

背景介绍

诗词生成任务是让计算机自动创作诗词的技术。诗词生成一直被认为是一种高难度，依赖人类突发灵感的高级思维活动。另一方面，诗词创作又需要遵循严格的规律，如平仄，韵律等。这意味着生成诗词是一项既要循规蹈矩，又要寻求新意的艰苦劳动，仅有对平仄，韵律，意境等具有敏锐感觉，且思维活跃度极高的少数人能够胜任。幸运的是，这种这种在严格框架下进行有限创新的工作，计算机具有天然优势，它可以充分保证生成作品的合规性，同时在合规下探索各种可能的创新。让计算机自动生成诗词，可极大减少人类进行诗词创作的工作量，且有望产生挣脱传统思路束缚新颖诗词。即使用机器生成的诗词还不能与人类的诗人相比，但机器作品可以为人类提供候选或初级作品，使诗人创作更加容易；同时，计算机生成的诗词还可以为诗人提供灵感和刺激，激发他们不断创造新的音乐，仿止因长期创作带来的风格惰性和思维困顿，帮助诗人永褒创作青春。因此，自动诗词生成具有非常广阔的应用前景。

实验介绍

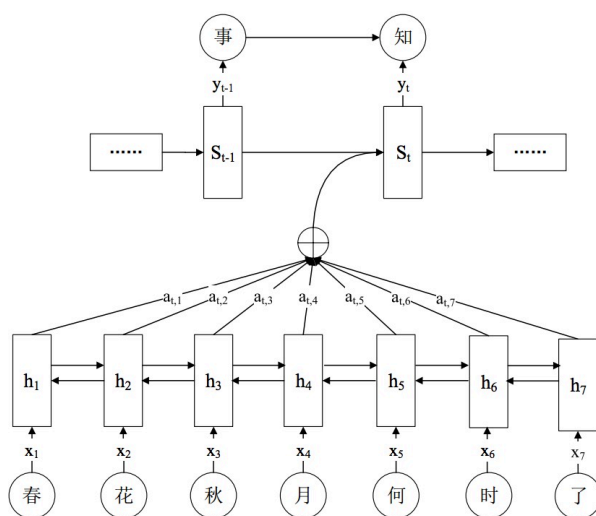
在本次实验中，主要的任务是对已有基于attention的诗词模型改进，使其能生成不同风格的诗词。一个诗人想写出不同风格的诗词，会在自己的记忆中寻找不同风格的诗，然后在平仄，韵律，主题的限定下，创作出不同风格的诗词。基于这个创作过程，我们想到用memory模型表示存储不同风格的诗，attention模型表示学到的规则（平仄，韵律等），然后通过某种方式把这两个模型结合起来，进而产生不同风格的诗词。

实验原理及参数

本次实验使用两个模型，分别是attention和memory模型，下面就这两个模型做一个简单的介绍。

attention模型

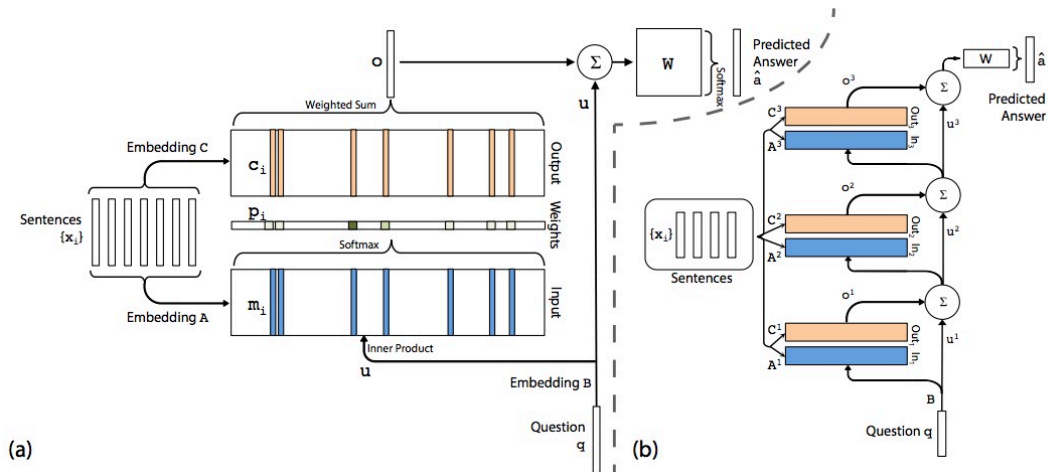
attention机制就是在decoder过程中，可以动态的参考encoder不同部分。这个过程像人类在阅读时根据关注点的不同，聚焦在不同的段落或者句子。attention模型最先在神经机器翻译中应用，取得不错的效果。我们实把attention机制用在了诗词生成中，取得很好的效果，论文被ijcar收录。下图展现attention机制：



memory模型

memory模型可以认为是对Istm的扩展，可以更加高效地筛选相关的信息，在QA和LM中取得不错的效果。下面介绍一下memory模型中在QA的工作机制：

1. 输入一个问题 q 及相应的上下文 x_i
2. 上下文 x_i 通过两个矩阵作为memory的输入和输出
3. 问题 q 与memory的输入进行点积，通过softmax计算出概率，与memory的输出点乘



