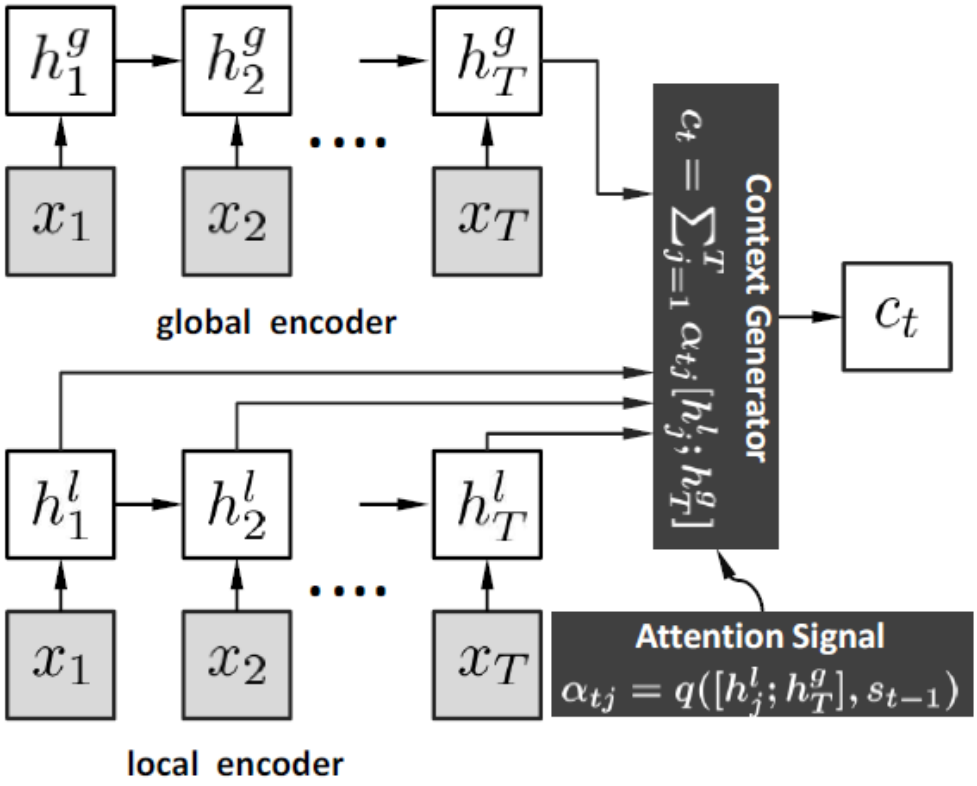
基于 RNN Attention 的生成式聊天模型

邢超（Chao Xing）
白子薇（Ziwei Bai）

2017/1/11

简介

本项目基于[1]进行序列对序列网络进行复现。使用的模型如图 1 所示。



图一 序列对序列模型图

数据处理

处理规则

数据的收集与初步处理请参考《聊天数据收集与清理建议》
为了方便起见，我们将对话对中的 Q 或 post 统一称为 post，A 或 response 统一称为 response。
为了使数据更加适合本模型，我们只保留符合以下某一条件的数据，并按条件不同分组：

- post 和 response 的词数量都大小 3，小于等于 5
- post 和 response 的词数量都大小 5，小于等于 10
- post 的词数量大于 10，小于等于 15；response 的词的数量大于 5，小于等于 10
- post 的词数量大于 15，小于等于 20；response 的词的数量大于 5，小于等于 10

数据处理程序

程序

bucket.py

运行示例

python buckey.py ../data/post ../data/response

使用说明

数据准备脚本

- **程序名称**
  - run_prepare.sh
- **功能**
  - 对数据进行分 bucket
  - 每组随机采样 500 句作为测试数据
  - 对训练数据进行补 MASK，根据之前对数据的分组，将每组数据补到该组数据词数量最大值。
  - 将通过词表示的补 MASK 后数据转化为通过词的 index 表示的数据
- **备注**
  - **建议训练时，**sample 100w **句进行训练**

训练脚本

- **程序名称**
  - run_train.sh
- **功能**
  - 训练模型并保存
- **使用示例**
  - sh run_train.sh　0
  - 参数‘0’代表实验运行指定的 GPU

测试脚本

- **程序名称**
  - run_test.sh
- **功能**
  - 根据训练得到的模型进行测试，测试模型在不同组数据下的表现， 并保存结果
- **使用示例**
  - sh run_test.sh　0
  - 参数‘0’代表实验运行指定的 GPU

核心程序说明

本次提交的 python 核心脚本分别为:simple_train_process.py，simple_test_process.py，toolbox。
其中：
rnn_local.py 为训练脚本，参数为：
-post:接受训练数据的 post 文件，文件内容是 post 数据中每个词的 index
-response: 接受训练数据的 response 文件，文件内容是 response 数据中每个词的 index
-vec:接受一个词向量矩阵,词向量需要进行 mean variance normalization
-dict：接受一个表示词和 index 对应关系的字典，文件类型是.pkl 文件，通过 cPickle 库读取。
-output：输出模型存放的位置以及名称。
-lr：学习速率
-batch:每个 batch 的大小。本文中所述的 RNN Attention 模型，采用 mini-batch 方法更新参数。
-epoch 模型训练的轮数。
-encoder：编码器的大小
-decoder：解码器的大小
使用示例：
python rnn_local.py -post ../data/data_all/post_filter -res ../data/data_all/response_filter -dict ../data/data_all/recut_word_to_index.pkl -vec ../data/data_all/word_matrix.npy -encoder 512 -decoder 512 -epoch 101 -batch 500 -show 1 -output ../model-new/model -lr 1e-5 >> run.log 2>> err.log

