

# 实践课题详细开发计划

---

## 简介

嘻哈越来越受人们的喜爱，目前已有一些诗歌、歌词生成的工作。该课题属于文本生成任务，目前主流的方法有 variant seq2seq、variant GANs，其中 seq2seq 为端到端模型，配上 attention 机制能够达到不错的效果。而 GANs 的生成对抗能够很好学到数据的语义分布。Rap 歌词段落主题突出，节奏性强，部分语句押韵明显，如何让结合不同结构模型或者加入一些限制条件，使得生成的 rap 被人认可，是一个探索性很强的课题。

本次课程将从主流的深度学习两个模型开始入手，首先从 Seq2seq 入手，探索其变体对文本生成的提升。接下来探索 GAN 与其各种变式。如果时间充裕，可以结合强化学习尝试做真正人与机器的 battle。

## 本次课程的目标与计划

1. 基于目前最好的文本生成模型，做出很优秀的结果展示
2. 在学习过程中，由于本次课程时间短，而这个课题涉及的知识点较多，可以分为简易版，进阶版，终极版。简易版可以单独探究 Seq2seq 一个模型，并将各种变体尝试完美。进阶版可以尝试更多的模型，包括不限于所提供的 GAN 的变体，包括结合增强学习的 GAN 的变体。终极版可以用增强学习最后做 battle。
3. 充分利用 Github 上已有的一些写好的模型，达到在几周内快速掌握并实践的目的。（现提出的几个模型都已找好相应模型，并调试完毕）
4. 最终展示目标，由所挑的难易程度可以有三种评价标准：
  - a) 生成歌词（可读性强，上下文有连接，与人写的歌词差别不大）
  - b) 生成嘻哈歌词（带有嘻哈歌词独有的押韵（包括二押，三押等等）
  - c) 所做系统可以进行 battle（要求较高，输入对方的句子，生成回应）

## 资源准备

- 1) 环境搭建：建议用 Anaconda 建立虚拟环境统一管理  
Python: 3.6, Tensorflow-gpu: 1.8.0, Cuda: 8.0, CuDNN: 6.0
- 2) 硬件：UCloud 的配置足够
- 3) 数据集：
  - a) 嘻哈歌词：暂时有的嘻哈歌词有脏句，押韵未标明，数据集大小共有 2800 首

歌（**仍在增量爬取**），内容中英混杂（这个需要选手自己做预处理，需要选手挑出中文，繁简转换，关键词提取歌曲信息并去除等）。

b) 押韵规则：从网上找到的二押，三押等规则，呈现方式 “.txt “文件。

c) 歌曲音乐信号：**暂无**，需要爬虫工程师的支持，需要接触过信号的工程师提供音乐转信号的方案。（**按需求可选择做与不做**）

## 按周计划

### 实践课程开始前：

1. Python 能读写系统级代码
2. 深度学习入门与 Tensorflow 的使用
3. 能参与讨论和对比各个技术方案，并能调通所给的代码

### ● 第一周：

- Linux 环境中配置环境，包括 Anaconda, Cuda, Cudnn, Tensorflow 等。熟悉 Anaconda 的使用。
- 跑通 Tensorflow-gpu 的测试脚本。
- 读 RNN, LSTM, GRU, Seq2seq 的论文，用 Tensorflow 简单实现几个简单的模型
- 目标：熟悉 Tensorflow 的使用，熟悉 RNN 的几种变体，并应用在 Seq2seq 上

### ● 第二周：

- 读 Attention 的论文和各种变体，读 Seq2seq 的各种变体论文和应用方法。
- 讨论制定利用 Seq2seq 生成歌词的初步方案，要求可读性强，上下文有连接，与人写的歌词差别不大。
- 投入训练歌词开始尝试。在之前做好数据预处理的部分。
- 目标：组内工作协调好，一部分人做歌词预处理，一部分人研究算法，一部分人做需求分析和调研，安排好每个人的工作。