实验输入及参数

- 数据源为58k首五言和七言的诗词，100首边塞诗和100首言情诗，其中100首边塞诗和100首言情诗分别通过运算形成两个不同风格的memory。
- encoder-decoder的参数为500
- attention的参数为1000
- maxout的参数为500

实验步骤

实验最终目的是探索一个attention和memory模型结合的方式，已达到生成不同风格的诗词的目的。

不过，我们现阶段的任务是证明memory模型有效，并探索两个模型结合方式。目前为止，我们总共设计了三种比较简单的改进方式，分别为encoder-memory，decoder-memory和在decoder-memory基础上结合方式的改变。

实验预备

这三种方式是在相同的基础上做出的改进。此基础是使用相同的attention模型，attention模型不进行更新。

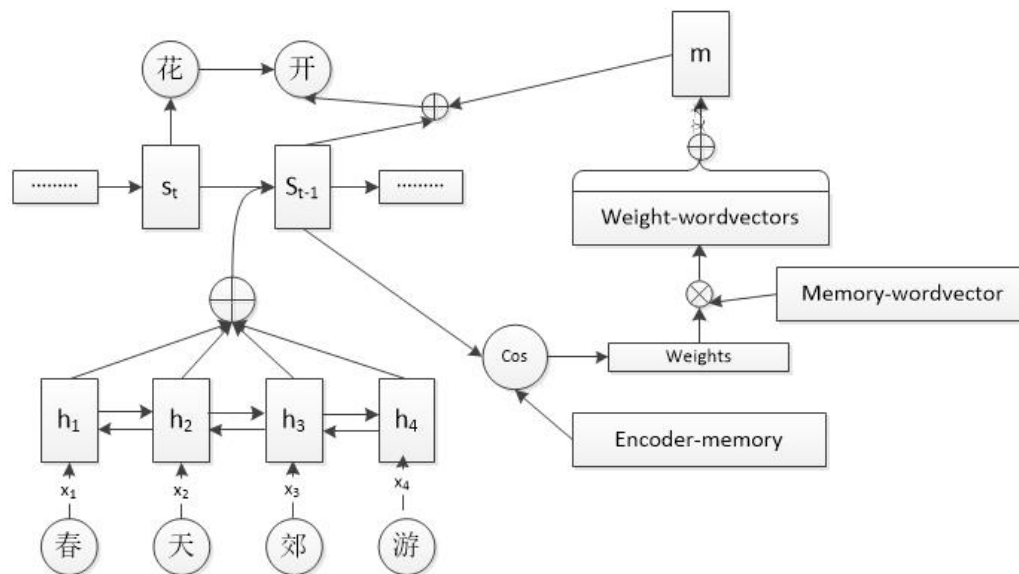
改进一：encoder-memory

encoder-memory是attention模型中encoder的部分生成，下面是encoder-memory的基本过

程：

1. 与当前decoder的隐层通过attention的方式计算出memory中每个state对应的权重
2. 让计算出的权重与其对应的memorywordvector相乘，得到权重化的wordvector
3. 将其相加得到memory的最终输出m
4. m经过一个转换为与decoder隐层相同的维度，与其相加

