自然语言处理
诗词生成系统文档

December 12, 2020

Professor：刘洋

小组：非必要不出校队
组员：余孙婕、邱艺境、汤胜中

1. 背景

中华文明源远流长，在几千年的文化长河中伟大的先人们为我们留下了不少文化瑰宝，其中占据重要地位之一的便是诗歌。诗歌通过对文字的高度凝练和加工，在表达深刻内涵，将文字高度抽象化的同时，将诗歌限制在规定的结构之中，因此一方面诗歌具有极高的文学艺术欣赏价值，另一方面由由于它严格的结构规则、丰富的内涵，被广泛地应用于自然处理领域的生成任务当中去。利用前沿的人工智能技术模拟诗歌的规则和使人的写作方式，生成主题明确、语句流畅、逻辑顺畅、内涵深刻的诗篇。诗歌生成的任务不仅可以让当代普通群众体会先人的智慧、辅助诗词文化研究，也借由对人类文字的高度逻辑化、认知抽象化、结构规一的产物——诗歌的研究，探索计算机理解人类语言和思维的方式，在宣扬中华传统文化的同时，促进自然处理领域文本生成任务的不断发展。 本文实现了一个诗词生成系统，利用多个深度学习模型实现诗词生成，并提供了交互性良好的 demo 界面进行展示。本文的组织结构如下：

1. 第 1 章介绍了本项目的背景；
2. 第 2 章介绍了小组成员及其完成的工作；
3. 第 3 章介绍了诗词生成任务的定义；
4. 第 4 章对该系统的模型进行了说明和实现；
5. 第 5 章对该系统实验结果进行了比较和分析，包括对使用数据集的介绍和基准系统的介绍；
6. 第 6 章是该系统的使用文档，包括对系统环境要求、部署和使用方式的介绍；
7. 第 7 章是对文本的总结。

2. 小组成员分工

本次项目的小组成员分工如表 1 所示。

表 1: 小组成员分工

成员	分工
余孙婕	负责 LSTM 模型的实现、基准系统准备、前后端 demo 系统开发和文档撰写
邱艺境	负责 bert 和 roberta 模型的实现，参与数据集收集、模型评估和文档撰写
汤胜中	负责 GPT-2 模型的实现、模型评估和文档撰写

3. 诗词生成定义

自动生成诗歌是实现计算创造力的重要一步。在诗歌生成文献中，生成器是在一个交互式的环境中运行的，用户首先向模型提供一组关键字，这些关键字表示概念，这些概念概述了主要的写作意图及其顺序。用户还负责为生成的诗歌选择特定的格式。例如，常见的格式是由四行句子组成的四行诗，或者是由八行句子组成的规范诗句。这个过程是互动的，作者可以不断修改术语来反映他的写作意图。诗歌生成是一个受限的文本生成问题，因为生成的诗歌中需要包含用户定义的概念。同时，它也可以是一个条件文本生成问题，对诗歌的文体特征有明确的制约作用。

[1] 在下面定义诗歌生成任务。给出一组关键字作为输入，这些关键字概括了作者的写作意图 $K = \{k_1, k_2, \ldots, k_{|K|}\}$，其中每个 $k_i \in V, i = 1, \ldots, |K|$ 是词汇表 V 中的一个关键词，目标是生成一首诗 $\mathcal{P} = w|w \in \Omega$，其中每个词 w 都是从候选词集 $\Omega = \{w|w \in \{K \cup \{V - K\}\}\}, K \subseteq P, P \subseteq \Omega$，以适应用户指定的诗歌格式约束。生成模型计算了行 $S_{i+1} = w_1, w_2, \ldots, w_m$ 的可能性，这些行由所有先前生成

的诗行 $S_{1:i}, i \geq 1$ 给出 (或者只有先前生成的诗行或词汇的 n-grams 给出)。

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

在 iPOET 中，诗歌写作被描述为生成性摘要框架中的约束优化问题 [2]。从一个大型人类写诗语料库中检索候选词以匹配用户意图，然后聚类以适应诗歌的格式、音调、节奏等要求。在多通道生成性摘要框架下，通过迭代替换项生成一行诗歌，使生成的诗歌与初始用户约束和诗歌偏好相匹配，最大限度地提高输出的相关性和连贯性。

4. 模型算法及其实现

4.1 LSTM

4.1.1 模型介绍

长短时记忆 (LSTM)[3] 网络是一种时间循环神经网络，应用特殊的方式存储历史信息，以前梯度较大的历史信息不会被简单地抹去，一定程度上克服了 RNN 中梯度消失带来的局限性。LSTM 具有一个隐藏层，并且每个隐藏层节点都被修改为包括一个具有固定权重的自连接递归边的存储单元，该单元存储长时间的信息。时刻 t 的 LSTM 跃迁方程为：

$$\begin{aligned}
i_t &= \sigma(W^{(i)}x_t + U^{(i)}h_{t-1} + b^{(i)}) \\
f_t &= \sigma(W^{(f)}x_t + U^{(f)}h_{t-1} + b^{(f)}) \\
o_t &= \sigma(W^{(o)}x_t + U^{(o)}h_{t-1} + b^{(o)}) \\
u_t &= \sigma(W^{(u)}x_t + U^{(t)}h_{t-1} + b^{(t)}) \\
c_t &= i_t \odot u_t + f_t \odot c_{t-1} \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}$$

