**题目：基于周杰伦歌词的情感分析与歌词生成**

小组成员：刘志伟 董琛扬

1. **背景**

6月12日零点，周杰伦新歌《MOJITO》首发，一上线就让音乐平台瘫痪，2000年出道的他，20年后还拥有如此高的人气和关注度，这与他的歌曲曲风和词风是分不开的。而歌词是一首歌曲的灵魂所在，整段文字承载了一首歌的感情基调与主旋律。要了解一位歌手，对于其用字、用词风格的深入挖掘是至关重要的。

本项目将基于python语言，利用数据挖掘可视化方法、深度学习方法，尝试深入挖掘周杰伦歌曲的用字、用词习惯并构建歌词情感预测模型和歌词生成器。

1. **项目要求**

**Task 1** 收集相关数据，可视化周杰伦歌词用字、用词习惯；

**Task 2** 对歌词进行情感分析，构建歌词情感预测模型；

**Task 3** 构建“周氏风格”歌词自动生成器

1. **项目实现流程与结果**

(一) 数据获取与初步预处理

1. 数据来源

本项目中所用到的数据来自qq音乐官方网站，通过官网的搜索引擎可以获取任意歌手的歌曲、所属专辑名以及相关歌词等信息。

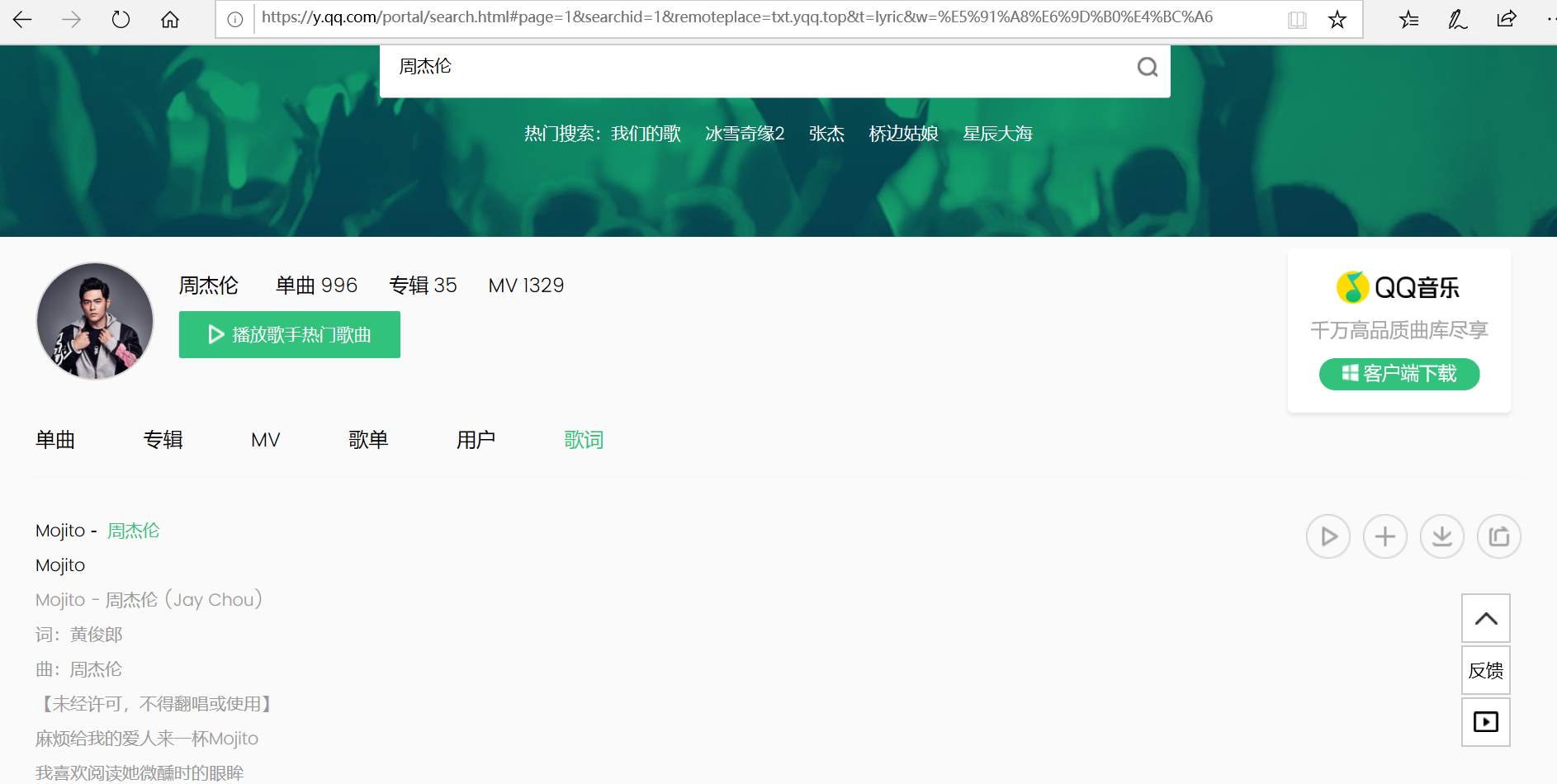


图1.QQ音乐搜索引擎

2. 数据爬取与解析

爬虫技术是一种利用程序自动访问指定浏览器的数据获取技术，而python中的requests和json库分别用于初步爬取网页源码信息和对爬取到的信息进一步解析，两者的配合能够实现爬取指定的目标数据。

