AI古诗创作初探——从有监督到无监督
Poem Generation in supervised and unsupervised scenario

朱大卫 郑书泓 朱家祺 岳鹏云 丁睿

2020 春 · AI引论

概述
Abstract

What did we do?

- 利用经典 N vs N RNN 网络初步训练了一个中文五言藏头诗创作器
- 在无监督情形下 *(无风格标签)*， 利用加入后验变分推断 *(posterior variational inference)* 的 Seq2Seq 模型训练了一个中文五言风格古诗创作器，采用相同训练数据集

How did we release our models and data?

- Available at https://github.com/dromniscience/nlp_ai_2020 . Meantime, a simple GUI and a bunch of comments are provided for convenience.
- Models are implemented in *Pytorch*. (version 1.4.0)

目录
Roadmap

Part I
藏头诗生成器 RNN

Acrostic Poem Generator

Part II
预训练词向量

Pre-trained word embedding

Part III
无监督风格古诗生成器 Seq2Seq + Attention + Variational Inference

Unsupervised Stylistic Poem Generator

Part I

Model built-up: All members
Post processing: 朱家祺 郑书泓
Slides written by: 郑书泓
Presenter: 丁睿

藏头诗生成模型

Acrostic Poem Generator

基于论文:

Chinese Poetry Generation with Recurrent Neural Networks

Xingxing Zhang et al.

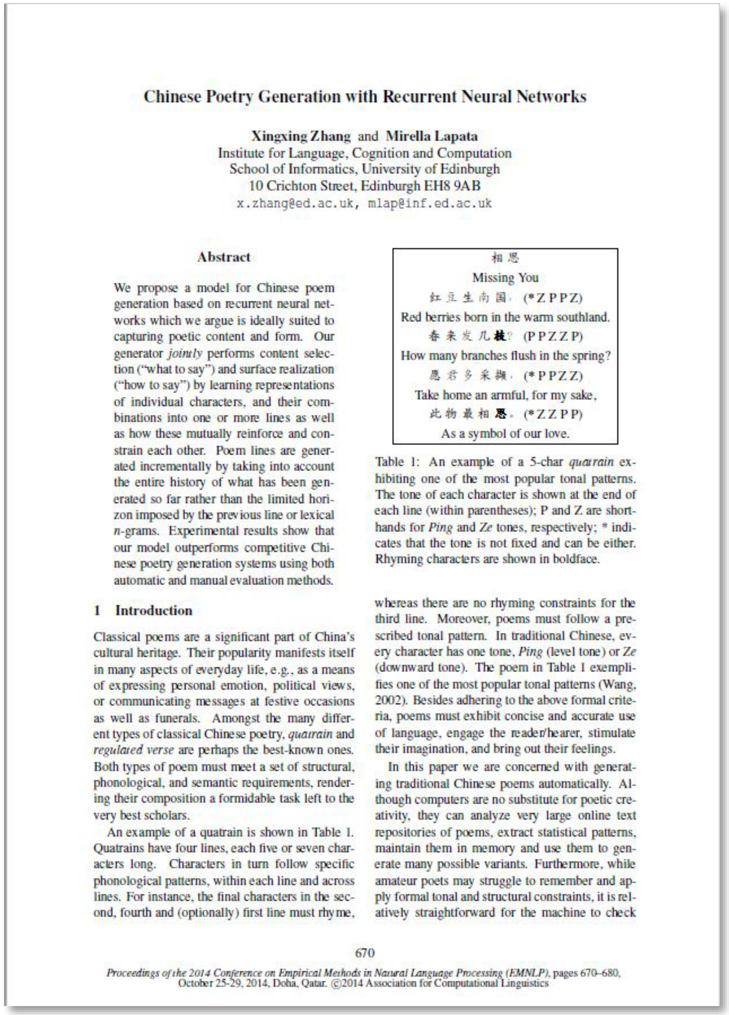
Proceedings of the 2014 Conference on Empirical Methods in NLP

运行环境:

Python 3.x

Pytorch 1.4.0 or later

模型基本是原论文网络结构的实现。



Chinese Poetry Generation with Recurrent Neural Networks

Xingxing Zhang and **Mirella Lapata**
Institute for Language, Cognition and Computation
School of Informatics, University of Edinburgh
10 Crichton Street, Edinburgh EH8 9AB
x.zhang@ed.ac.uk, mlap@inf.ed.ac.uk

Abstract

We propose a model for Chinese poem generation based on recurrent neural networks which we argue is ideally suited to capturing poetic content and form. Our generator *jointly* performs content selection ("what to say") and surface realization ("how to say") by learning representations of individual characters, and their combinations into one or more lines as well as how these mutually reinforce and constrain each other. Poem lines are generated incrementally by taking into account the entire history of what has been generated so far rather than the limited horizon imposed by the previous line or lexical *n*-grams. Experimental results show that our model outperforms competitive Chinese poetry generation systems using both automatic and manual evaluation methods.

相思
Missing You
红豆生南国，(*Z P P Z)
Red berries born in the warm southland.
春来发几**枝**？(P P Z Z P)
How many branches flush in the spring?
愿君多采撷，(*P P Z Z)
Take home an armful, for my sake,
此物最相**思**。(*Z Z P P)
As a symbol of our love.

Table 1: An example of a 5-char *quatrain* exhibiting one of the most popular tonal patterns. The tone of each character is shown at the end of each line (within parentheses); P and Z are shorthands for *Ping* and *Ze* tones, respectively; * indicates that the tone is not fixed and can be either. Rhyming characters are shown in boldface.

1 Introduction

Classical poems are a significant part of China's cultural heritage. Their popularity manifests itself in many aspects of everyday life, e.g., as a means of expressing personal emotion, political views, or communicating messages at festive occasions as well as funerals. Amongst the many different types of classical Chinese poetry, *quatrain* and *regulated verse* are perhaps the best-known ones. Both types of poem must meet a set of structural, phonological, and semantic requirements, rendering their composition a formidable task left to the very best scholars.

An example of a quatrain is shown in Table 1. Quatrains have four lines, each five or seven characters long. Characters in turn follow specific phonological patterns, within each line and across lines. For instance, the final characters in the second, fourth and (optionally) first line must rhyme, whereas there are no rhyming constraints for the third line. Moreover, poems must follow a prescribed tonal pattern. In traditional Chinese, every character has one tone, *Ping* (level tone) or *Ze* (downward tone). The poem in Table 1 exemplifies one of the most popular tonal patterns (Wang, 2002). Besides adhering to the above formal criteria, poems must exhibit concise and accurate use of language, engage the reader/hearer, stimulate their imagination, and bring out their feelings.

