分布式智能软件课程项目

项目名称 基于深度学习的中国古诗生成应用

组长姓名 牛哲

成员姓名 刘雨瑶，胥雅雯，吴欣，曾沂江

学科专业 软件工程

培养单位 华南理工大学软件学院

报告提纲

1 摘要

我们实现了一个基于 word2vec 与 seq2seq 模型的古诗词生成模型，该模型可以根据用户输入的诗词译文生成对应的七言诗句。

2 引言

近年来，随着深度学习的不断发展，越来越多的研究尝试用深度学习模型生成艺术作品。例如使用深度对抗网络根据文字生成图像 [1]、基于深度学习生成乐谱 [2]、利用 LSTM 生成莎士比亚短诗 [3]。

古诗是中华民族的瑰宝。今天，仍然有很多人会阅读古诗。古诗引发人们的感情共鸣、传授人道理。不同于一般的文字序列生成，中文古诗不但对生成文字的逻辑有所要求，同时也要求所生成文字的押韵、对仗。

我们使用 word2vec 模型 [4] 对爬取的七言古诗及其译文进行编码，接着，使用 seq2seq 模型 [5] 训练出了根据意象生成七言古诗的七言古诗生成器。最后，我们基于上述模型，开发了一个简易的 Web 应用（华工小施）。

3 相关工作

我们参照了 Xiaoyuan Yi 等人使用 RNN 进行古诗生成工作 [6]。他们构建了一个可以根据前一句诗生成下一句诗的 RNN 编解码模型。除了汉字本身，汉字的声调也被用作特征。相比之下，我们建立了一个无需使用声调特征，直接根据译文生成古诗文的翻译模型。

4 数据获取

由于网络上并没有可以直接用于训练的开源数据集，我们通过编写爬虫获取训练数据。

4.1 源网站结构分析

经过搜索和筛选，我们选择了“古诗文网”作为数据获取的源网站。网站首页如图1。

图 1: 古诗文网

“古诗文网”是一家权威的古代诗文收录网站，包含了各个朝代各种类型的古代诗文。该网站前端页面书写规范工整，符合 W3C 标准，后端使用.NET 技术搭建，适合作为数据集构造的源网站。通过探索网站的 URL 相对结构，我们发现可以使用正则表达式得到所有相关的原始诗文列表 URL 地址。

而对于诗文的译文，网站有两种方式呈现，相应的我们可以通过两种形式获取。一种是直接访问诗文的标题，跳转到每首诗文所在的单个页面，解析网页中的译文所在的标签元素；另外一种是通过 AJAX 拿到异步返回的部分 HTML 代码，解析这部分代码得到网页的译文。出于开发效率和成本的考虑，我们选用了第二种方法。