基于python语言，爬取网页中的数据步骤如下：

Step 1 获取网址和用户设备、浏览器信息

网址信息在网页搜索框中即可获取；用户设备、浏览器信息可通过在目标网页下，点击F12通过开发者工具获取。



图2.获取QQ音乐官网网址



图3.获取QQ音乐搜索引擎所在网页网址和用户相关信息

Step 2通过requests库爬取网页源码

基于上一步获取到的两个网址和用户信息，通过requests.get（）函数，初步获取QQ音乐中所有周杰伦歌曲相关信息。

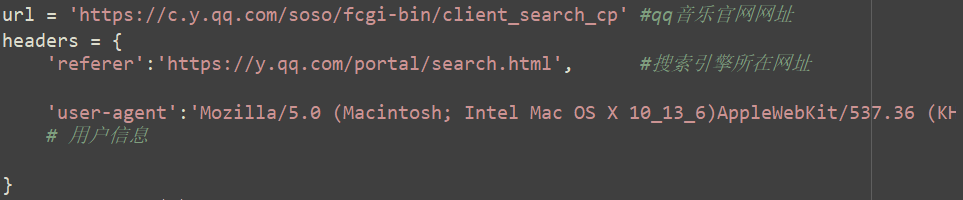




图4.相关指令



图5.部分源码信息

Step 3 通过json 解析数据、提取指标与保存数据

源码信息中有大量无用字符，通过json.loads（）函数解析源码，去除无用字符，提取歌名、所属专辑、年份、歌词等信息 ，并将结果保存为csv文件到本地

但由于需要下载qq音乐客户端并开通vip权限才能查看QQ音中周杰伦的全部歌曲的信息，由于爬虫技术有限，本项目爬取到的歌曲为当前热度排行Top 60的歌曲， 利用pandas打开数据框如下：



图5.部分数据

3. 数据标注

由于Task 2需要构建情感预测模型，而目前得到的数据是缺少情感标签的，因此需要做数据标注，即给现有的数据打上标签。

本项目根据百度百科中对于周杰伦的歌曲鉴赏，将周杰伦歌词情感分为积极（label=2）、中性（label=1）和消极（label=0）三类：



图6. 周杰伦的歌曲鉴赏

按照该分类准则，将原始数据打上标签，如下图：

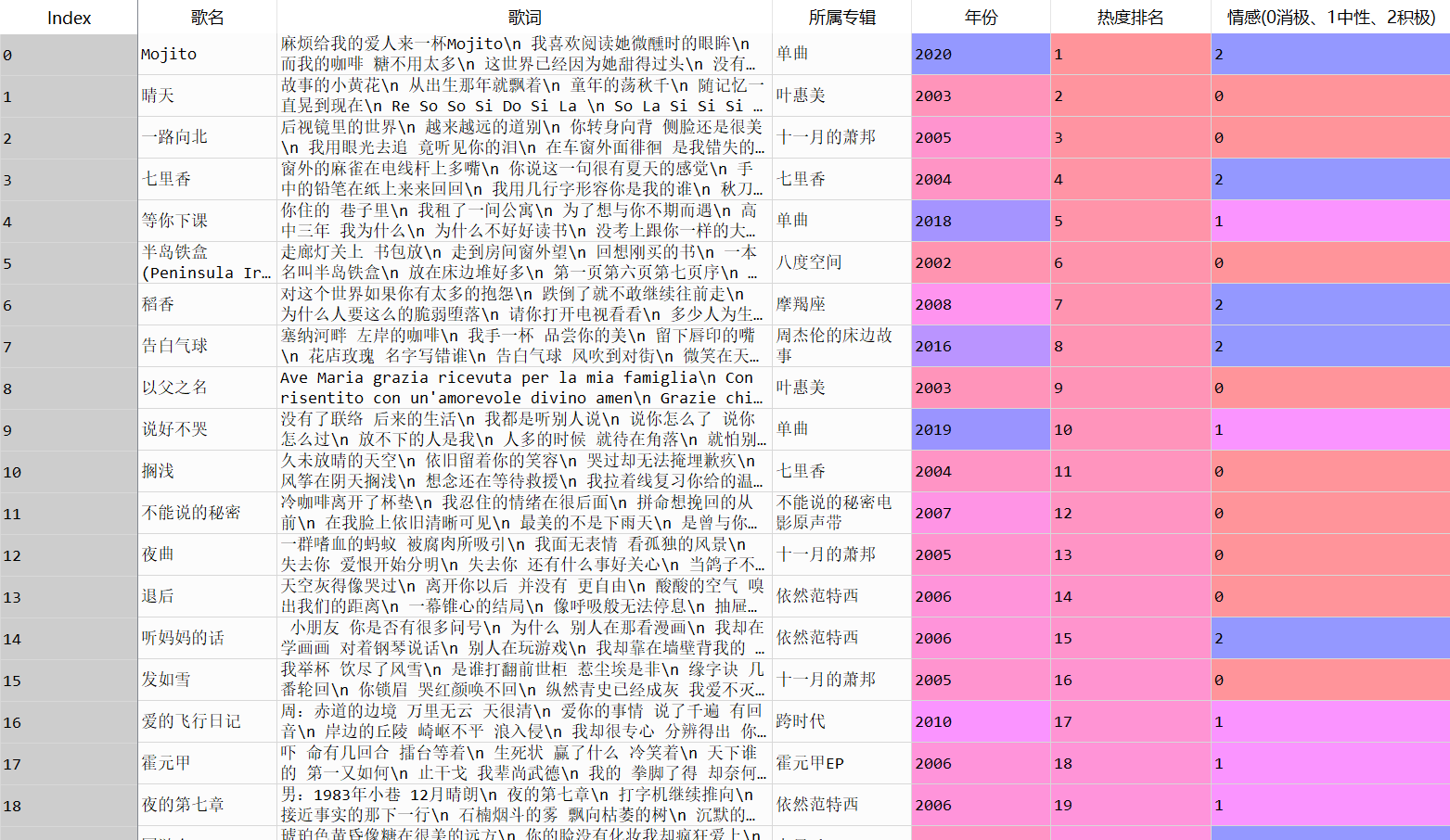


图7.打过标签后的数据

4. 歌词分词

利用Python中的jieba库对歌词进行分词操作:

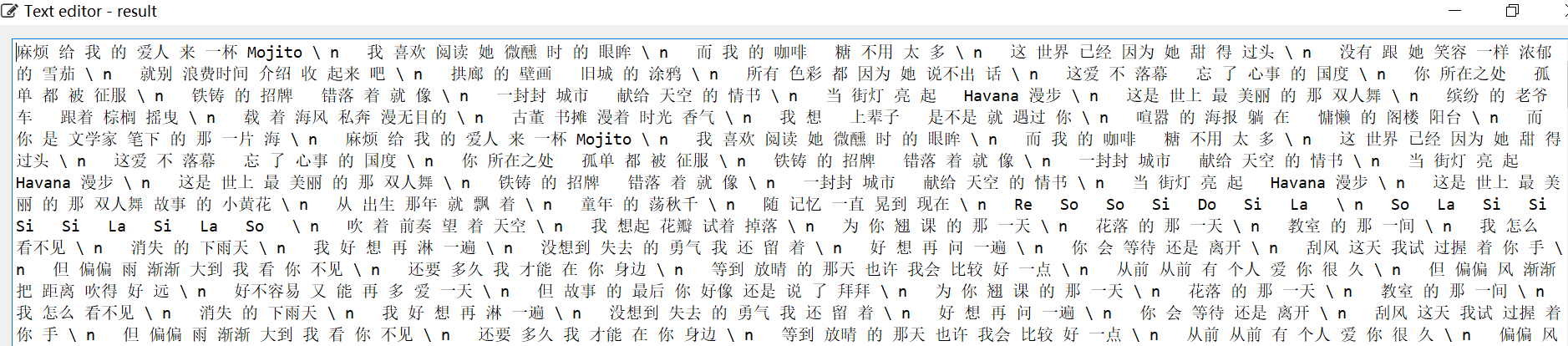


图8.部分分词结果

将分词之后的歌词与原数据框合并:

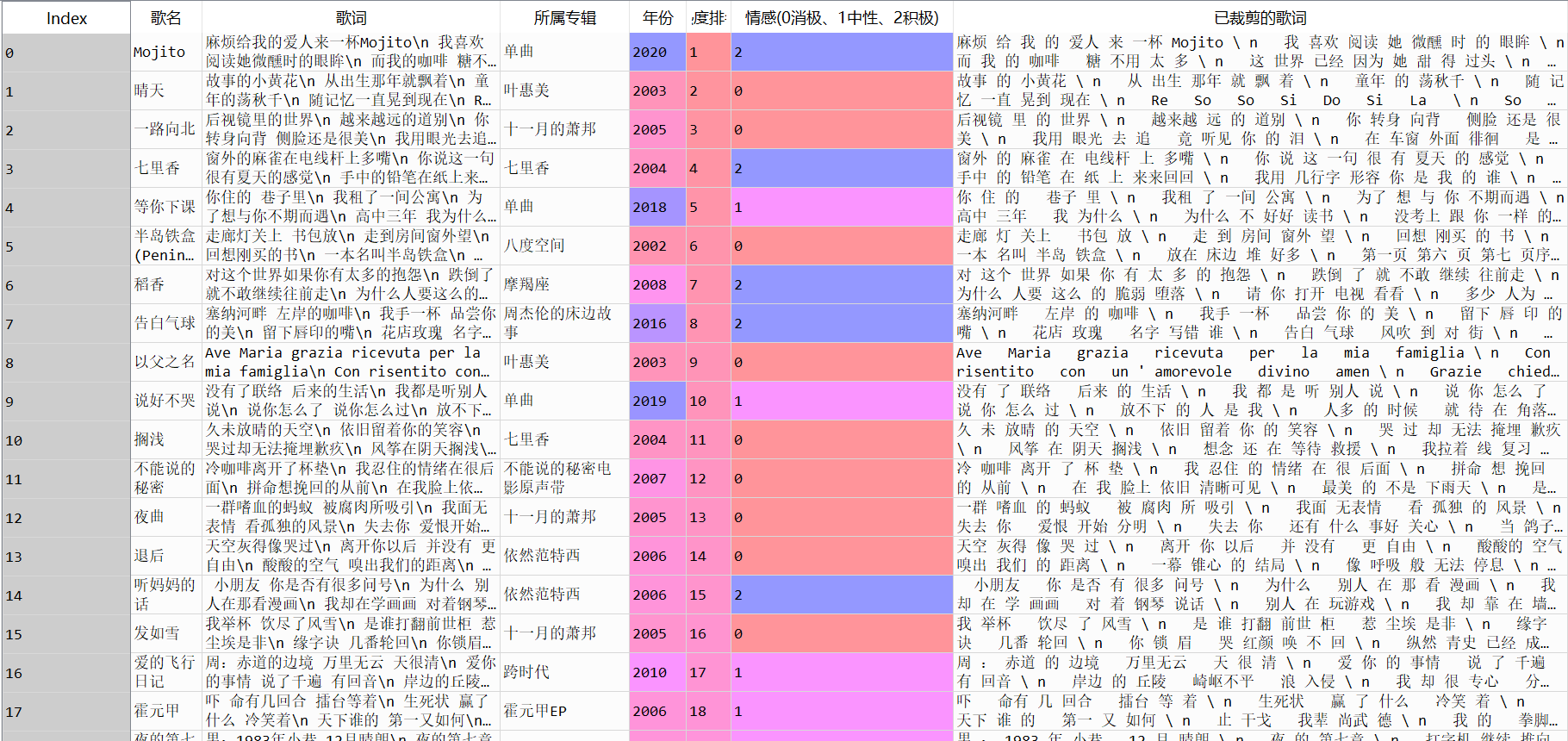


图9.合并后的数据框

(二) 描述性统计分析

对于初步预处理好的数据可以做一些简单的可视化，直观地展示周杰伦专辑热度、年份热度与用字用词习惯。

首先利用pandas进行分组聚合操作，可以分别统计出专辑频数和年份频数：

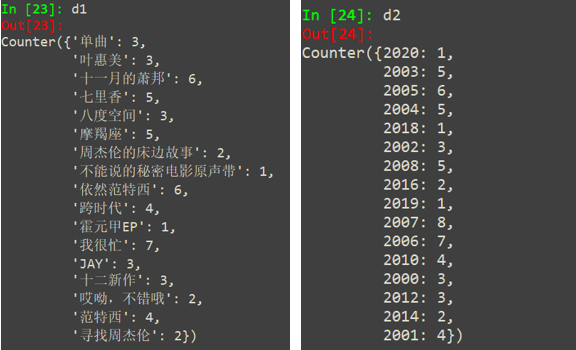


图10. 对应的专辑频数和年份频数

之后利用matplotlib分别将上述两个字典可视化：

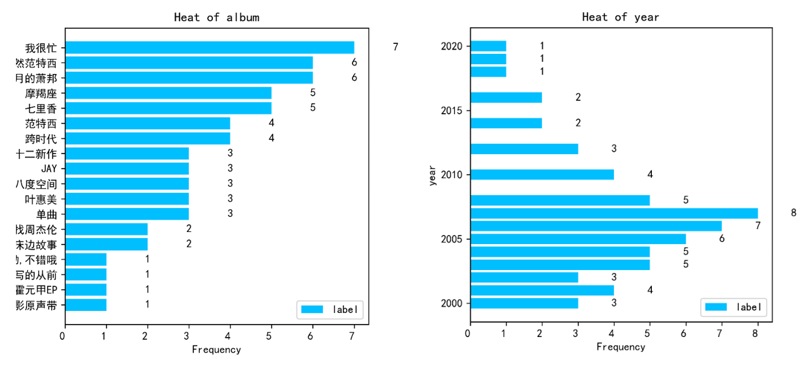


图11.可视化结果

由于本项目数据为周杰伦热度Top60歌曲，因此一张专辑出现的频数可以近似当成其热度，年份也如此。从专辑直方图中可以看出,《我很忙》、《依然范特西》、《十一月的肖邦》等老专辑目前依然很受欢迎；从年份直方图中可以明显看出周杰伦出金曲的巅峰时期在2007年前后。

进一步，进行字频统计并可视化：

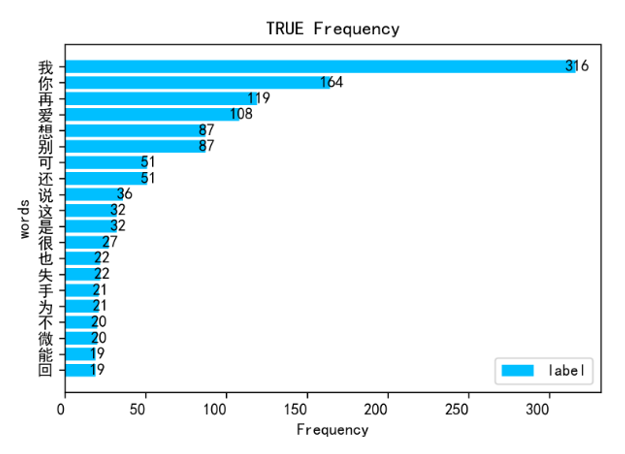


图12.字频统计（不包含英文字符）

该图中字的分布情况展示了周杰伦歌曲的用字风格，可以看出”我”、“你”、“在”、“爱”、“想”、“别”等字出现频率高。

另外，由分词结果，可以利用wordcloud进行词频统计并画出词云图：



图13.词频统计

从可视化结果很直观地可以看出，英文词汇较少，而且“我们”、“没有”、“离开”等词汇出现频率高，一股浓浓的周氏用词风格扑面而来。

(三) 歌词情感分析

1.snownlp库预测情感

从自然语言处理技术的角度来看，情感分析的任务是从文本中提取出作者所表达的情感倾向。Python中的snownlp库中的SnowNLP（）函数目前被广泛使用于预测现实生活中的各种评论的情感，此函数输入为一段文本，输出越接近1情感越积极，下图为简单测试：

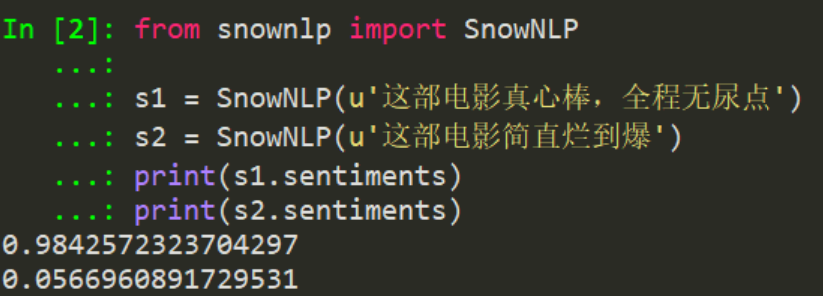


图14.snownlp预测评论

可以看出该测试结果不错，之后用SnowNLP（）预测本项目数据中前十首歌词的情感，由于数据标签是三分类，因此将预测结果小于0.4的歌词视为消极情感，大于0.6的视为积极情感，处于0.4~0.6中间的是中性情感，得到预测结果：



图15.snownlp预测歌词结果

可以看到预测歌词的效果不好，预测准确率仅为40%，且将十首歌全部预测为积极情感，这说明SnowNLP（）不能较好的预测歌词情感，原因是歌词不同于现实中的评论，歌手喜欢用某些写实的词来代表情感，比如周杰伦夜曲中的“蚂蚁”，SnowNLP（）认为它是中性词，而其实是带有消极情感色彩的，而且不同歌手的用词习惯有差异，因此用snownlp库来预测歌词情感不妥。

2.深度学习方法

正如上所提到的利用snownlp来预测歌词的弊端，本项目将基于深度学习方法，基于现有歌词数据自动学习出一个周杰伦歌词情感预测模型，即输入一首歌的歌词就能输出该段歌词的情感标签的预测值的一个分类器。

Step 1 数据准备（训练集和测试集、将词one-hot编码）

为了测试模型的泛化能力，首先将数据分为训练集（50首）和测试集（10首）；

为了让歌词能够作为模型的输入，利用CountVectorizer模块使用one-hot编码原则将歌词中已经分好的词进行编码，使得每一个词汇对应高维空间中的一个单位基向量，空间的维度即为分词的词种类数；