### ● 第三周：

- 加入对押韵的考虑，从句尾单押，到句内的二押，三押等。
- 读 GAN 的论文和各种变体，熟悉其结构与数学推导。
- 讨论制定需要尝试的 GAN 与相应的变体。
- 投入数据开始做实验，比对 Seq2seq 的结果，观察结果。做 error analysis。
- 目标：加入押韵机制，包括大量的 coding 部分。要求达到可允许范围的押韵即可。

## ● 第四周

- 尝试增强学习做 battle 的思路（用最简单的 Q-learning 即可），利用已有的生成模型做迁移学习。制定评价标准，即相应的 reward 机制。制定每一步 action 更新的方向，此步需要更多的思考。
- 如果达不到这一步，可以用之前的生成模型，尝试改成 battle 的形式。例如提取对方句子中的主题，生成我们相应的应对。
- 做一个简单的视觉呈现页面。
- 解决完善存在的问题，完善最终效果，做进一步的 error analysis。
- 目标：录制最终自己觉得最完美的 demo。

## ● 团队的 milestone：

- 1) 首先学习并调通且尝试 Seq2seq 与 GAN 的各种尝试，以我们对面试成员水平的了解。最少需要 2 周或两周以上。结合押韵的考虑，因为工程量较大，可能要做很多尝试，可能会占满三周的时间。
- 2) 因此 milestone 设为
  - ii. 学习结束后的第一周末：尝试 Seq2seq 及其所有变体
  - iii. 学习结束后的第二周末：尝试 GAN 及其所有变体
  - iv. 学习结束后的第三周（提交环节）：尽量结合押韵做进一步的改进
  - v. 如果有的组能进度比较快，可以尝试对 battle 的强化学习做进一步的尝试

## 分组方案

- 实践开始前，各自准备
- 实践课开始后，以 3-4 人为一个小组，10 个人分为三组。三组分别分为 1) 需求分析，error analysis 组 2) 算法分析组 3) coding 组。成员可看自己的实力选择进入哪一组，组内成员可互相帮助。每组一名组长，整队 10 人选出一人统筹规划。

## 相关学习资料

历史工作：

1. <https://www.jiqizhixin.com/articles/2017-05-22>
2. <https://blog.csdn.net/zhangzeyuaaa/article/details/77596348>

视频学习资料：

1. RNN1：<https://www.bilibili.com/video/av10590361/?p=36>

2. RNN2 : <https://www.bilibili.com/video/av10590361/?p=37>
3. Seq2seq: <https://www.bilibili.com/video/av9770302/?p=25>
4. GAN : <https://www.bilibili.com/video/av9770302/?p=15>
5. Reinforcement Learning: <https://www.bilibili.com/video/av10590361/?p=39>

## 论文：

1. RNN : A Critical Review of Recurrent Neural Networks for Sequence Learning. 第一篇  
太久了
2. LSTM: <https://dl.acm.org/citation.cfm?id=1246450>
3. Seq2seq : <https://arxiv.org/abs/1409.3215>
4. Attention : <https://arxiv.org/abs/1409.0473>
5. Attention : <https://arxiv.org/abs/1506.07503>
6. SkipThought: Skip-Thought Vectors
7. VAE : Auto-encoding variational bayes
8. GAN : [Generative Adversarial Networks](#)
9. ARAE: Adversarially Regularized Autoencoders
10. SeqGAN : [SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient](#)

## 诗歌生成：

1. Chinese Poetry Generation with Planning based Neural Network
2. Chinese Song lyrics Generation with Neural Attention-Based Model

## GAN的变种：

1. <https://zhuanlan.zhihu.com/p/26491601>

## 歌曲生成，例子：

1. seq2seq+attention 赵雷：  
<https://36kr.com/p/5064512.html><https://mp.weixin.qq.com/s/uYiq2knJ0rrzdpuvbdmWhw>
2. 用char-CNN生成 <https://github.com/leido/char-rnn-cn>

## 书单：

Deep Learning, Ian Goodfellow, Yoshua Bengio, Aaron Courville

课件：

text-generation-survey(附件中)