其中，存储单元 c_t 由一个具有内部隐藏状态 h_t 的节点和一系列门组成，包括输入门 i_t 控制每个 LSTM 单元的更新量，忘记门 f_t 控制前一个存储单元被遗忘的程度，输出门 o_t 控制内部存储器状态的暴露，x_t 是当前时刻 t 的输入，σ 表示 sigmoid 函数，$\odot$ 表示元素乘法，U, W 是学习的权重矩阵。

总的来说 LSTM 是 RNN 的一个优秀的变种模型，继承了大部分 RNN 模型的特性，同时缓解了 RNN 梯度回传过程由于逐步缩减而产生的梯度消失问题。因此，LSTM 非常适合用于处理与时间序列高度相关的问题。但与此同时，LSTM 也存在一定的局限性，主要是无法进行大规模的并行处理，在并行处理任务上存在天然劣势。

4.1.2 模型实现

为实现诗词的自动生成，模型实现主要包括以下步骤：

1. 引入 Word2Vec 预训练模型对数据集进行预处理；
2. 主题词聚类；
3. 模型训练；
4. 主题词输入处理。

4.1.3 数据预处理

数据预处理包括对脏数据的剔除、对数据的分类和格式化，将数据按照字数和句数分为四类，分别为：五言绝句、七言绝句、五言律诗、七言律诗。将不满足这四类要求的诗歌剔除，去除诗歌标题，仅保留主体内容部分，并为诗歌开头和结尾添加标识符^[开头]、$[结尾]。

语料库我们选择的是中文词向量语料库 Chinese Word Vectors[4] 中的四库全书部分，作为诗歌词向量 Word2Vec 预训练模型，利用该预训练模型，找到数据集中每首诗的中心词作为该诗的主题词，与诗歌内容一起结构化地保存为 JSON 文本。

最后为数据集构建字 id 到 token 的映射表。

4.1.4 主题词聚类

观察到很多主题词的意象极为接近，诸如：“雪”和“冬”、“山”和“水”、“离”和“别”等，这些相近的词不仅意象极为接近，也经常成对地出现在一首诗中。因此我们可以认为，这些主题词表达的是一个意思，可以被聚集地分为一类。

为了对主题词进行聚类，将训练集的诗歌主题词抽取出来，利用 Word2Vec 模型转换成词向量，对词向量利用 K-Means 算法进行聚类，值得注意的是，聚类的时候应该尽可能保证样本均匀的分布在各类中，而尽量避免样本集中于某一类。由于训练集的规模不大 (2098 首)，故最终将诗歌分为 12 类，数量分别为：258/197/278/299/259/194/250/225/218/194/202/334。

抽取各类主题词中的中心词，分别为：柴、佯、尸、般、冶、侠、菑、卤、颉、毛、莹、沃。

4.1.5 模型训练

我们设置三层网络，分别为 Embedding、LSTM、TimeDistributed。BatchSize 为 8，共训练了 80 轮，损失大约在 0.3∼0.4 之间。

4.1.6 主题词输入处理

对输入的主题词，首先通过 Word2Vec 模型转换成词向量，利用余弦相似度找到 12 个主题词的中心词中与该主题词词向量最接近的中心词，并将该中心词对应的类号作为模型的输入。

4.2 BERT

4.2.1 模型介绍

BERT 的全称是 Bidirectional Encoder Representation from Transformers [5]，即双向 Transformer 的 Encoder，因为 decoder 是不能获要预测的信息的。模型的主要创新点都在 pre-train 方法上，即用了 Masked LM 和 Next Sentence Prediction 两种方法分别捕捉词语和句子级别的 representation。

BERT 提出的是一个框架，主要由两个阶段组成。分别是 Pre-training 以及 Fine-Tuning**??**。其在 NLP 业内引起巨大反响，认为是 NLP 领域里程碑式的进步。BERT 模型在机器阅读理解顶级水平测试 SQuAD1.1 中表现出惊人的成绩：全部两个衡量指标上全面超越人类，并且还在 11 种不同 NLP 测试中创出最佳成绩，包括将 GLUE 基准推至 80.4%（绝对改进 7.6%），MultiNLI 准确度达到 86.7%（绝对改进率 5.6%）等。

BERT 有两个版本，$BERT_{BASE}$（L=12 H=768 A=12，与 GPT 规模相当）和 $BERT_{LARGE}$（L=24 H=1024 A=16）。