根据one-hot编码后得到的词向量，可以表示出每一首歌词所对应的向量。比如，“一个”对应的one-hot编码向量为（1,0,0,0,…,0），第十首歌歌词中出现“一个”的次数是两次，因此对应的歌词向量的第一个维度“一个”处的值为2，基于此方法，每一首歌词都可以向量化：

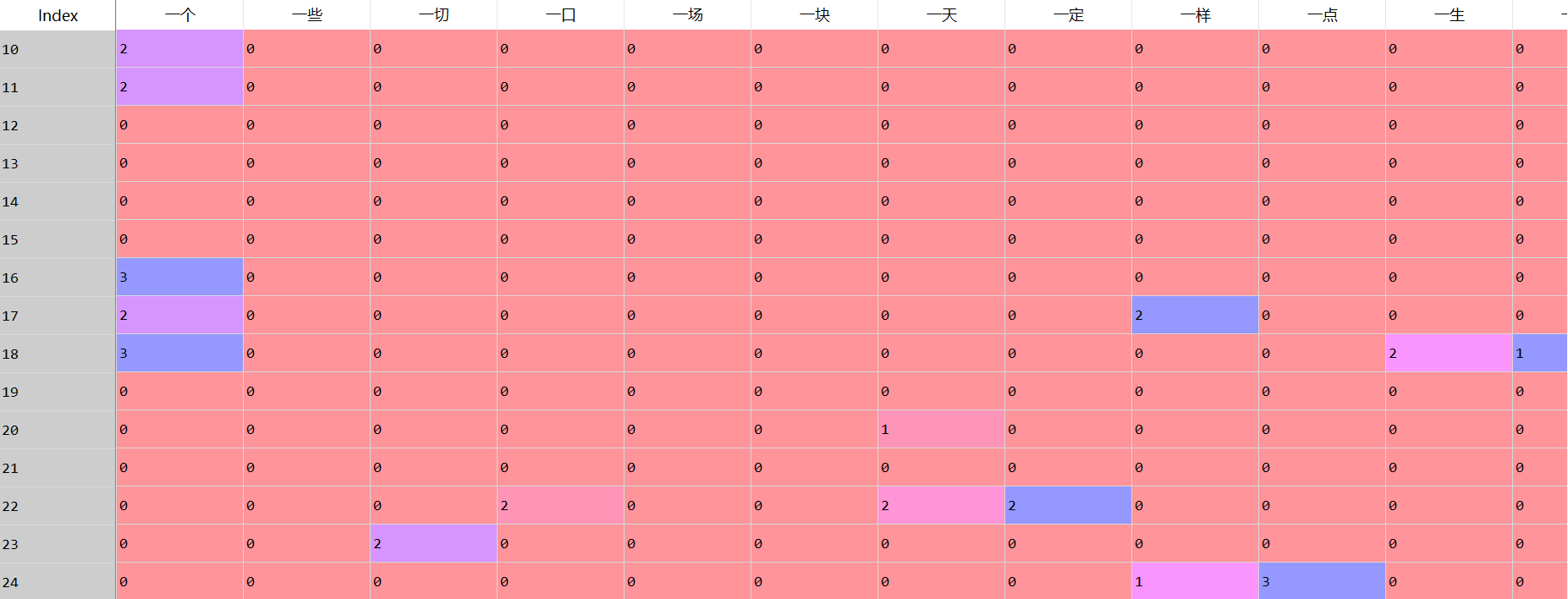


图16. 部分歌词向量化后的结果

Step 2 DNN模型与结果展示

一个标准的前馈神经网络往往能够解决很多实际问题，本项目首先基于tensorflow框架尝试构造一个简单的单隐层DNN模型，网络框架如下图所示：

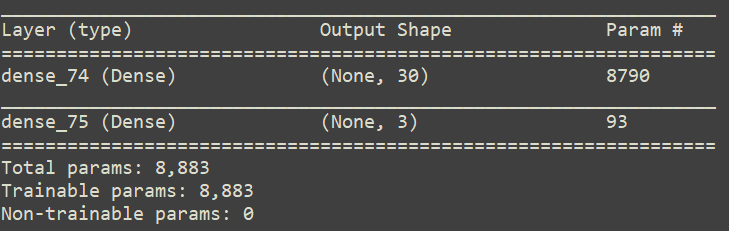


图17. 单隐层DNN（输入节点与歌词向量维度一致；隐层节点数为30，激活函数设置为”relu”；输出层节点数为3，激活函数为”softmax”）

由于是分类问题，选择loss为交叉熵损失，优化算法设置为adam算法，设置验证集为测试集的10%，由于本项目训练集基数小，因此设置迭代周期epoch（遍历完整个训练集的次数）为20次：