In this paper we are concerned with generating traditional Chinese poems automatically. Although computers are no substitute for poetic creativity, they can analyze very large online text repositories of poems, extract statistical patterns, maintain them in memory and use them to generate many possible variants. Furthermore, while amateur poets may struggle to remember and apply formal tonal and structural constraints, it is relatively straightforward for the machine to check

Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 670–680,
October 25-29, 2014, Doha, Qatar. ©2014 Association for Computational Linguistics

模型介绍
Our models

核心框架
CSM + RCM + RGM

产生之前每句诗的表示

Convolution Sentence Model -> 1D Convolution
Recurrent Context Model -> RNN encoder
Recurrent Generation Model -> RNN decoder

利用 当前行已生成字的信息
和 之前诗句的表示
生成 当前诗句的下一个字

为当前行的每个位置
产生 context vector

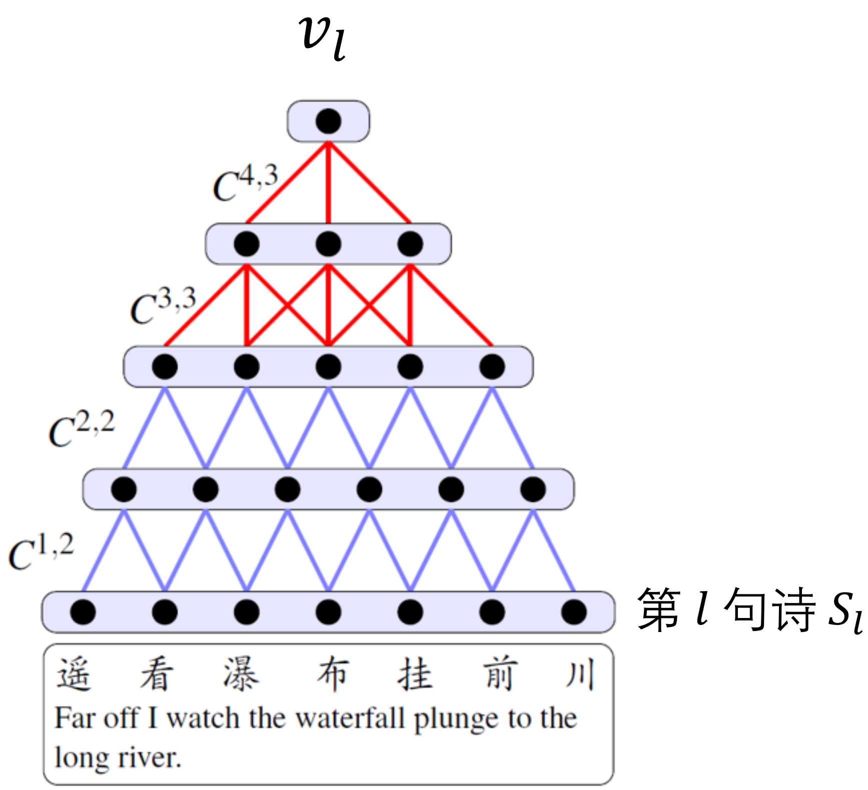
模型介绍
Our models

核心框架
CSM + RCM + RGM

CSM -> 1D Convolution
RCM -> RNN encoder
RGM -> RNN decoder

CSM 是一组 1D-卷积核，将词嵌入的诗句(seq_len * vec_dim)卷到 1*vec_dim.

1D意味着卷积核做Hadamard积.



模型介绍
Our models

核心框架
CSM + RCM + RGM

CSM -> 1D Convolution
RCM -> RNN encoder
RGM -> RNN decoder

常规的RNN结构.

假设当前在产生第 i 句诗每个位置 j 的context vector, 记为 u_i^j . 那么

Input Sequence: $v_{1:i-1}$, initial hidden state $h_0 = 0$

Output: last hidden state h_i

Then: $u_i^j = \sigma(U_j \cdot h_i)$

模型介绍
Our models

核心框架
CSM + RCM + RGM

CSM -> 1D Convolution
RCM -> RNN encoder
RGM -> RNN decoder

RGM根据该公式生成第 $i+1$ 句诗 S_{i+1}：

$$P(S_{i+1}|S_{1:i}) = \prod_{j=1}^{m-1} P(w_{j+1}|\boxed{w_{1:j}}, u_i^j)$$

Teacher Forcing

模型介绍
Our models

核心框架

CSM + RCM + RGM

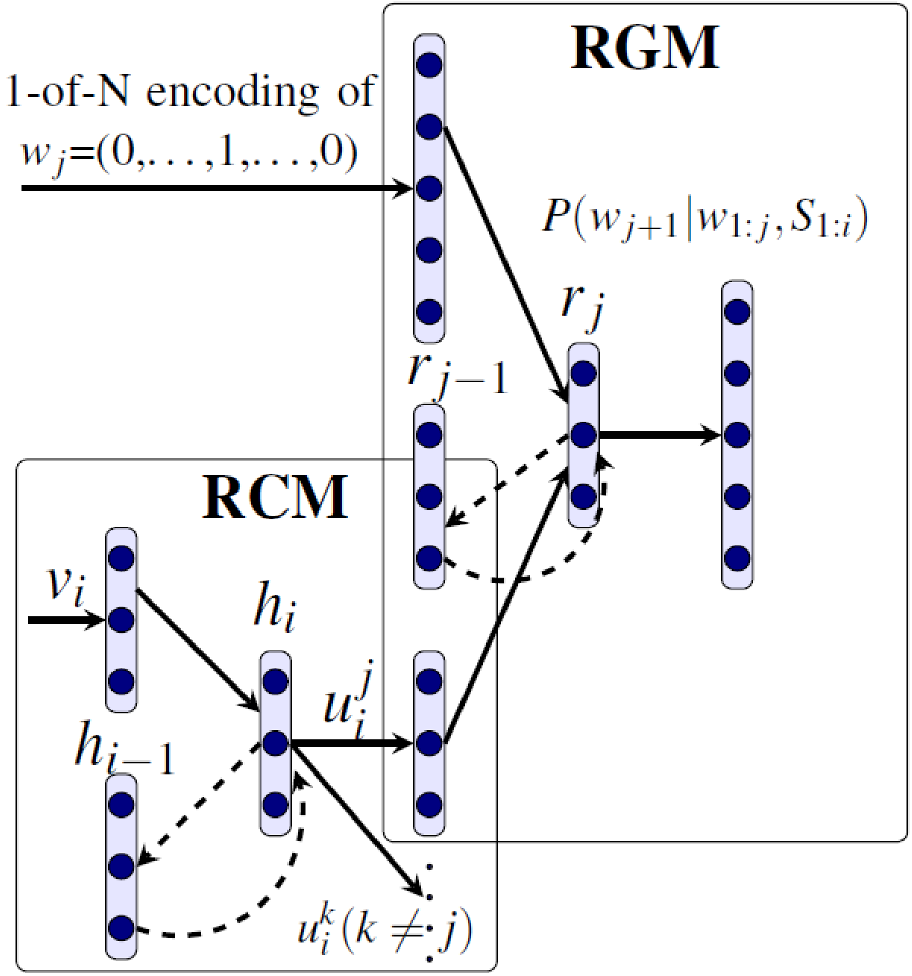
CSM -> 1D Convolution
RCM -> RNN encoder
RGM -> RNN decoder