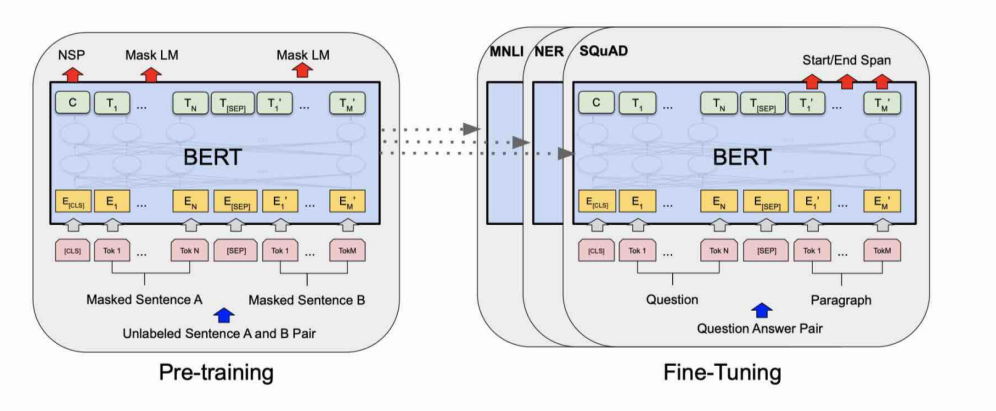

图 1: BERT

4.3 RoBERTa

4.3.1 模型介绍

RoBERTa [6] 模型是基于 BERT 的一种改进版本。是 BERT 在多个层面上的重大改进。

RoBERTa 在模型规模、算力和数据上，主要比 BERT 提升了以下几点：

更大的模型参数量（从 RoBERTa 论文提供的训练时间来看，模型使用 1024 块 V 100 GPU 训练了 1 天的时间）

更多的训练数据（包括：CC-NEWS 等在内的 160GB 纯文本）

此外如下所示，RoBERTa 还有很多训练方法上的改进。

1. 训练更长时间，使用更大的 batch_size，更多的数据
2. 删除 next sentence prediction 任务
3. 在更长的序列上进行训练
4. 动态改变训练数据的 masking 模式

4.3.2 Whole word masking

Whole Word Masking(wwm)，暂翻译为全词 Mask 或整词 Mask，是谷歌在 2019 年 5 月 31 日发布的一项 BERT 的升级版本，主要更改了原预训练阶段的训练样本生成策略。简单来说，原有基于 WordPiece 的分词方式会把一个完整的词切分成若干个子词，在生成训练样本时，这些被分开的子词会随机被 mask。在全词 Mask 中，如果一个完整的词的部分 WordPiece 子词被 mask，则同属该词的其他部分也会被 mask，即全词 Mask。

需要注意的是，这里的 mask 指的是广义的 mask（替换成 [MASK]；保持原词汇；随机替换成另外一个词），并非只局限于单词替换成 [MASK] 标签的情况。

同理，由于谷歌官方发布的 BERT-base, Chinese 中，中文是以字为粒度进行切分，没有考虑到传统 NLP 中的中文分词（CWS）。该方法将全词 Mask 的方法应用在了中文中，使用了中文维基百科（包括简体和繁体）进行训练，并且使用了哈工大 LTP 作为分词工具，即对组成同一个词的汉字全部进行 Mask。

4.4 BERT 和 RoBERTa 的实现

4.4.1 数据集

我们使用了从网上收集到的约 43000 首诗作为训练文本。

4.4.2 预训练模型

我们分别使用了 BERT-base, Chinese 和 RoBERTa-wwm-ext, Chinese [7] 作为预训练模型，这两个模型都是 base 模型，RoBERTa 使用了 wwm 和 EXT 数据进行预训练，EXT 数据包括：中文维基百科，其他百科、新闻、问答等数据，总词数达 5.4B。

4.4.3 数据预处理

包含格式化数据，分词，去停用词，特征提取，构建特征语料库。我们过滤掉出现次数少于 5 次的低频词，按词频排序后只保留词列表，构建新的 token 到 id 的映射和新词表。最后将特殊词和数据集中的词添加到词典中，使用新词典重新建立分词器。

4.4.4 模型微调

我们使用 Adam 学习器，学习率设置为 0.0001，使用交叉熵作为损失函数，进行模型的微调训练。

训练 20 轮之后，BERT 模型的损失下降到 0.6666，RoBERTa 模型的损失下降到 0.5371，虽然后续还有优化空间，但我们发现继续训练损失大概能降到 0.4 左右，此时模型把训练文本记住了大部分，即只会背诗。然而我们希望我们的模型具有一定的创造性，所以经过我们在每一轮训练之后随机输出一首诗，发现在 20 轮左右模型能作出比较好的诗。