4.2 爬虫框架设计

明确源网站 URL 结构后，我们开始着手设计爬虫，爬虫流程如图2所示。

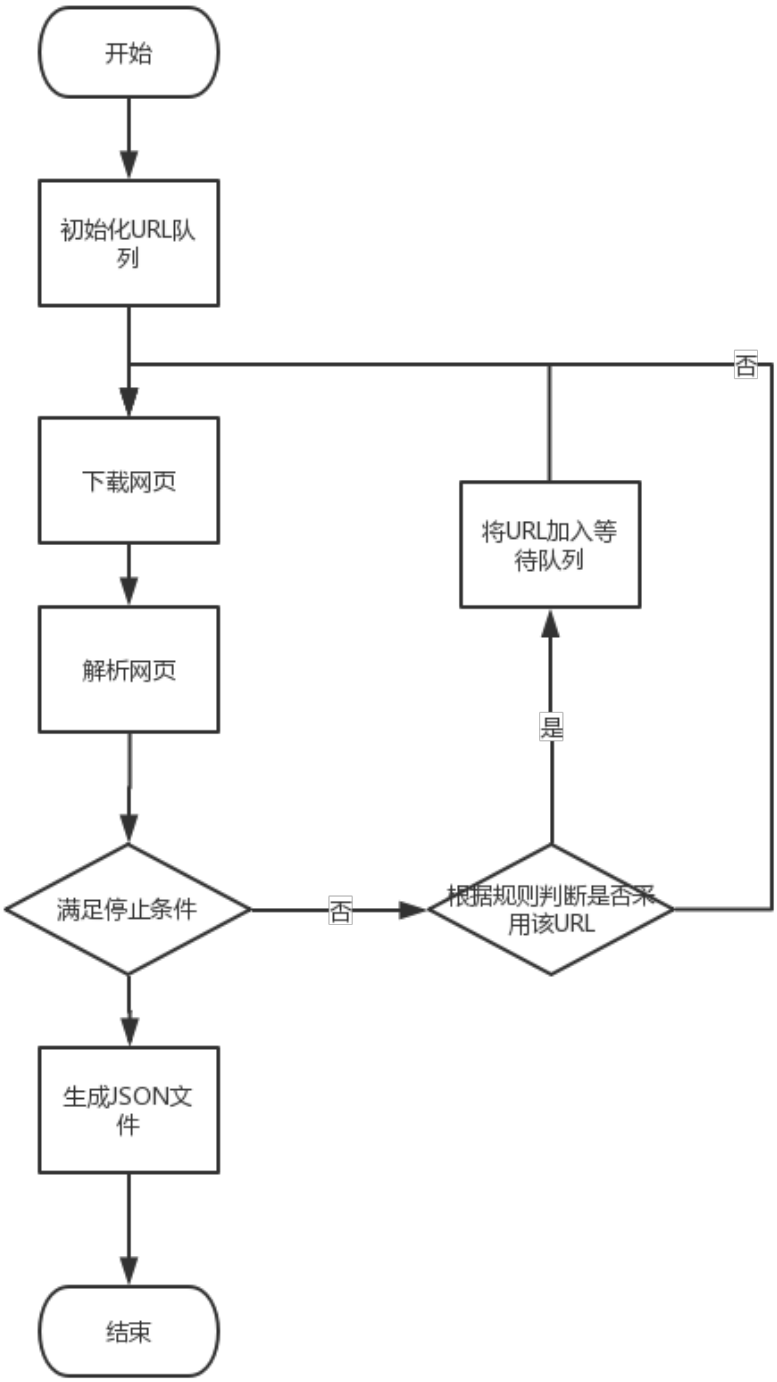

图 2: 爬虫流程图

首先进行 URL 的初始化，将种子 URL 加入到等待队列。然后爬虫从 URL 等待队列中取得任务 URL，根据 URL 下载网页，解析网页，获取超链接 URL。接着代码判断是否是符合既定规则的 URL，如果是，加入到等待队列。继续执行步骤，

直到结束条件停止。由于本次爬虫代码非通用爬虫，所以只需要保证特定的 URL 可以加入等待队列即可，无需对所有 URL 地址进行爬取和解析。

经过网络抓取到所有可用数据后，我们对数据进行分类整理，并将其导出到 JSON 文件。出于模型性能的考虑，我们将古诗文按照朝代、类型和字数进行简单的分类。按照朝代分为清代、明代、元代、金朝、宋代、五代、唐代、隋代、南北朝、魏晋、两汉和先秦；按照类型分为诗、词和曲，按照字数分为五言、七言和一般类。最终，我们采用了上述类别中的七言诗进行模型的训练。

5 古诗生成器

我们首先使用 word2vec 模型对爬取的七言古诗及其译文进行编码，接着采用 seq2seq 模型训练出根据意象（译文）生成七言古诗的七言古诗生成器。

5.1 汉字编码与解码

我们使用爬取的译文、古诗文作为数据集，训练出一个 word2vec 模型。该模型可以将汉字编码为一个 N 维的向量，即**字向量**。

解码时，我们通过搜索预制字向量集，找出与目标字向量点乘值最大（即相似度最高）的预制字向量所对应的汉字作为解码汉字。其中，**预制字向量集**为所有训练译文与古诗文中出现频率最高的 i 个字所对应的字向量的集合。