完整结构 ->

$$v_i = \mathrm{CSM}(S_i)$$

$$u_i^j = \mathrm{RCM}(v_{1:i}, j)$$

$$P(w_{j+1}|w_{1:j}, S_{1:i}) = \mathrm{RGM}(w_{1:j+1}, u_i^{1:j})$$



Training Objective: To maximize

$$P(S_{i+1}|S_{1:i}) = \prod_{j=1}^{m-1} P(w_{j+1}|w_{1:j}, u_i^j)$$

实现与效果展示

Implementation & Performance

藏头诗模型的生成模式为给定每首诗的第一句话和后三句的首字，让模型生成后三句后面的字 (一共生成字数3×4=12 for 五言， 3×6=18 for 七言) , 训练10 - 35个epoch.

山居秋暝

空山新雨后 天气晚来秋 明月松间照 清泉石上流	→	空山新雨后 天高一剑杵 明月下蘋渚 清江烟雾里

旅夜书怀

细草微风岸 危樯独夜舟 星垂平野阔 月涌大江流	→	细草微风岸 危堤别离情 星山断肠断 月满敝箧中

* 生成诗歌皆非在训练集中, 表示模型已经学到了一些诗歌的结构

效果展示 & 简要评价

Performance & Brief Assessment

- 生成的字是能与前一个字组成词组的字
- **Culprit:** RNN直接输入只有生成的前一个字, 缺乏同一句中远距离的信息 (下一个模型中将用bi-GRU)
- **E.g.** 给定首字为“夜”： 夜夜夜夜夜
  给定首字为“笑”： 笑嘻嘻嘻嘻嘻嘻

效果展示 & 简要评价

Performance & Brief Assessment

- 模型学到的参数可能并不是在当前语境下如何创作, 而是复述一些固定搭配。
- **E.g.** 经常产生短语“不成匹” (出自“理丝入残机，何悟不成匹”)， 但这个短语在除了织布外的绝大多数语境下都没有意义。
- **Culprit:** 网络拓扑必须调整, 尤其是RCM中 $u_i^j = \sigma(U_j \cdot h_i)$ 局限很大。

Overall

- 模型产生时间较早，效果有局限。在不考虑 音韵(tonal pattern) 和断句(segment) 等约束帮助的情况下并不非常理想。

Part II

Work done by: 朱家祺
Slides written by: 朱家祺
Presenter: 丁睿

This part provides suitable word embedding for models in Part III.

跳字模型
Skip-gram

用中心词预测语境词

对于中心词，指定一个背景窗口大小，可以采集到它对应的背景词。假设给定中心词情况下，背景词的生成相互独立。

例 假设“举头望明月”的中心词是“望”，背景窗口大小为 2. 则

$$P(\text{举}, \text{头}, \text{明}, \text{月}|\text{望}) = P(\text{举}|\text{望}) \cdot P(\text{头}|\text{望}) \cdot P(\text{明}|\text{望}) \cdot P(\text{月}|\text{望}).$$

设中心词 w_c 的字典索引为 c, 背景词 w_o 的字典索引为 o. 模型利用内积和 softmax 可得到:

$$P(w_o|w_c) = \frac{\exp(u_o \cdot v_c)}{\sum_{i \in V} \exp(u_i \cdot v_c)}.$$

Caution

同一个词在 中心词 和 背景词 时分别有一个词向量!

跳字模型
Skip-gram

假设一个长度为 T 的文本 S, 窗口大小为 n. 假设时间步为 t 的词 $w^{(t)}$. 跳字模型的似然函数为:

$$L(S) = \prod_{t=1}^{T} \left(\prod_{\substack{j=-n,\\ j\neq 0}}^{n} P(w^{(t+j)} | w^{(t)}) \right)$$

或者等价的负对数版本:

$$L_{-log}(S) = -\sum_{t=1}^{T} \sum_{\substack{j=-n,\\ j\neq 0}}^{n} \log P(w^{(t+j)} | w^{(t)})$$

为了最大化似然函数, 就是最小化 $L_{-log}(S)$. 每次随机采样较短的子序列计算损失, 并更新模型参数. 在训练完成后, 用中心词向量作为表示该词的向量.

跳字模型

Skip-gram

最后以余弦相似度评价词的相似程度即可

我们提供了一个寻找古诗中相似字的API.

常规的近义关系

一些更深刻的近义关系

吴音 -> 家乡话 / 客家话
侬 -> 已婚妇女(有时引申为家乡所在, 台湾话还在用)

跳字模型
Skip-gram

You shall know a word by the company it keeps.

非常经典且有效的想法，但是仍然会有一些瑕疵。具体地，如反义词有时也会训练出相关度高的词向量。因为总体上反义词与近义词几乎不会同时出现，而各自独立的出现环境又是相似的。

Part III

Model built-up: 朱大卫 岳鹏云 丁睿
Post processing: All members
Slides written by: 丁睿
Presenter: 丁睿

无监督的风格化古诗生成模型
Unsupervised Stylistic Poem Generation

基于论文:

Stylistic Chinese Poetry Generation via Unsupervised Style Disentanglement

Cheng Yang et al.