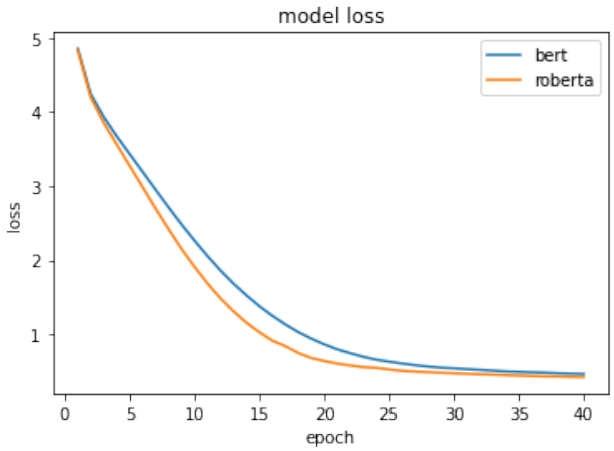

图 2: Loss

4.4.5 模型输出

我们的模型可以随机生成一首诗，也可以根据给出的词句进行续写，还可以写藏头诗。

如果给出词句，我们将其转为 token，然后不断与模型预测出的 token 扩展到一起作为输入，然后保留第一个样例最后一个 token 预测的不包含 [PAD]、[UNK]、[CLS] 的概率分布，对其进行逆序排列，对前 100 的概率进行归一化处理，然后按照预测出的概率从中随机选择一个词作为预测结果。最后，如果输出格式符合要求，我们输出结果，否则重新进行生成。如果没有给出词句，我们使用一个空字符串作为输入。

要生成藏头诗，需要给出若干个字组成的字符串，然后我们按照上面的方法对每个字分别进行续写，如果输出格式符合要求，我们输出结果，否则重新进行生成。

4.5 GPT-2

4.5.1 模型介绍

GPT-2 是 OpenAI 在 2019 年发表的模型。其前身是 OpenAI 在 2018 年发布的模型 GPT。GPT-2 和 GPT 在本质上没有太大差别，都是基于 transformer 来实现。transformer 的 self-attention 机制能够很好利用字词的上下文特征，并且通过深度的叠加来得到内容的长跨度下的相互影响。另一个重要的地方在于，transformer 结构相比于 RNN 结构其并行化更好，具有更高的训练效率。而 GTP-2 与 GPT 的主要区别在于其增加了训练的数据量和模型的参数量，使得模型具有更好的效果。

GPT-2 的模型按照参数大小分可以分为 4 种 small、medium、large 和 extra large，本任务中使用的是最小的模型 small，具有 117M 的参数量，模型体积大小约 500Mb。

4.5.2 模型实现

本任务是为了生成能够根据要求的体裁和关键词生成相应的诗歌。大概包括以下步骤：

1. 数据预处理：包括处理数据的 bias 和生成对应格式的训练集
2. Tokenizer 处理：加入新的 token
3. 模型 fine tune：包括调参、训练和评估
4. 模型测试

整个任务流如下图所示：

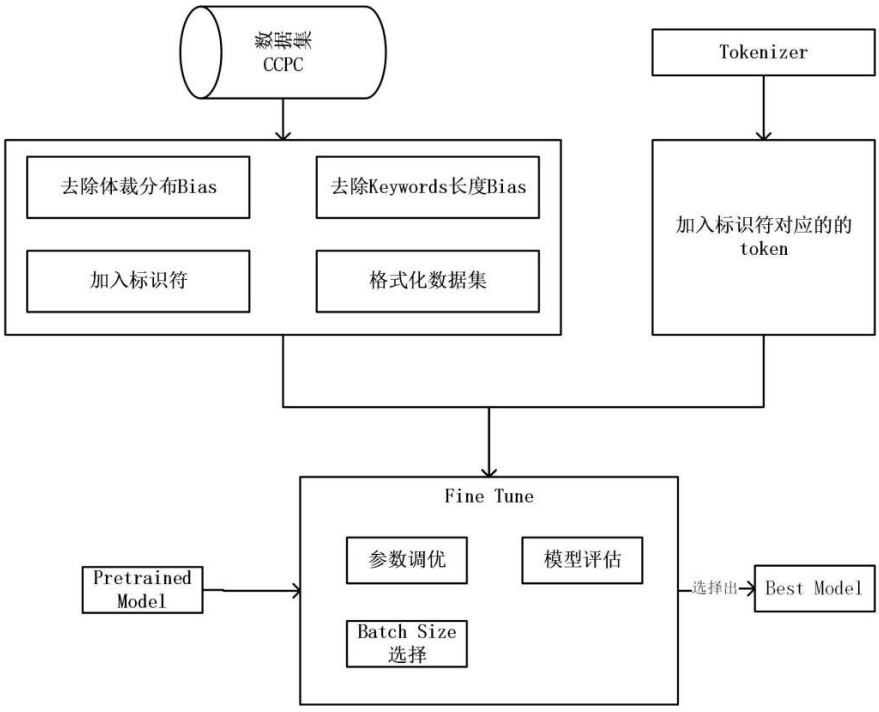

图 3: GPT-2 模型实现任务流

4.5.3 数据预处理

本任务使用的是清华大学的 THU-CCPC(THU Chinese Classical Poetry Corpus) 数据集，其中的诗歌都是五绝、七绝的体裁。数据集中的数据包含了 dynasty、author、content、title 和 keywords 五个字段。对于本项任务而言，需要用到的是其 content 和 keywords。

对于本任务而言，目的是根据体裁和 keywords 来生成对应的诗歌，因此设计数据集每个数据的格式如此：“[TYPE]type[CP] 词牌 [THE]keyword[CLS]content[SEP]”，其中“...”的字段为相应的数据内容。