rnn_local.predict.py 为训练脚本，参数为：
-post:接受训练数据的 post 文件，文件内容是分好词的 post 数据
-res: 接受训练数据的 response 文件，文件内容是分好词的 response 数据
-post-len:post 补齐 mask 后的长度
-res-len：response 补 end 和 mask 后的长度
-dict：接受一个表示词和 index 对应关系的字典，文件类型是.pkl 文件，通过 cPickle 库读取。
-result：输出结果存放位置及名称
-encoder：编码器的大小
-decoder：解码器的大小
-model：预测所使用的模型

使用示例：
python ../src/rnn_local.predict.py -post /work5/baizw/Chatting_Model_RNN/data/data_new/P2R_v1_sample/P2R_post_0 -res /work5/baizw/Chatting_Model_RNN/data/data_new/P2R_v1_sample/P2R_response_0 -post-len 6 -res-len 6 -vec ../data/word_matrix.npy -dict ../data/recut_word_to_index.pkl ../data/recut_index_to_word.pkl -model /work4/chaos/RNN-ATT/model_1-10/$model -encoder 512 -decoder 512 -result ../result2/0_$model

实验设置

词向量准备

我们选择谷歌的开源工具 word2vec 对语料进行训练。

- 语料分布
  - 汇联 1.7G 语料，对话数据
- 分词
  1. 使用 jieba 分词工具，通过结巴分词中的默认词表对数据进行分词，允许使用 HMM 模型,得到一个词表 vocab1
  2. 将数据按字分开，得到另一个词表 vocab2
  3. 将 vocab1 中词频前 3W 的词，和 vocab2 中所有词合在一起，构成一个新的词表 vocab
  4. 使用 jieba 分词工具，根据 vocab 再次对数据进行分词，不允许使用 HMM 模型，我们训练词向量及后续模型的训练和测试都依赖于这一套分词结果。（该词表已提交）
- word2vec 训练参数设置如下表所示

Window	Iteration	Cbow	Size	Sample	negative	hs	binary
8	20	0 (skip-gram)	200	1e-4	5	0	0

参数设置

encoder:512
decoder:512

实验结果展示(model-290)

ID	Post	生成的 response	标准 response
1	但 我 悄悄 告诉 你	我 再也 不 相信 爱情 了	那 都 不 重要
2	安全感 是 自己 给 的	这 句 话 说 的 好	没 安全感 是 吧
3	说 你 爱 我	我 爱 你 你 永远 爱	我 我 我 我 说不出口
4	这 套 戏 要 看 有 惊喜	好 厉害 惊喜 你 都 知道	早 前 看 了 宣传片 还 不错
5	所以 说 不要 惹 女人 女人 是 很 可怕 滴	我 靠 这 句 话 太 有 水平 了	话 说 有 几个 女 淫 能 做到
6	他 在 保护 自己 的 朋友	看来 他 不够 爱 你 啊	看来 他 不够 爱 你 啊
7	中午 啦 吃饭 吃饭 吃饭 吃饭	吃饭 的 孩 纸 们 伤 不 起 啊	这 是 早饭 还是 昨天 的 晚饭
8	你 太 聪明 了 这 都 被 你 发现 了 不 简单	我 优点 多着呢 这 只是 冰山 一角	还 行 还 行 排 的 上前 三
9	据说 喜欢 在 句 末 发 这个 的 孩子 有 一颗 玲珑 婉转 的 心	我 也 是 我 也 是	我 有 一颗 玲珑 心 有 谁 懂得 珍惜
10	你 一生 中 看过 的 小说 中 最 希望 哪部 能 改编 为 电影	爱神 的 黑白 羽翼 盗墓 笔记	太 多 了 眼下 希望 是 百年 孤独
11	这 是 一个 从 上 厕所 到 坐 电 车 都 有 准备 的 女人	这 桀 骜 的 女人 真 危险	还 真 方便 佩服 这 位 大姐 太 有 才 啦
12	无论 我 此时 是 多么 的 彷徨 迷茫 最终 我 都 要 过 上 自己 想 要 的 生活	我 也 要 这样 的 生活	感觉 讲 到 心里 去 了 你们 呢
13	我 就 喜欢 看 哈 林 还 没 转身 前 纠结 像 吃 了 大便 的 那 种 表情	我 也 喜欢 我 也 喜欢	怎么办 又 想 看 快男 又 想 看好 声音
14	罗斯 训练 中 的 这个 球 有 让 你 更 怀念 他 的 折叠 暴 扣 吗	我 也 想 训练 求 方法	弹跳 明显 弱 了 一成 啊 难道 保守 实力
15	一起 来 填空 每天 摸 着 大 肚 肚 我 最 想 跟 宝宝 说 的话 是	我 想 我们 应该 会 成为 好 朋友 的	快点 跟 我 变 身 滚 出来

[1] Shang L, Lu Z, Li H. Neural responding machine for short-text conversation[J]. arXiv preprint arXiv:1503.02364, 2015.