Proceedings of the 2018 Conference on Empirical Methods in NLP

运行环境:

Python 3.x

Pytorch 1.4.0 or later

Recommended on GPU

根据实际情况对论文的框架多有改动，
但无监督部分的核心框架没有改变。

Stylistic Chinese Poetry Generation via Unsupervised Style Disentanglement

Cheng Yang[1,2,3], Maosong Sun[1,2,4*], Xiaoyuan Yi[1,2,3], Wenhao Li[1,2,3]
[1]Department of Computer Science and Technology, Tsinghua University
[2]Institute for Artificial Intelligence, Tsinghua University
[3]State Key Lab on Intelligent Technology and Systems, Tsinghua University
[4]Jiangsu Collaborative Innovation Center for Language Ability, Jiangsu Normal University
{cheng-ya14,yi-xy16,liwh16}@mails.tsinghua.edu.cn
sms@tsinghua.edu.cn

Abstract

The ability to write diverse poems in different styles under the same poetic imagery is an important characteristic of human poetry writing. Most previous works on automatic Chinese poetry generation focused on improving the coherency among lines. Some work explored style transfer but suffered from expensive expert labeling of poem styles. In this paper, we target on stylistic poetry generation in a fully unsupervised manner for the first time. We propose a novel model which requires no supervised style labeling by incorporating mutual information, a concept in information theory, into modeling. Experimental results show that our model is able to generate stylistic poems without losing fluency and coherency.

Figure 1: An example of poems in diverse styles under the same keyword.

1 Introduction

Classical Chinese poetries are great heritages of the history of Chinese culture. One of the most popular genres of classical Chinese poems, *i.e.* *quatrains*, contains four lines with five or seven characters each and additional rhythm and tune restrictions. During 1,000 years history of quatrains, various styles, *e.g.* pastoral, descriptive and romantic, have been developed to express different feelings of poets. In human poetry writing, poets are able to write completely different poems in diverse styles even given the same keyword or first sentence. For example, as shown in Fig. 1, when a poet mentioned "月" (the moon), she/he may write about the Great Wall in the battlefield style or the sleepless feeling in the romantic style. Such ability to write stylistic poems under the same poetic imagery is an important characteristic of human poetries.

Automatic poetry generation is one of the first attempts towards computer writing. Chinese quatrain generation has also attracted much attention in recent years. Early works inspired by statistical machine translation explored rule-based and template-based methods (He et al., 2012; Yan et al., 2013), while recent works (Zhang and Lapata, 2014; Wang et al., 2016; Yan, 2016; Zhang et al., 2017; Yang et al., 2017; Yi et al., 2017) employed neural network based sequence-to-sequence approaches which have shown their effectiveness in neural machine translation for poem generation. Most works target on improving the coherency among all lines and the conformity between the theme and subsequent lines by planning (Wang et al., 2016), polishing schema (Yan, 2016), poem block (Yi et al., 2017) and conditional variational autoencoder (Yang et al., 2017). Different from these previous works, we aim to learn the ability of diverse stylistic poetry generation which can generate multiple outputs (poems) in various styles under the same input (keywords or the first sentence). This ability enables a poetry generation system to be closer to a real poet and allows the model to generate more expressive and creative poems.

However, there is no explicit label about what style or category a poem or a sentence is for thousands of poems in the database. Therefore, traditional supervised sequence-to-sequence mod-

* corresponding author: sms@tsinghua.edu.cn

3960

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3960–3969
Brussels, Belgium, October 31 - November 4, 2018. ©2018 Association for Computational Linguistics

模型介绍
Our Models

最基本的框架

Seq2Seq + Attention

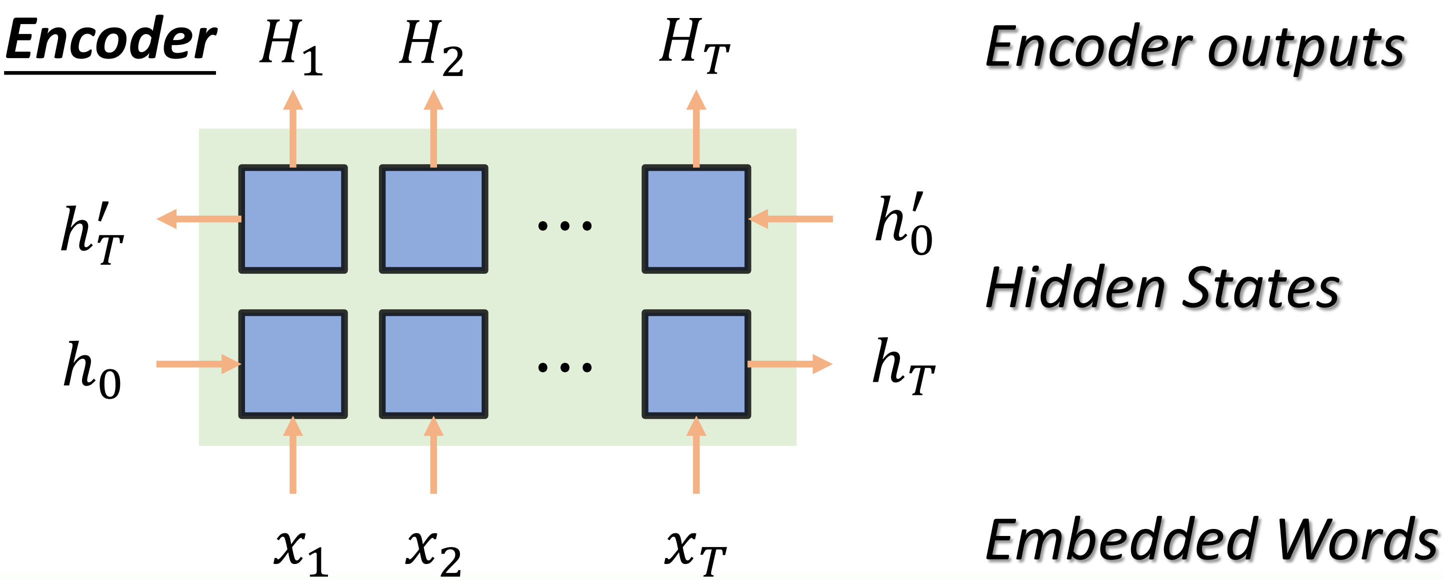
Encoder -> bi-GRU

Decoder -> GRU + Transform Layer

避免Overfitting :

$H_i = h_i + h'_{T+1-i}$ instead of $H_i = [h_i\ ;\ h'_{T+1-i}]$

Warning: It is task-specific! It is due to the scale of our corpus.