这样设计的理由为：

1. 使用不同的标识符，有利于模型区别，减少歧义，提高模型训练速度和效果

2. 对 content 的开始沿用预训练模型中的“[CLS]”，结束使用“[SEP]”，提高模型训练速度和效果

需要注意的是，type 字段由于训练集的局限，目前只支持“五绝”和“七绝”，而由于没有词牌项，所以所有数据都填入“[NULL]”token，留待未来扩展使用。

原数据集对于本任务而言有两个 bias，一个在于其中七绝的诗歌数量远大于五绝的诗歌数量，大概是 4:1，因此模型容易偏向于增加七绝的生成概率来减少 loss；另一个在于其 keywords 都是多个，而且多为 3 或 4 个，而使用时模型的 keywords 输入个数是不一定的。

对于第二个 bias，可以考虑将每个数据增加为 n_keywords 个，其中第一个数据有一个 keyword，的个数据有两个 keyword，以此类推。keywords 的选择方式为从全部的 keywords 中采样以保证随机性。这样就得到不同长度 keywords 的数据。

对于第一个 bias，最简单的想法是将五绝的诗句复制 4 倍，对两个 bias 的处理会使得五绝的每个诗句被复制 12~16 倍左右，这样可能导致数据的重复性太高。发现到到两个 bias 之间恰好的数据关联性，可以想到将每个七绝的诗句的 keywords 采样出 random(n_keywords) 个 keyword 作为 keywords 标签。这种分策略的处理方式使得五绝的数据内容只增加了 3~4 倍，而七绝的数据没有增加，大致使得数据量的对比达到了平衡。经过处理后的数据集中，七绝诗句数量/五绝诗句数量 =1.14，大致达到了平衡。处理效果如下

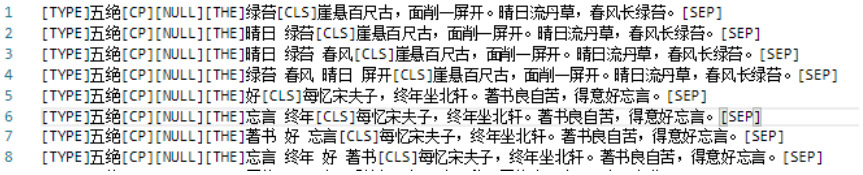

图 4: 五绝诗句的处理结果，每个诗句增加了 n_keywords 倍的数据

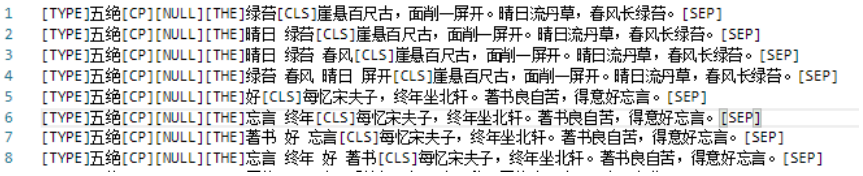

图 5: 七绝的处理结果，每个诗句的 keywords 通过采样变成了 random(n_keywords) 个

4.5.4 Tokenizer 的修改

为了减小训练量，避免对 word embedding 部分的重建，对于 Tokenizer 只增加了任务中新增的 token：“[TYPE]”、“[CP]”、“[NULL]”、“[THE]”

4.5.5 Fine Tune

预训练模型基于 transformers 库中针对诗歌数据库预训练好的 GPT2 模型“uer/gpt2-chinese-poem”，使用 transformers 库进行 fine tune

Batch size 的选择有利于减少模型在训练时参数的梯度出现反复抵消的情况，提高训练效率，碍于训练显存大小，本任务中将 Batch size 设为 80。

Learning rate 的选择有利于加快训练速度和提高训练效果，本任务中，经过实验，选择 learning rate=5e-4 时，模型 loss 的下降速度有较好的效果。

5. 实验结果与分析

5.1 基准系统

我们选择的系统为清华大学自然语言处理与社会人文计算实验室制作的九歌——人工智能诗歌写作系统 [8]。整个系统分成编码-解码网络和记忆读写部分。其中，编解码网络用 GRU 实现。九歌的两个特点是：

(1) 引入了两个约束向量，a 和 v，其中 a 用于约束生成诗句不脱离主题，v 用于约束生成诗句之间的内在联系。

(2) 工作记忆机制的引入，心理学上工作记忆 = 短时记忆 + 逻辑推理，系统的短时记忆体现在仅记忆前两行的内容，逻辑推理体现在两个约束向量（不断更新，包含了更早生成的信息）上。

5.2 数据准备

为了对本系统的生成结果进行测试，随机由 LSTM、GPT-2、BERT、RoBERTa 生成 100 首左右的五言绝句和七言绝句。其中，GPT-2、BERT 和 RoBERT 都有很好的泛度可以随机生成多首诗歌，但 LSTM 由于缺少较好的泛度，在不加约束控制下，无法生成多首诗歌。因此，我们随机地选择 ID，找到语料库中对应的 token，以该 token 为开头，为 LSTM 生成诗歌。