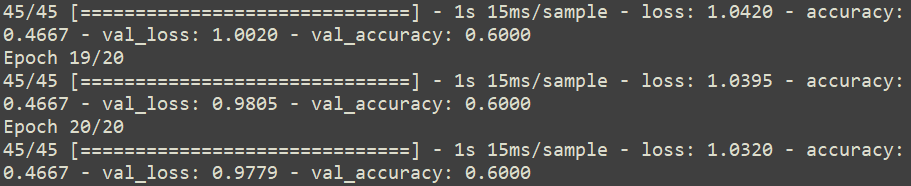


图18. 训练过程展示

可视化学习曲线如下图：

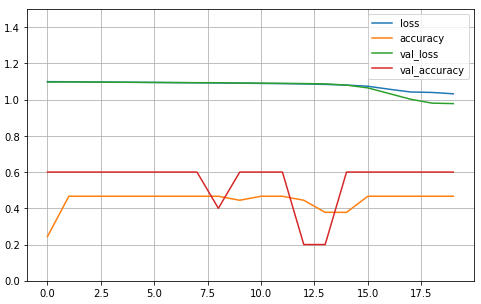


图19. 单隐层DNN学习曲线

从结果上看，模型在训练集上的准确率只有46.7%，而学习曲线在17步之后已经趋于平缓了，模型存在欠拟合风险。

基于上述问题，应当使用更复杂的模型拟合数据，因此尝试增加两个隐层，得到三个隐藏层的DNN：

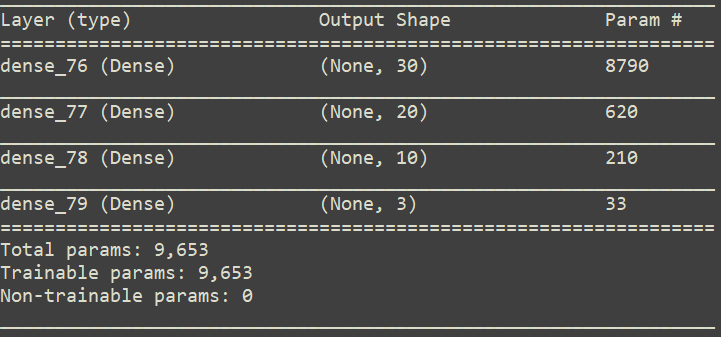


图20. 三隐层DNN（输入节点与歌词向量维度一致；隐层节点数分别为30、20、10，激活函数设置为”relu”；输出层节点数为3，激活函数为”softmax”）

按照同样的loss、算法，迭代周期epoch设置为50次，可视化学习曲线如下图：

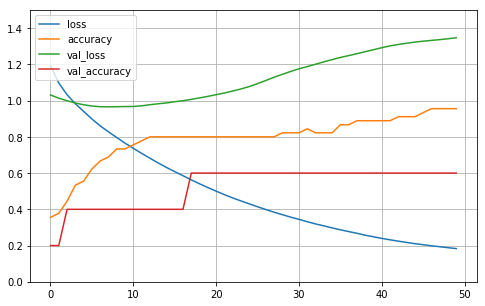


图20. 三隐层DNN学习曲线

相比于单隐层网络，添加了两个隐层之后模型在训练集和验证集上的准确率都得到了提升，但从图中可以看出绿线所代表的验证集的loss在20步之后逐渐上升，模型存在过拟合风险，进一步查看其在测试集上的表现：

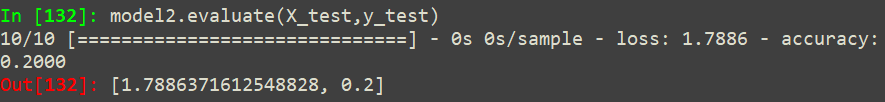


图21. 三隐层DNN测试集结果

在测试集上模型精度仅为20% ，因此通过算法提前终止试图进一步改善，将epoch下调为20，重新训练并可视化结果：

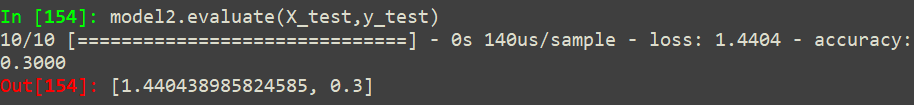


图22. 算法提前终止测试集结果

模型在测试集上结果仍不乐观。

事实上，DNN模型结构上较为简单单一，用其来做歌词情感预测，需要将每一首歌词”向量化”，而一首歌词在向量化之后，歌词的顺序已然被打乱，前后词之前的联系对情感造成的影响对于DNN模型来说是没法学习到的，比如“欲扬先抑”类的歌词用此模型预测效果可能不佳，因此需要一个更加适合语言的序列模型来解决问题。

Step 3 基于词嵌入的LSTM模型与结果展示

LSTM又称长短期记忆网络，作为一种特殊的RNN，适合于处理和预测时间序列中间隔和延迟相对较长的重要事件，其基本结构如下图：

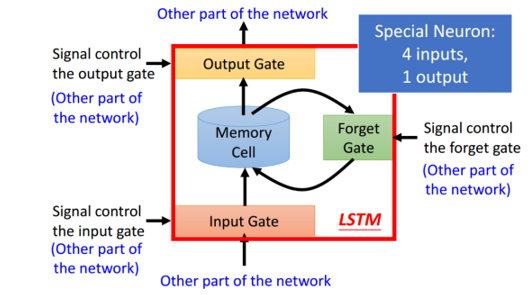


图23. LSTM基本框架图示

而词嵌入技术，在自然语言处理领域一直被广泛运用，概念上而言，它是指把一个维数为所有词的数量的高维空间嵌入到一个低维数的连续向量空间中，每个单词或词组被映射为实数域上的向量。本质上就是把基于one-hot编码后的高维词向量嵌入到低维空间并成为低维空间中的向量，此技术可以让词与词之间被赋予相关性。

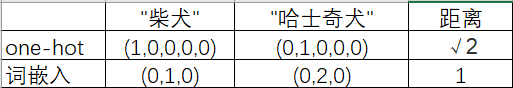


图24. one-hot与词嵌入举例

由于深度学习在NLP领域中的兴起，将词嵌入技术运用在LSTM模型中是很自然的，将词嵌入层作为输入层和LSTM层的中间层，可以使得网络自动学习词嵌入映射方式：

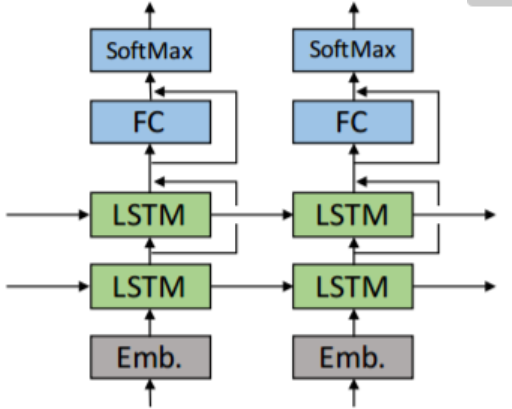


图25. 基于词嵌入的LSTM

同样利用深度学习框架Tensorflow，构建基于词嵌入的LSTM:

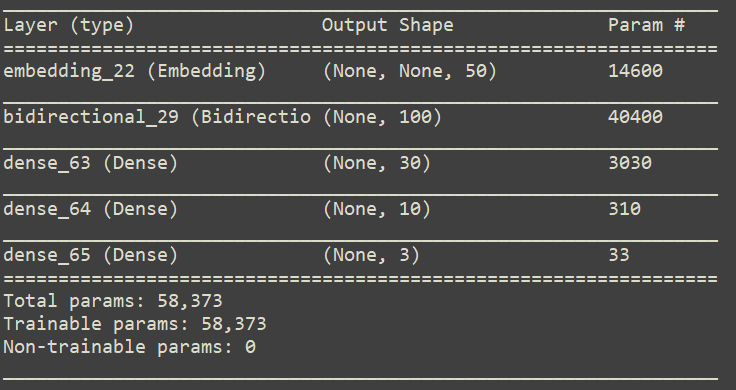


图26. 基于词嵌入的LSTM（输入节点与one-hot编码后的词向量维度一致；隐层分别为嵌入维度为50的词嵌入层、LSTM层与两个密度层，激活函数设置为”relu”；输出层节点数为3，激活函数为”softmax”）

选择loss为交叉熵损失，优化算法设置为adam算法，设置验证集为测试集的10%且迭代周期epoch为20次：

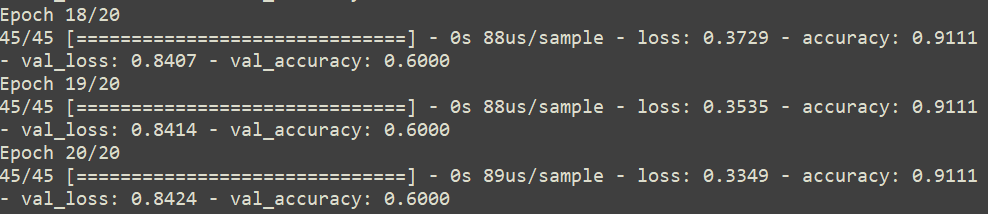


图27. 训练过程展示

接着可视化学习曲线：

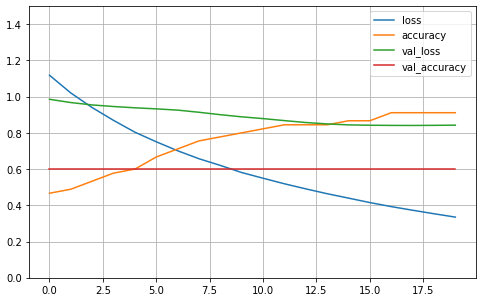


图28. 训练过程展示

结果表明训练好的模型在训练集和验证集的表现不错，loss的下降也趋于平缓，查看该模型在测试集上的表现：



图29. 测试集表现

在测试集上的准确率达到了70%，相比于DNN模型准确率提高了不少。具体预测结果如下图：

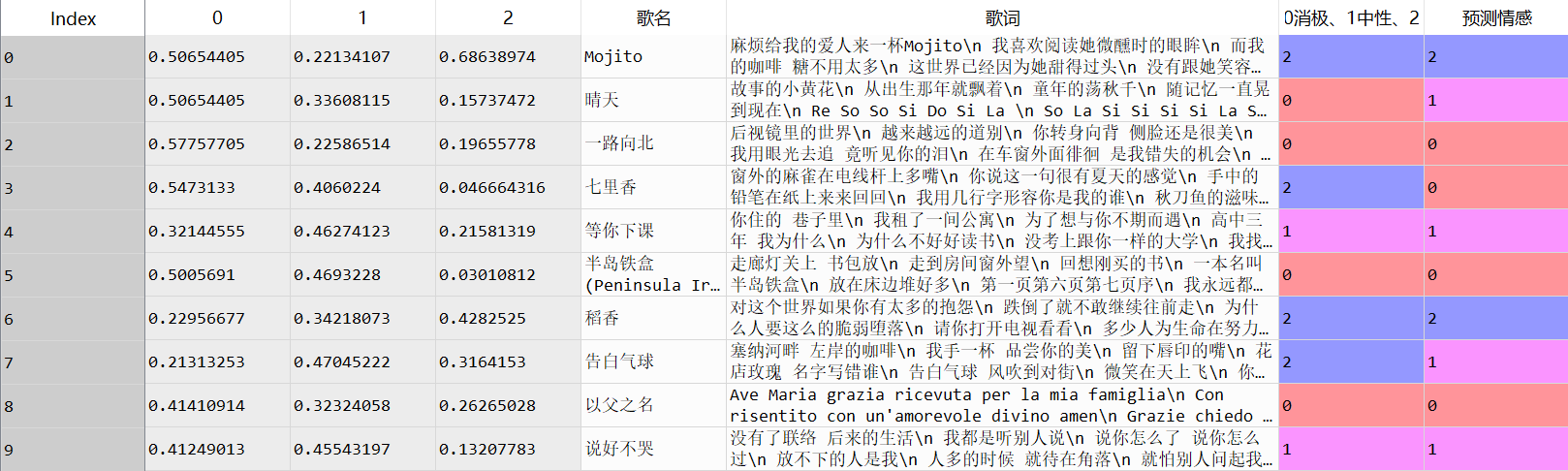


图30. 具体测试结果

(四) 歌词生成器的构建

在上一节用LSTM构建了周杰伦歌词情感分类器，且性能相比于DNN更好，这意味着LSTM似乎是很合适的语言模型。为了完成Task3，本小节我们将继续用LSTM构建具有周式风格的歌词生成器。