模型介绍

Our Models

最基本的框架

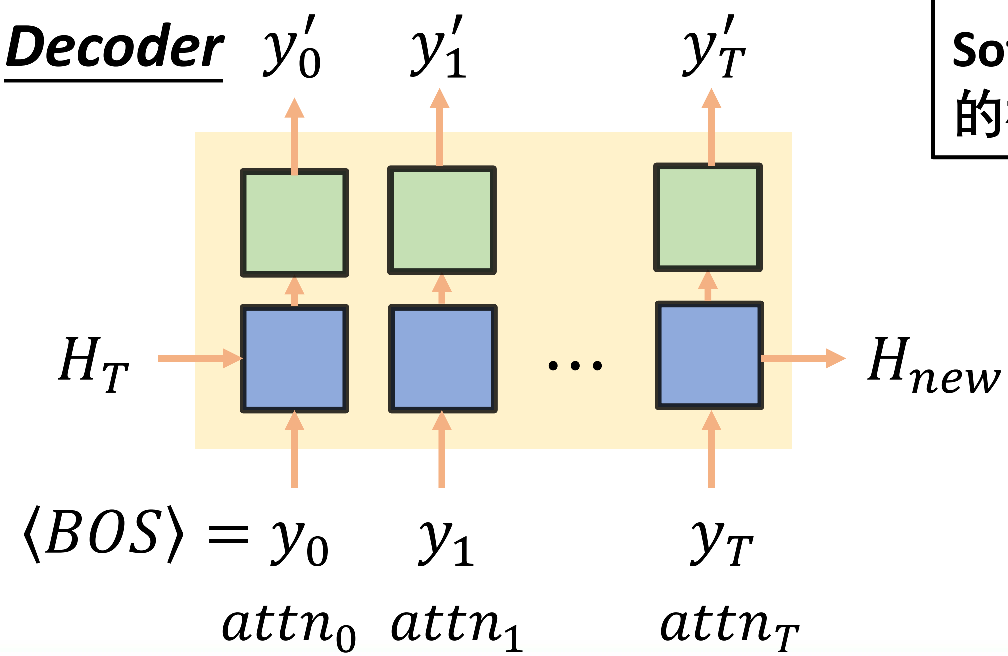
Seq2Seq + Attention

Encoder -> bi-GRU

Decoder -> GRU + Transform Layer

Blue Block -> GRU Cell
Green Block -> Linear Transform

Softmax(y_i') 表示下一个字在字典中的概率分布



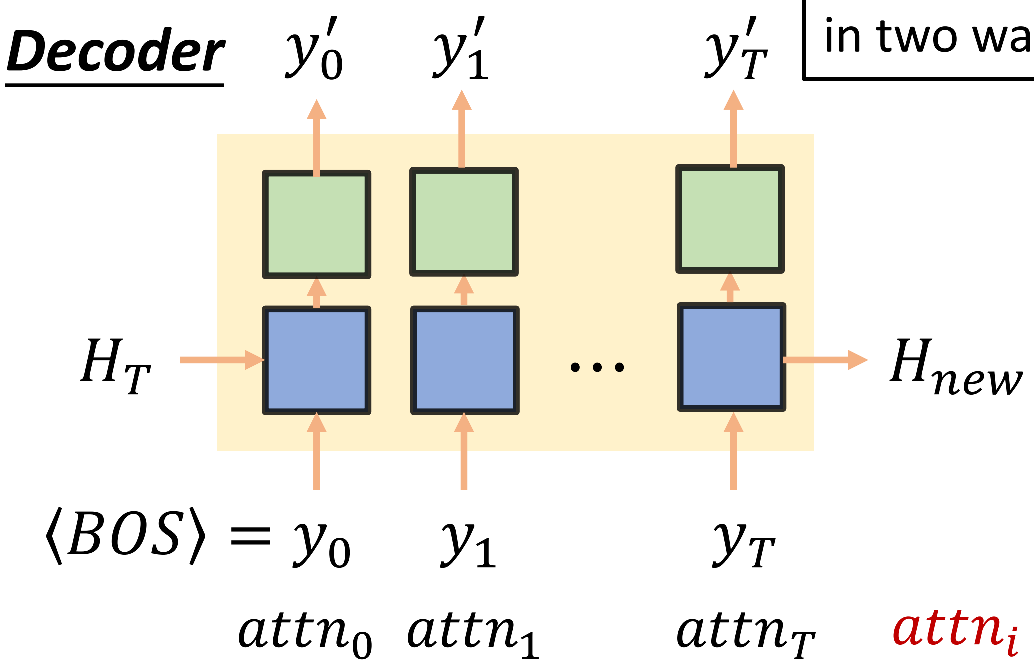
模型介绍
Our Models

最基本的框架
Seq2Seq + Attention

Teacher forcing:

Both in y_i and $attn_i$

It is fragile in terms of generating, since no ground truth is provided. We will address it in two ways.(both explicitly and implicitly)



Weighted Sum

$$attn_i = \boldsymbol{Attention}(y_i, hidden_i, H_{1:T})$$

模型介绍
Our Models

无监督情形

Input = (X=$[x_1, x_2, \ldots, x_T]$, style id)
风格以one-hot向量表示，直接拼接到decoder的initial hidden state中

Problems

- 不能仅通过one-hot向量就保证输出与style id有较好的相关关系
- Style的意义不明确，且没有相应风格的后继文本供训练

One Possible Solution

在损失上增加一个风格损失的正则项。同时将风格和生成的文本都看成其所在空间上的概率分布。希望生成文本尽可能与style id相关，可以理解为最大化两个概率分布间的“相关性”，即给定了其中一个变量，能得到另一变量分布的信息尽可能多。

互信息
Mutual Information

Sorry to display math formula a bit :)

给定两个随机变量 X, Y, 则 X 与 Y 的互信息定义为

$$I(X,Y) = \int_Y \int_X p(X,Y) \log \frac{p(X,Y)}{p(X)p(Y)} dX dY.$$

Intuitive Facts

- 如果 X, Y 相互独立, 那么 $I(X,Y) = 0$.
- 如果 X, Y 在它们的每个实现上存在一一对应关系, 那么$I(X,Y) = H_X = H_Y$.
- $I(X,Y) = I(Y,X)$, 且不会超过 H_X 或 H_Y.(H 代表 entropy operator)

训练目标
Objective

最大化风格与生成诗句的互信息

无妨假设先验的风格分布是一个均匀分布 (uniform distribution), 即

$$\Pr(Sty = k) = \frac{1}{K}$$

其中 K 是风格总数。风格的取值为1,2,…,K. 于是

$$\begin{aligned}
& I(\Pr(Sty), \Pr(Y;X)) \\
&= \sum_{k=1}^{K} \Pr(Sty = k) \int_{Y|k;X} \log \frac{\Pr(Y, Sty = k; X)}{\Pr(Sty = k)\Pr(Y;X)} dY \\
&= \sum_{k=1}^{K} \Pr(Sty = k) \int_{Y|k;X} \log \frac{\Pr(Y, Sty = k; X)}{\Pr(Y;X)} dY \\
&\quad - \sum_{k=1}^{K} \Pr(Sty = k) \log \Pr(Sty = k)
\end{aligned}$$

这不是我们展示的目标，让我们先跳过它。其中的运算是基础的。

最大化互信息
Maximize Mutual Info.

$$
\begin{aligned}
& I(\Pr(Sty), \Pr(Y;X)) \\
&= \sum_{k=1}^{K} \Pr(Sty = k) \int_{Y|k;X} \log \Pr(Sty = k|Y) dY + \log K \\
&= \int_{Y;X} \sum_{k=1}^{K} \Pr(Sty = k|Y) \log \boxed{P(Sty = k|Y)} dY + \log K.
\end{aligned}
$$

真实的分布可能过分复杂，我们利用一个后验分布估计它。这个后验分布包含可训练的参数。通过学习，它能够在自己的表示范围内拟合真实的分布。

Further explanation and its intuitiveness are contained in the following two slides.