5.2 seq2seq 网络结构

我们将两个 LSTM 网络进行拼接，并采用注意力机制 [7]，构建出 seq2seq 网络（图3）。

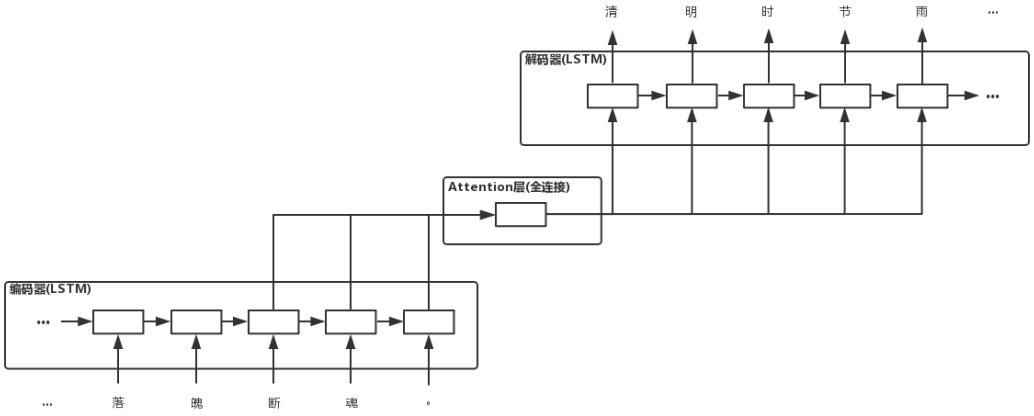

图 3: 采用注意力机制的 seq2seq 模型

该网络模型首先对译文进行编码，接着将译文编码后的信息进行注意力加权，最后将加权后的译文送入解码器，得到解码后的古诗文。图3展示了在使用译文“江南清明时节细雨纷纷飘洒，路上羁旅行人个个落魄断魂。”与古诗文“清明时节雨纷纷，路上行人欲断魂。”训练 seq2seq 模型的场景。值得注意的是，seq2seq 的输入和输出并非汉字本身，而是汉字所对应的字向量。

6 实际训练效果

6.1 word2vec

我们通过训练 word2vec 模型对汉字进行编码，训练的数据集来为译文与古诗文混合的文本数据集。

字	最临近字（最左为最近）
你	我, 它, 君, 尔, 局
无	不, 反, 蹶, 鸪, 鸦
城	楼, 戍, 轿, 禾, 蟋
山	岭, 辍, 巳, 豚, 源
声	语, 夜, 软, 唉, 类