encoder-memory结构，如下图所示：



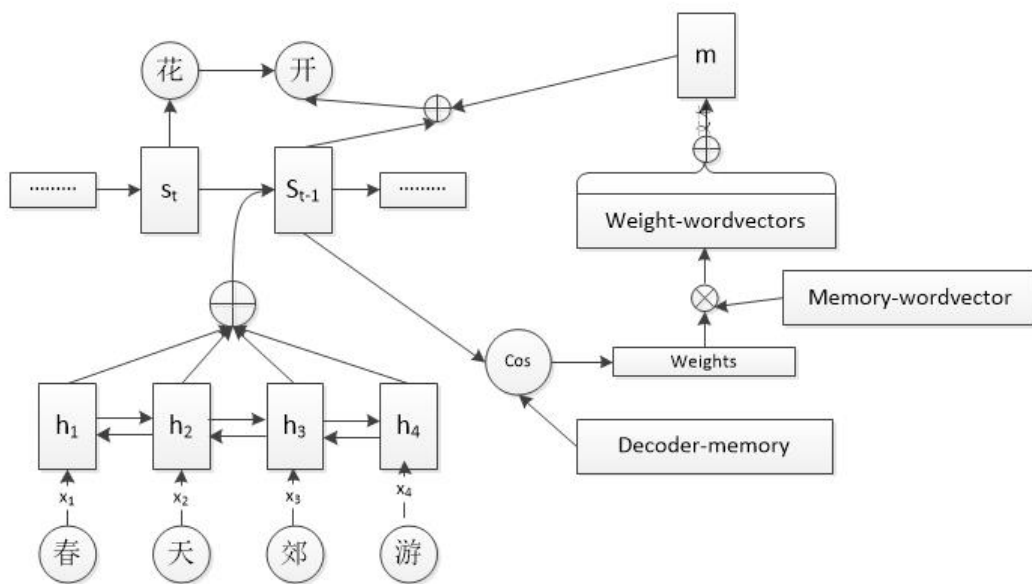
结论：并不能生成不同风格的诗词，因为最后得到的m对结果的影响很有限

改进二：decoder-memory

decoder-memory是attention模型中decoder的部分生成，下面是decoder-memory的基本过程：

1. 与当前decoder的隐层通过cos的方式计算出memory中每个state对应的权重
2. 让计算出的权重与其对应的memorywordvector相乘，得到权重化的wordvector
3. 将其相加得到memory的最终输出m
4. m和decoder的隐层相加

decoder-memory结构，如下图所示：



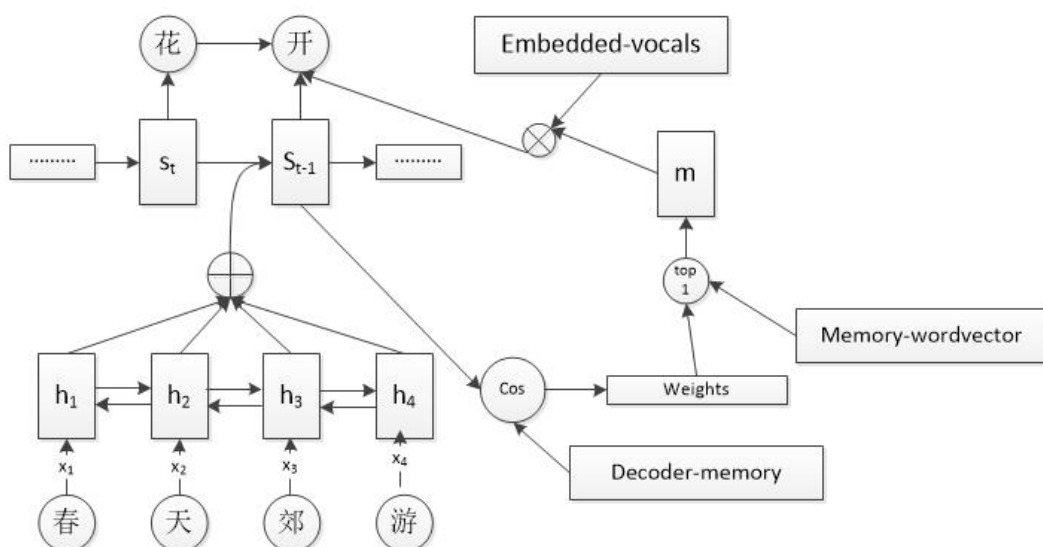
结论：并不能生成不同风格的诗词，因为最后得到的 m 对结果的影响很有限

改进三：基于decoder-memory的结合方式改进

decoder-memory是attention模型中decoder的部分生成，下面是decoder-memory的基本过程：

1. 与当前decoder的隐层通过cos的方式计算出memory中每个state对应的权重
2. 让计算出的权重
3. 选出权重最大的wordvector
4. 用这个wordvector与整个词表做点积

decoder-memory结构，如下图所示：



结论：能生成不同风格的诗词，以这种方式结合memory中的信息可以影响生成的结果

示例：输入关键字“胡”，用边塞诗和言情诗的memory分别得到的结果，如下

边塞诗memory：一身不复无穷地 蓟口城边古木风 夜半胡笳声里雪 何须马上送君翁

言情诗memory：一身不复蝴蝶去 玉髻无人莫道啼 十二年来何处士 妆儿未免却相迷