变分推断技术
Variational Inference Maximization

设 Q 为选择的后验估计函数，那么最大化

$$\sum_{1}^{k} Pr(Sty=k)\int_{Y|k;X}\log Q(Sty=k|Y)\mathrm{d}Y,$$

就是最大化真实分布的一个下界:

$$
\begin{aligned}
& I(\Pr(Sty),\Pr(Y;X)) - \log K \\
&= \int_{Y;X}\sum_{k=1}^{K}\Pr(Sty=k|Y)\log\Pr(Sty=k|Y)dY \\
&= \int_{Y;X}\sum_{k=1}^{K}\Pr(Sty=k|Y)\log Q(Sty=k|Y)dY \\
&\quad + \int_{Y;X}\boxed{\sum_{k=1}^{K}\Pr(Sty=k|Y)\log\frac{\Pr(Sty=k|Y)}{Q(Sty=k|Y)}}dY
\end{aligned}
$$

$KL(\Pr(Sty|Y),Q(Sty|Y))$

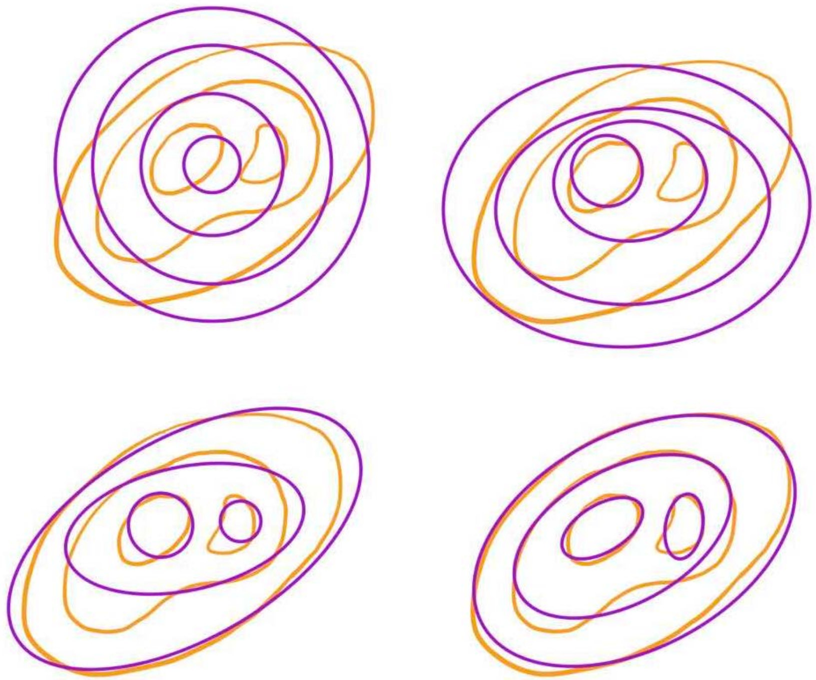
所以是非负的

变分推断技术

Variational Inference Maximization

直观理解（非常粗糙的）

Orange Curves -> actual distribution
Purple Curves -> approximation



Initialize

Mimic

Converge

Stable

拟合的性能取决于后验函数的形式。极端情况下，后验等于真实分布，那么KL散度为零，就得到紧的下界。

If more info is needed, please search for VAE(Variational Auto Encoder).

我们的设计

In this scenario

对生成的诗句 $Y=[y_1,y_2,\dots,y_T]$(已词嵌入), 其后验估计函数定义为

$$Q(Sty|Y)=\mathrm{softmax}(W\cdot\frac{1}{T}\sum_{i=1}^{T}y_i).$$

余下的就是积分空间的问题, 我们不可能遍历所有的 Y. 于是我们试图选择一个代表性的 Y, 并认为它可以代表整体的平均 (或者说有 100% 的概率将会生成这个 Y).

$$\mathrm{expect}(i;k,X)=\sum_{c\in V}\mathrm{prob}[c]\cdot\mathrm{embed}(c)$$

其中 k,X 是输入风格编号和诗句, i 表示当前正在产生第 i 个位置上的期望词向量. V 指词空间, prob 由 output 向量经过 softmax 得到, 而 embed 表示词嵌入 (这里的"词"实际都是指"字").

训练目标 最终版

Complete objective

风格损失

$$\mathcal{L}_{reg} = \frac{1}{K}\sum_{k=1}^{K} \log Q(k \mid [\text{expect}(i; k, X) \text{ for } 1 \le i \le T])$$