表 1: 一些字向量相邻的字

经过 PCA 降维之后的最高频 100 字的字向量可视化（图4）。

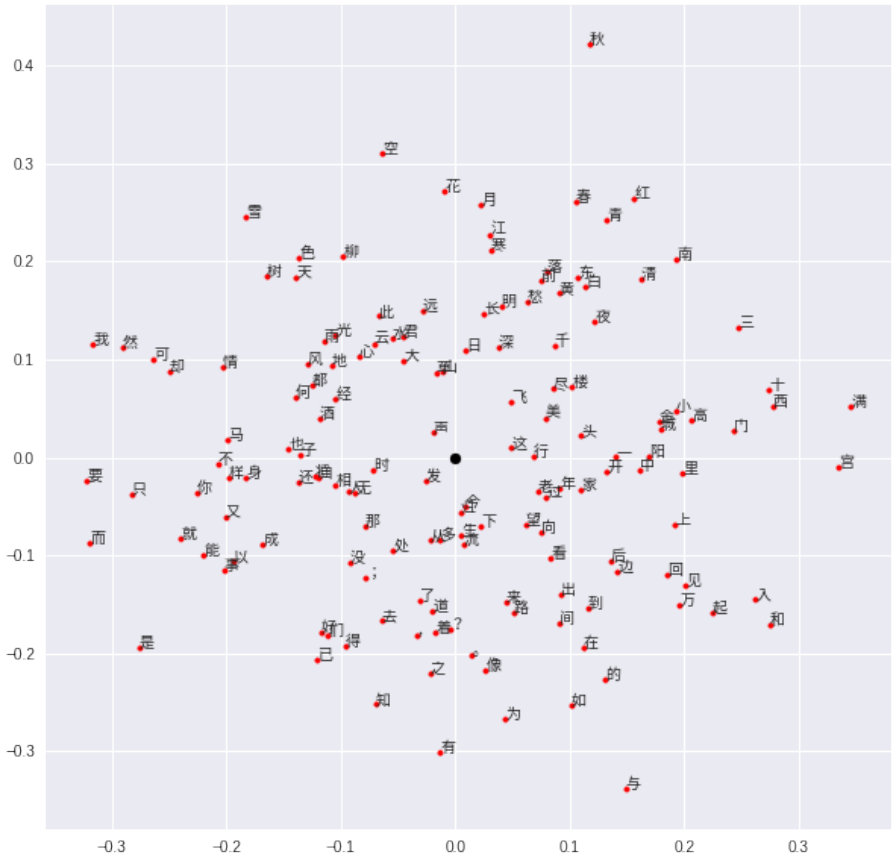

图 4: 最高频 100 字的字向量

6.2 seq2seq

我们先在小训练集（200 句诗）上，对模型进行了实验性的训练。损失（均方根误差）随着训练轮数的变化曲线如图5。

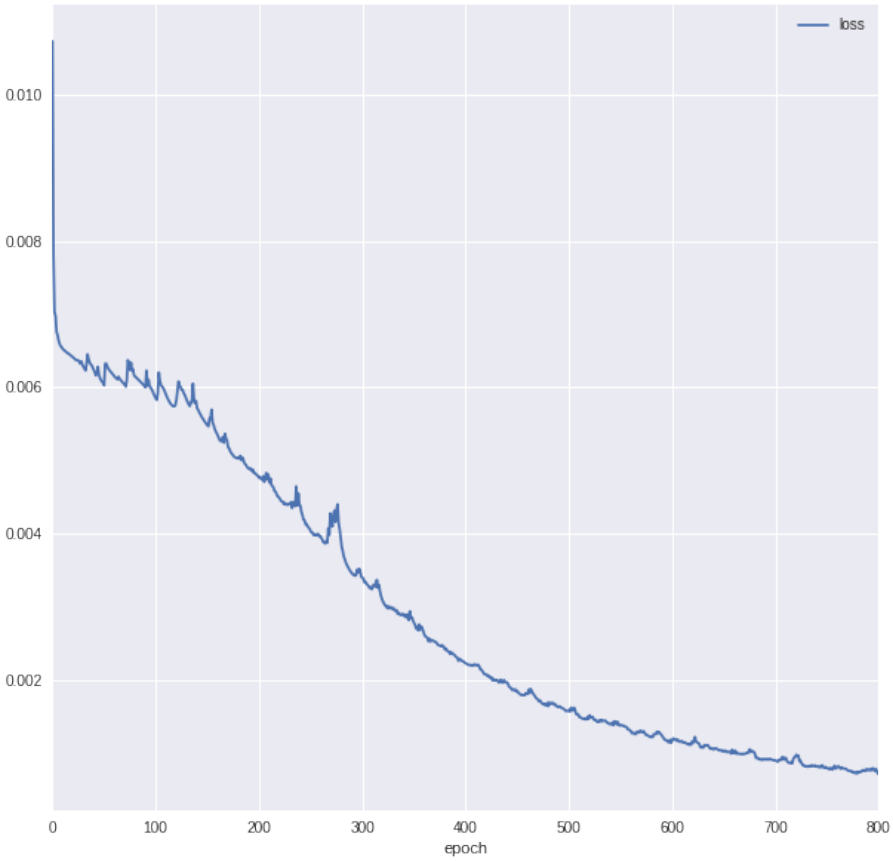

图 5: 损失曲线。

为了更加直观地反应训练过程，我们使用了四句话对模型进行验证。这四句分别为取自训练集的“江南清明时节细雨纷纷飘洒，路上羁旅行人个个落魄断魂。”(表2)，以及不含于训练集的“树上的鸟儿成双对，绿水青山带笑颜。”(表3)、“要找到我的宝物，师傅招牌必须保住。”(表4) 与“北方的冬天是物理攻击，南方的冬天是魔法攻击。”(表5)。

训练轮数	生成结果
0	个个饮饮，，，。。。。。。。。。
100	，春时。。，，，，，，，人。。。
200	时时时时纷纷纷，上上行人人断魂。
300	时时时节雨纷纷，路上行人欲断魂。
400	明时时节雨纷纷，路上行人欲断魂。
500	明明时节雨纷纷，路上行人欲断魂。
600	清明时节雨纷纷，路上行人欲断魂。
700	清明时节雨纷纷，路上行人欲断魂。
800	清明时节雨纷纷，路上行人欲断魂。

