，仅在字数及平仄关系上进行的改进。使生成诗句具有基本的诗句表达习惯，但不足也很明显，生成的内容显得无意义或难以解释。这与网络结构与过程有关，由于诗的特殊性，字与字之间的关系没有白话文中如此紧密，情感上亦难以分析。下一步的改进可以首先训练定长的诗句译文，而后根据译文再联系原诗句，这样即可加入语义及情感因素。

另一个方面，就目前的网络结构及训练而言，语义库的限制也会导致训练结果不理想，若对语义库进一步分类，按照诗的情感分类会更好。pypinyin 模块仅返回一个最常用的读音，事实上同一个字有多音，古诗中的字读音与现代字的读音也不相同。例如“了”在本实验中处理为“le（轻音）”，即按照平仄规则不会出现该字，但事实上古诗中并不乏该字的使用，如“了却君王天下事”。

参考文献

- [1] Junyoung Chung, Çağlar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014.
- [2] S Hochreiter and J Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.
- [3] Zheng Xiao and PiJun Liang. Chinese sentiment analysis using bidirectional LSTM with word embedding. In *Cloud Computing and Security - Second International Conference, ICCCS 2016, Nanjing, China, July 29-31, 2016, Revised Selected Papers, Part II*, pages 601–610, 2016.