Implicitly dealing with the drawback of teacher forcing

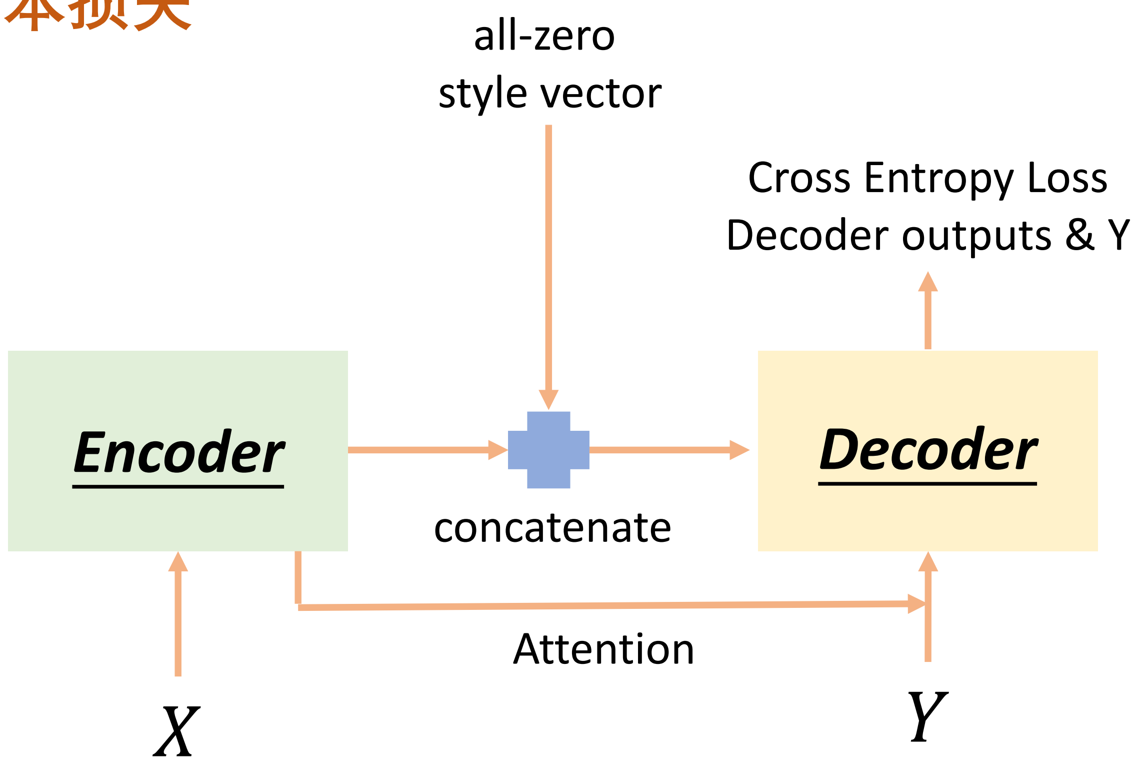
总损失

$$Train(X, Y) = \sum_{i=1}^{T} \log p(y_i | y_1 y_2 \ldots y_{i-1}, X) + \boxed{\lambda \mathcal{L}_{reg}},$$

注意这里 Y 是 X 的下句.这是从训练集直接可以得到的.原来讨论风格损失的计算时积分变量的 Y 是指随机变量的任意实现(realization),它已经被我们用 expect 函数取代了.