表 2: “江南清明时节细雨纷纷飘洒，路上羁旅行人个个落魄断魂。”（训练样本）

训练轮数	生成结果
0	个个饮饮，，，。。。。。。。。。
100	江江山。。，，，，，，江。。。。
200	江山山水水水尺，不看看看日日。。
300	江山山山片水山，不看看围无行时。
400	青山山水梨水色，孤看不牛无自云。
500	墙山山水梨如色，孤看江山无自云。
600	凤阶山水凤如天，借看黄肯无自云。
700	凤山耸水凤顾天，偏看黄肯对自云。
800	凤山耸水凤顾天，偏应黄思对照人。

表 3: “树上的鸟儿成双对，绿水青山带笑颜。”（测试样本）

训练轮数	生成结果
0	个个饮饮，，，。。。。。。。。。
100	春春春。。，，，，，，，，。。。。
200	春日知知春风风，我人人夜不春。。
300	春知知知知才来，花花宫园妆门人。
400	春知知知登常，，每花富酒妆人人。
500	至知灰酒病登，，每花潮月愁丝人。
600	至知灰欣时暮来，每花潮月愁笑人。
700	至知灰莫时谁日，每花兰月愁市人。
800	至知诗诗院谁日，每花潮月满市人。

表 4: “要找到我的宝物，师傅招牌必须保住。”（测试样本）

训练轮数	生成结果
0	个个饮饮，，，。。。。。。。。。
100	春春春。。，，，，，，日。。。。
200	春春春春春春色，不上日日日日。。
300	春春春春春日花，春风不看日日花。
400	春春春春色月，，人上不挂日夜山。
500	春春白春春色晚，人上秋挂日夜看。
600	春里白夜春来晚，人上镜镜日夜风。
700	春里即夜无来，，将舞秋镜日时风。
800	春里即夜无来晚，将舞晴门日夜风。

表 5: “北方的冬天是物理攻击，南方的冬天是魔法攻击。”（测试样本）

7 华工小施

7.1 功能简述

华工小施（图6）是一款简易的古诗词生成 Web 应用。它可以根据用户输入的意象“译文”，生成一句对应的古诗文。华工小施采用了训练集中的所有七言诗句（约 5000 句）作为训练集，训练出了最终模型。

图 6: 华工小施

7.2 技术架构

华工小施前端使用 semantic-ui，vuejs 开发，后台使用 flask 框架开发。前端将用户的意象（译文）文本发送至服务器，并在服务器返回生成古诗文后将其呈现给用户。后端则负责接受前端的译文文本，将其输入模型进行古诗文的生成，接着返回生成结果。小施的系统架构如图7所示。

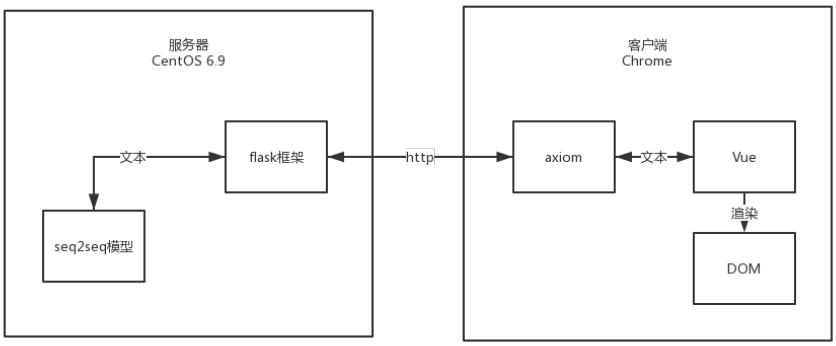

图 7: 系统架构图

8 工作总结

我们从网上爬取了古诗文本，利用这些文本训练出了一个古诗文生成器。这个古诗文生成器可以按照译文，生成相关的古诗文。最后，我们通过构建 Web 应用，部署了我们的模型。

参考文献

[1] Reed S, Akata Z, Yan X, et al. Generative adversarial text to image synthesis[J]. 2016:1060-1069.

[2] Sturm B L, Santos J F, Bental O, et al. Music transcription modelling and composition using deep learning[J]. 2016.

[3] S Xie, R Rastogi, Deep Poetry: Word-Level and Character-Level Language Models for Shakespearean Sonnet Generation.

[4] Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space[J]. Computer Science, 2013.

[5] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[J]. 2014, 4:3104-3112.

[6] Yi X, Li R, Sun M. Generating Chinese Classical Poems with RNN Encoder-Decoder[J]. 2016.

[7] Bahdanau D, Cho K, Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate[J]. Computer Science, 2014.