为了与基准系统进行比较，我们同样需要为九歌生成一定数量的诗歌，值得注意的是，九歌系统并为提供随机生成诗歌的接口，因此我们同样选择随机数 ID，找到对应的 token 作为九歌诗歌生成的主题关键词。由于九歌接口对 user_id 进行了加密，我们选择 puppeteer 工具包抓取网页信息。同样生成一百首左右的五言绝句和七言绝句。

5.3 BLEU

[9] 最先将四句古诗的生成看成是一个 SMT 问题，并且首次使用 BLEU 来衡量四句古诗的生成质量。虽然在 [9] 中的实验体现了 BLEU 的衡量效果与人类评价有较好的一致性，但是使用 BLEU 来衡量诗歌的效果仍然具有较大争议。在本实验中分别测试了 LSTM、GPT-2，BERT、RoBERTa 以及九歌模型的 BLEU 值，结果如下图所示：

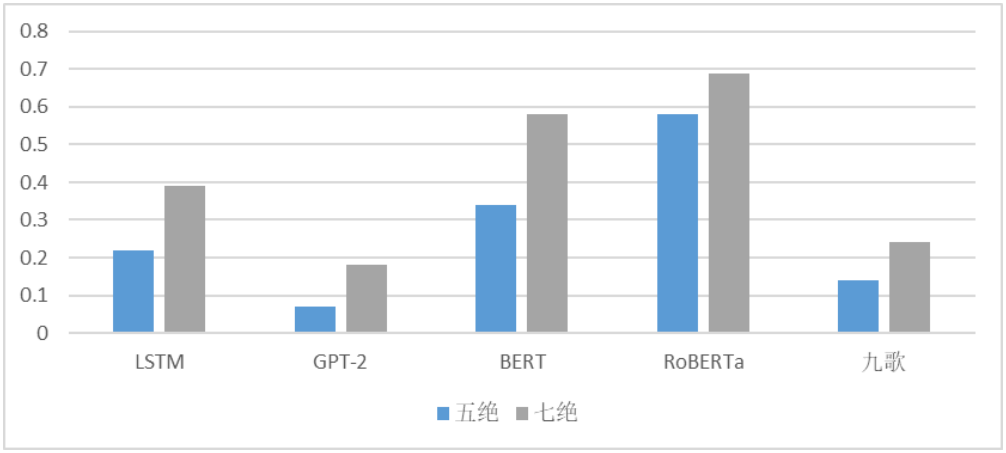

图 6: BLEU 值

5.4 Perplexity

困惑度是交叉熵的指数形式。和交叉熵一样都可以用来评价语言模型的好坏。对于测试集其困惑度越小, 准确率也就越高，语言模型也就越好。

对于一个平滑的 $n-gram$ ，其概率为 $p(w_i|w_{i-n+1}^{i-1})$，可以计算句子的概率：

$$p(s) = \prod_{i=1}^{m} p(w_i|w_{i-n+1}^{i-1})$$

假定测试语料 T 由 l_T 个句子构成 $(t_1, \ldots, t_{lT})$ ，则整个测试集的概率为：

$$p(T) = \prod_{i=1}^{l_T} p(t_i)$$

模型 $p(w_i|w_{i-n+1}^{i-1})$ 对于测试语料的交叉熵：

$$H_p(T) = -\frac{1}{W_T} \log_2 p(T)$$

其中，W_T 是测试文本 T 的词数。

模型 p 的困惑度 $PP_p(T)$ 定义为：$PP_p(T) = 2^{H_p(T)}$

表 2: 困惑度

BERT	RoBERTa	GPT-2	LSTM-五绝	LSTM-七绝
1.95	1.71	2.39	3.39	12.64

5.5 韵律和平仄分析

诗歌是一种高度浓缩的符合一定结构规范的文体形式。机器自动生成的诗歌也应该需要符合一定的规范，换句话说，作为诗歌这个文体，结构规范是最基本的要求，其次才是意境和意象。

格律诗有三大要素：平仄的抑扬；押韵的和谐；对仗的工整。

平仄：即声调。古代汉语有四个声调：平声、上声、去声和入声。按现代汉语拼音来分：一声为阴平，二声为阳平，三、四声为仄声。

押韵：即把同一韵部的字，按平仄规律放在规定句式的句尾，也叫韵脚。韵脚的位置是固定的。首句入韵的绝句有三韵脚，不入韵的只有两韵脚；首句入韵的律诗有五韵脚，不入韵的只有四韵脚。首句除外，韵脚都安在偶句的末尾。