网络及训练
Train this Model

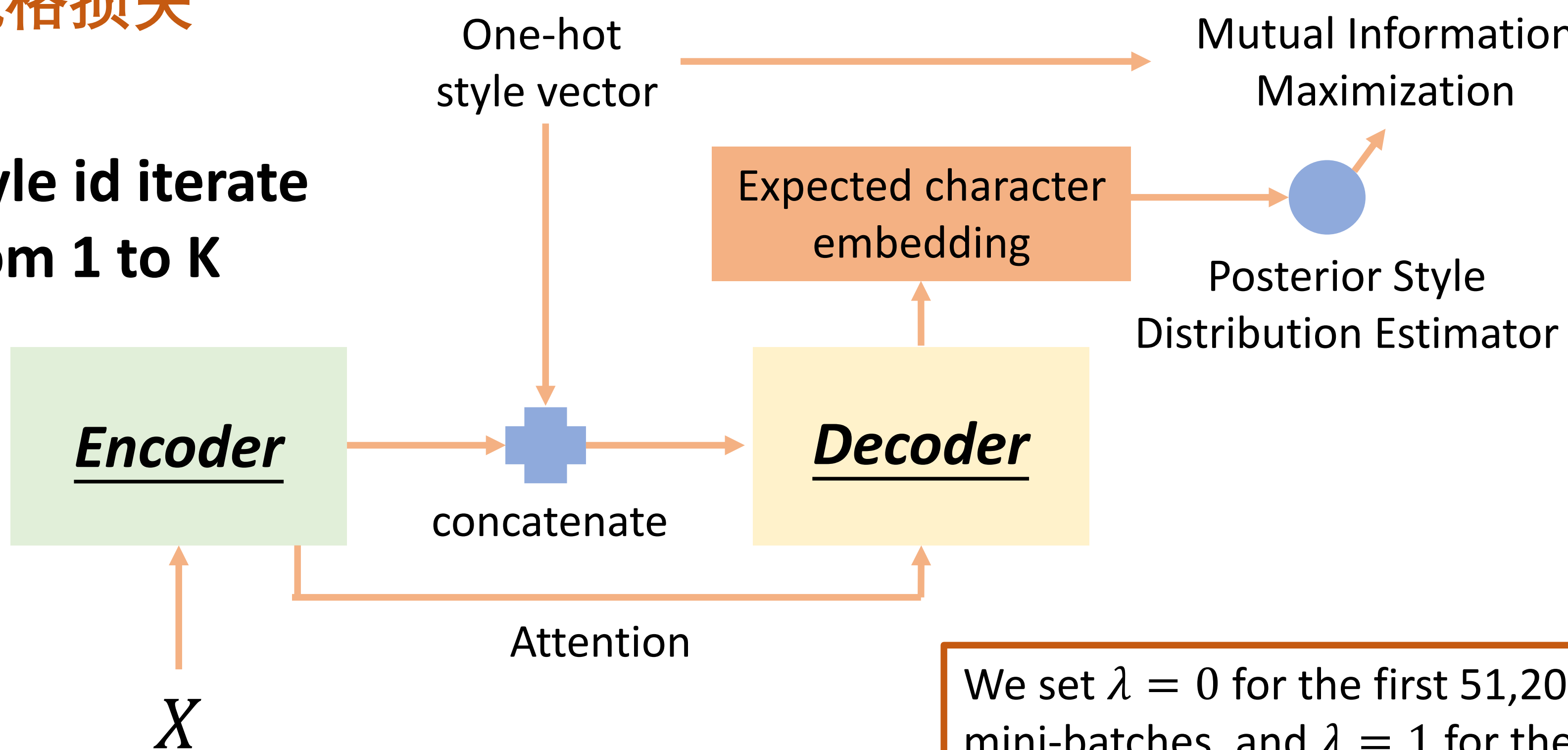
Compute the first term of Loss:
文本损失



网络及训练
Train this Model

Compute the second term of Loss:
风格损失

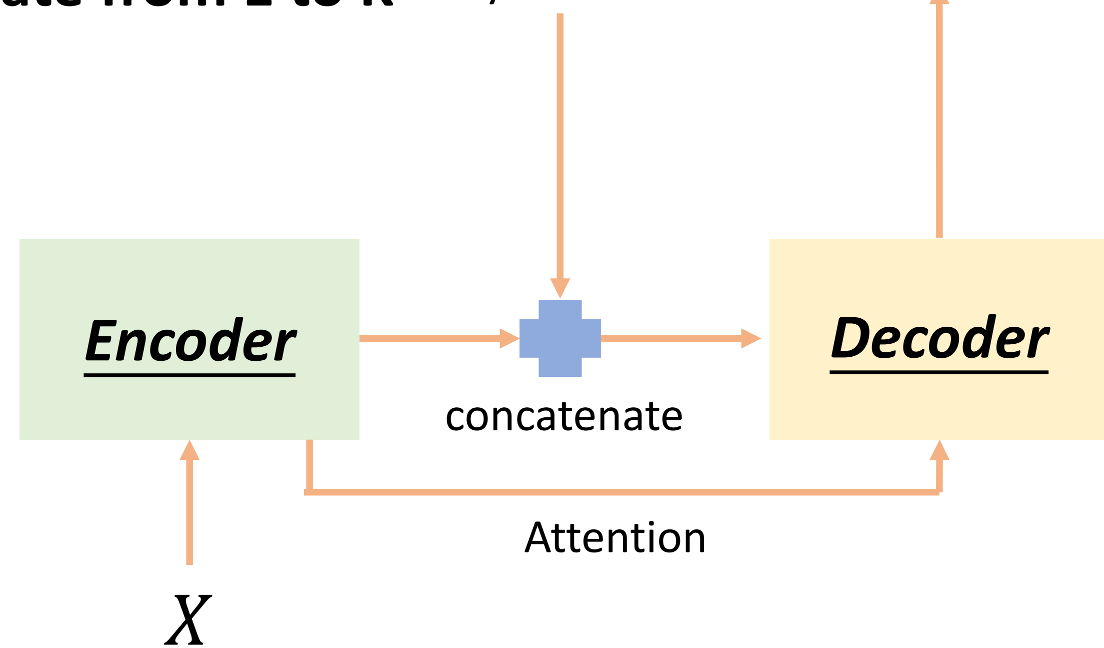
Style id iterate from 1 to K



We set $\lambda = 0$ for the first 51,200 mini-batches, and $\lambda = 1$ for the rest.

古诗生成
Generation

为每个风格生成诗
Iterate from 1 to K



Beam search
以防一步错，步步错

Explicitly dealing with the drawback of teacher forcing

代码展示
Code Display

Sincerely apologize for skipping code display because of the time limit of our presentation.

Our code is available on GitHub. Please contact *https://github.com/dromniscience/nlp_ai_2020*

A simple GUI and a bunch of comments are provided.

The only point I would love to make is:

Pytorch is unarguably much more consistent with the philosophy of Python than *tensorflow*!

(Maybe we can switch to pytorch as the standard package next year. :)

风格的解释

Interpretability of Learned Styles

- 注意这个风格并不是我们理解的诗歌风格，而是诗的某个抽象特征。它是互信息最大化的产物，将决定于已有的隐参数(或者说是features)，尤其是词向量等内容。
- 无法事先预知风格的含义，但是可以通过生成得到样本来评判风格。有一定的概率某些风格会对应我们理解的风格，如情感、意境等特定内容。
- 这也是我们在无监督情况下训练风格的代价——不可预知性。

示例 Examples

Style id 1	Style id 5	Style id 9
春到村居好， 园林亦可怜。 谁知趱桃李， 应似帝王家。	春到村居好， 园林亦可怜。 欲知芳草树， 寂寞淡清樽。	春到村居好， 园林亦可怜。 谁知趱桃李， 应有一枝春。
花间一壶酒， 静处见斜晖。 莫遣清风月， 长涛一泓声。	花间一壶酒， 静处见星斜。 坐问江湖上， 清风起馀情。	花间一壶酒， 静处见斜阳。 坐得千载月， 冷浸玉池香。

效果分析
Performance Assessment

Training Efficiency

单个epoch有176*64条数据, 在 gpu 上训练约 5 min/epoch.

Poem Quality

因为一直在优化, 我们没有来得及使用评估模型(比如BLEU)或者请专业人士评价。欢迎使用我们的模型并将结果反馈给我们!

粗略地说，较之于第一个模型，它在Coherency/Fluency/Poeticness上表现更出色。Coherency指语句之间的联系，Fluency指诗句的格律通顺连贯，Poeticness指诗的意境感。

效果分析 Performance Assessment

Stylistic Assessment

风格上，目前 K=10 中，大约只有3-4个风格大致会有分别，如 id = 1 一般伴随比较壮丽的场景，如 涛 / 潮 / 王 / 万古 等等；id = 5 的意境一般空旷寂寥。个别 id 下有时出现生僻的字，或者喜欢在后两句诗的第四五个字重复。

平均而言风格之间的区别还不够明显。

Future Works

考虑首先完善评价机制。其次，可以添加一些机械性约束，如韵脚、重复字的限制。*(这些我们之后会首先关注的)*

网络本身已达到了不错的效果，但是注意到我们对于模型的风格产生解释后，可以引入同步学习机制优化互信息中风格的 normal distribution 的先验假设*(时间与精力所限，很遗憾我们目前不打算进行这个尝试)*。

最后重要的一点是认识到这里的正则损失项不是 task-specific的。

收获 Conclusions

Talks are cheap. Show me your code.

即使彻底理解论文的内容，也不一定能很好复现成果。一方面编程的基本功力要扎实；更重要的是，论文介绍的仅仅是它的工作中最重要/最富创新性的部分。而一个类似古诗生成这样要求较高的NLP任务往往必须结合传统的一些剪枝方法、合适的网络拓扑以及针对训练的词向量。

模块化文件的组织

规模稍微大一些的项目中文件模块化非常重要。一方面它促进分散与解耦核心功能，使代码运行更高效 (指不必总是愚蠢地执行全部的代码)；另一方面，它在多人合作的情形大大降低了命名冲突和命名空间的问题。

计算资源

该项目使用了较大的数据规模 (原始论文甚至更大,以致会炸掉一个8GB的PC RAM)。我们确实也在节省参数而保留效果的问题上做了不少改进 (最直接的就是bi-GRU output的处理)。

深度学习方法的局限性

面对复杂的真实分布，我们很难选择计算资源允许范围内极好的某个后验去逼近。同样无监督的风格训练也直接反映了解释性较差的问题。但在目前发展的方法论中，深度学习对于庞大数据内在规律的学习能力还是诱人的。

感谢倾听!

Thanks for your attention!

If you are fond of our project, welcome to contact us at any time!

朱大卫 郑书泓 朱家祺 岳鹏云 丁睿

2020 春 · AI引论