为了研究生成诗词的结构是否符合规范，我们找到了诗歌吾爱网对我们的诗歌进行结构上的分析。得到如下结果。

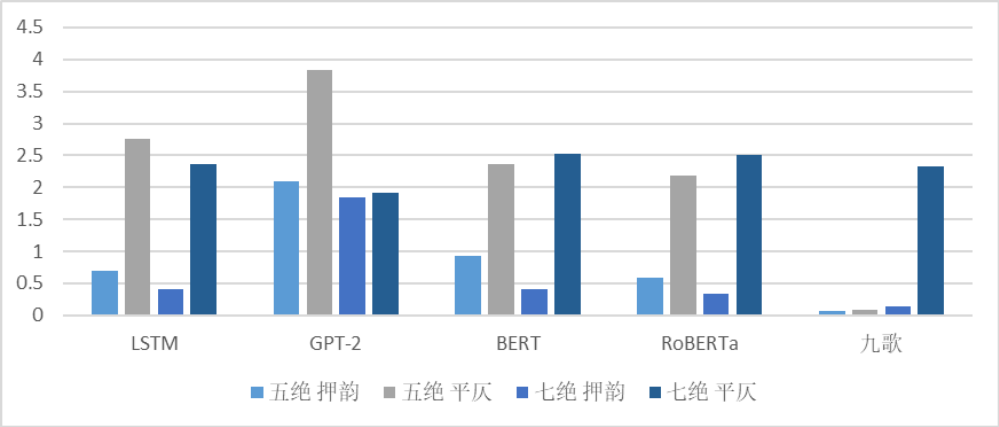

图 7: 韵律和平仄分析

5.6 GPT-2 的体裁与关键词分析

根据 [10] 中的测试指标，首先测试了模型对于指定体裁的生成正确率，结果为：

表 3: 体裁正确率

五绝	0.87	七绝	0.83

另一方面，由于此任务根据关键词来生成诗句，因此需要做根据关键词的生成效果衡量。考试到有时候模型的生成结果，有时不会让关键词以刚好邻接方式出现或者让关键词中的每个字全部出现。因此以关键词中的每个字为单位，测试了关键字生成率。因此测试中定义了关键字生成率这个指标来衡量：关键字生成率 = 生成诗中包含的关键字/关键词中的字数。

通过实验，还可以发现，一些频率不高的关键词占了所有关键词的很大一部分，比如在 CCPC 的 test 数据集中，关键词频率小于等于 4 的关键词占到所有关键词的 1/3 之多，而这些低频率的关键词由于在训练中更少出现，会降低模型的表现。另一方面，模型在生成过程中是否采取采样策略也会影响模型的表现，因此我们对这几个因素的影响做了分别的测试，结果如下：

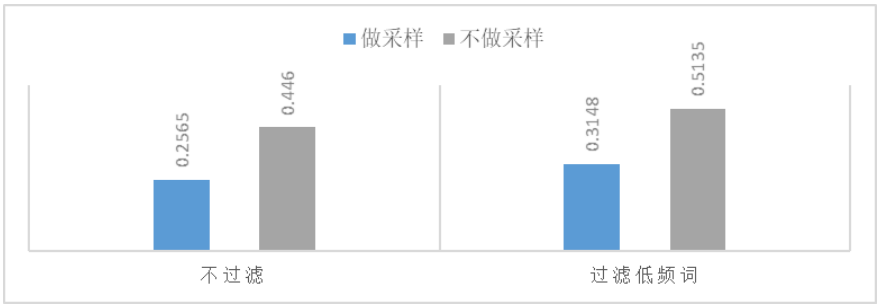

图 8: 不同因素影响下的模型关键字生成率 (过滤频率阈值取 4)

5.6 结果分析

一般来说，自然语言处理系统的主要方法为基于内在评价和基于外在评价，内在评价注重目的，外在评价侧重系统功能。我们对该系统所做的评估工作基本是针对于外部评价的，内部评价一般会使用专家人工评估的方法，由于条件限制，我们的内部评估模块暂缺。针对系统的功能特点，我们找了三个

维度对系统进行自动化的评价，分别是：BLEU 分数代表的生成质量、困惑度评价模型准确度、韵律和平仄分析结合语言学知识评估文本结构。为了衡量关键词生成诗歌效果，特别地，我们针对 GPT-2 模型，比较了不同条件下的模型关键字生成率。结合以上四个维度的分析，我们可以以下结论：

1. BERT 模型和基于 BERT 的改进模型 RoBERTa 在 BLEU 中得分较高，且 LSTM、BERT 和 RoBERTa 模型的 BLEU 得分均比九歌生成的诗歌得分高。一定程度上说明我们的模型与标准诗歌集的匹配程度更好，也即准确度更高。

2. BERT 模型和基于 BERT 的改进模型 RoBERTa 的困惑度也较其他模型较低。

3. GPT-2 模型在诗歌结构方面与其他模型相比较差，同样很明显的特点是，在韵律和平仄分析模块，九歌系统生成的诗歌体现出绝佳的性能。

4. 在 GPT-2 基于关键字生成诗歌的时候，过滤低频词比不过滤关键字生成率更高，不做采样比做采样关键字生成率更高。

6. 环境部署

6.1 环境要求

本项目的环境要求为：

- 操作系统：Windows 10、Mac OS、Linux
- Python：3.6+
- @vue/cli：4.5.9
- 机器学习库：
    - tensorflow-gpu：2.1.2
    - bert4keras:0.9.3
    - keras：2.3.1 | 2.4.3
    - h5py：2.10.0
    - transformers: 4.0.1

6.2 安装步骤

具体步骤如下：
1. python main_bert_roberta.py // 启动服务器
2. python main_lstm_gpt.py // 需要进入 python 虚拟环境，虚拟环境安装依赖 keras==2.4.3
2. cd ./poetry-generation // 进入子文件夹
3. yarn install // 安装 package.json 里所有包
4. yarn serve // 启动 vue-cli 项目
5. 浏览器输入地址 http://localhost:8080/#/，即可访问 demo 页面

6.3 使用介绍

项目一共实现了 4 个模型以作对比：LSTM、BERT、RoBERTa、GPT-2。

运行服务并访问相应的地址即可进入前端交互界面。选择好题材和生成方式，并按需在输入框键入提示词后，点击搜索框右侧的搜索按钮即可生成对应要求的古诗。在七言绝句体裁和随机生成模式下的生成及对比效果如下图所示：

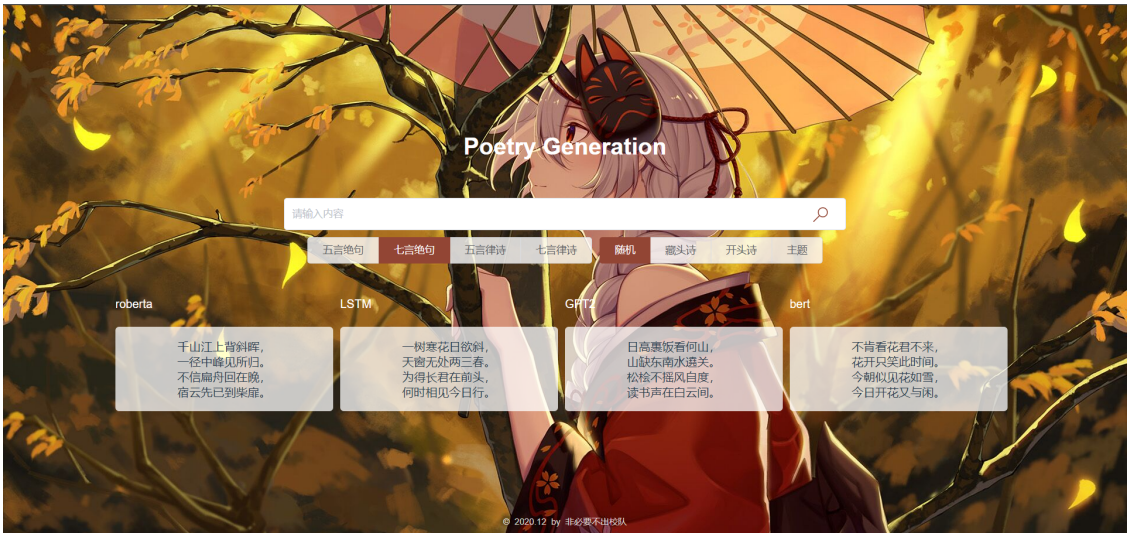

图 9: demo 界面

7. 总结

本工作中尝试将 NLP 领域的几个重要模型应用到中文古诗生成任务中，既有经典的模型 LSTM，也有近来热门的 BERT 和 GPT-2 模型，具有很好的代表性。文中描述各个模型在实现和训练方面的细节以及对应的评测结果对比，展现了各个模型之间的特点和优劣。基于上述工作成果，还实现了相应的 WEB 交互系统，可以方便展示实现的各个模型所具备的功能，以及对比不同模型在同一任务下生成结果，供使用者评价和赏析。

参考文献

[1] Cristina Garbacea and Qiaozhu Mei. Neural language generation: Formulation, methods, and evaluation, 2020.

[2] Rui Yan, Han Jiang, Mirella Lapata, Shou-De Lin, Xueqiang Lv, and Xiaoming Li. I, poet: Automatic chinese poetry composition through a generative summarization framework under constrained optimization. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, page 2197–2203. AAAI Press, 2013.

[3] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[4] Shen Li, Zhe Zhao, Renfen Hu, Wensi Li, Tao Liu, and Xiaoyong Du. Analogical reasoning on chinese morphological and semantic relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 138–143. Association for Computational Linguistics, 2018.

[5] J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[6] Y. Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692, 2019.

[7] Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. Revisiting pre-trained models for Chinese natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 657–668, Online, November 2020. Association for Computational Linguistics.

[8] Guo Zhipeng, Xiaoyuan Yi, Maosong Sun, Wenhao Li, Cheng Yang, Jiannan Liang, Huimin Chen, Yuhui Zhang, and Ruoyu Li. Jiuge: A human-machine collaborative chinese classical poetry generation system. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 25–30, 2019.

[9] Jinyi Hu and Maosong Sun. Generating major types of Chinese classical poetry in a uniformed framework. *LREC 2020 - 12th International Conference on Language Resources and Evaluation, Conference Proceedings*, pages 4658–4663, 2020.

[10] Jinyi Hu and Maosong Sun. Generating major types of chinese classical poetry in a uniformed framework. *arXiv preprint arXiv:2003.11528*, 2020.