Step 1 字符拆分

首先将每首歌词拆成字符，共1786种字符（包括汉字、字母、空格、换行符）：

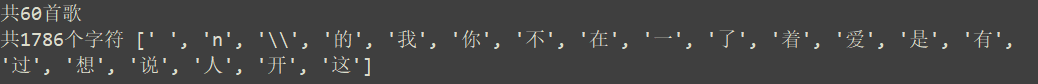


图31. 拆分结果

接着对于字符进行one-hot编码，使每个字符对应一个编号，进而对应1786维空间中的一个单位向量；

Step 2 构建LSTM

在本节中，效仿上一节，同样利用tensorflow框架构建基于嵌入的LSTM模型：

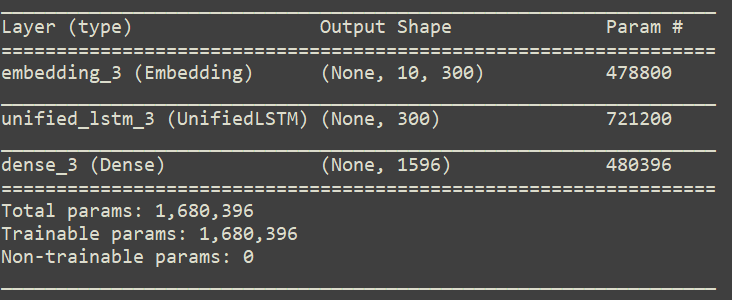


图32. 模型框架

该网络输入层节点数为10，嵌入维度为300，输出节点数为1596（激活函数为”softmax”），即每次按顺序依次丢入十个字符对应的字符向量，该模型将输出1596维的一个高维向量，该向量的分量为该模型预测的此分量对应的位置所代表的字符的出现概率，且规定预测结果为概率最大的值对应的字符。

Step 3 训练与结果

损失设置为交叉熵损失，优化算法为adam，迭代周期epoch=20，在全数据集上进行训练（验证集设置为数据集的10%）：

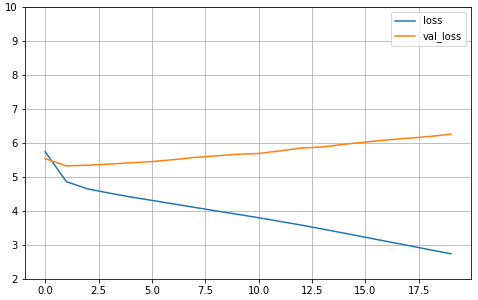


图33. 学习曲线

模型的验证误差从第一步之后就开始逐渐爬升，模型存在过拟合风险，因此将嵌入维度下调至100：

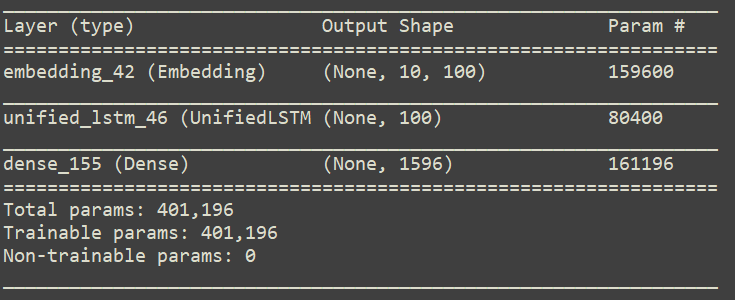


图34. 简化模型

同样训练20个epoch：

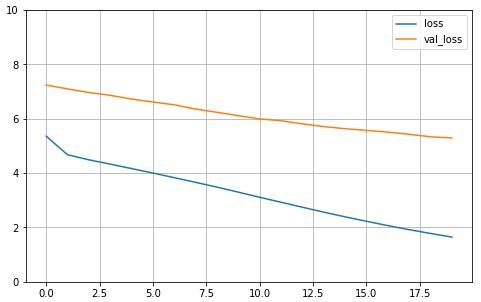


图33. 学习曲线

结果表明两条loss都下降到了比较低的水平，用该模型随机生成一段歌词：

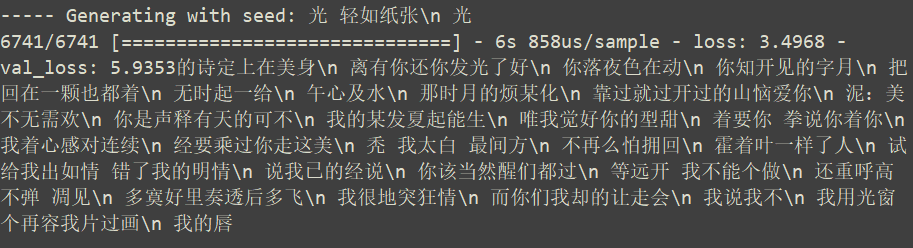


图34. 随机生成的一段歌词

从生成结果中可以看出部分语句不是很通顺，且出现不成词的问题，但是对于断句的情况、每一句话的主谓宾结构，该模型学习地还不错，且“周氏风格”似乎被生成出来了。

进一步，利用已经训练好的生成器，生成一大段话，字符数为2000，利用jieba分词后画出词云图，与真实词云图进行比较：

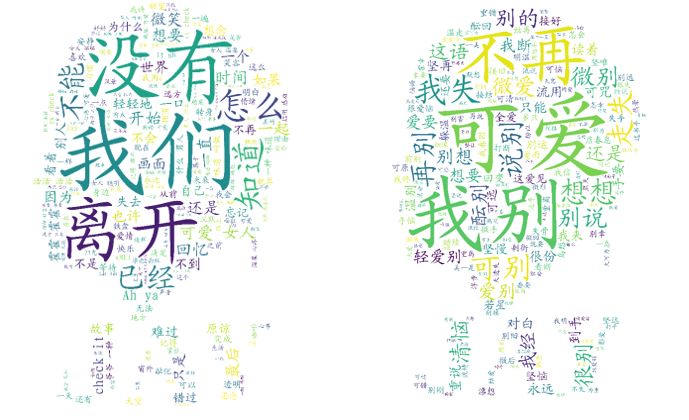


图35. 用词分布比较(左真实、右生成)

上图表明，该生成器生成词的效果并不是特别理想，出现了很多“假词“。再将结果中的汉字进行字频统计且可视化字的频数直方图，与真实情况进行对比：

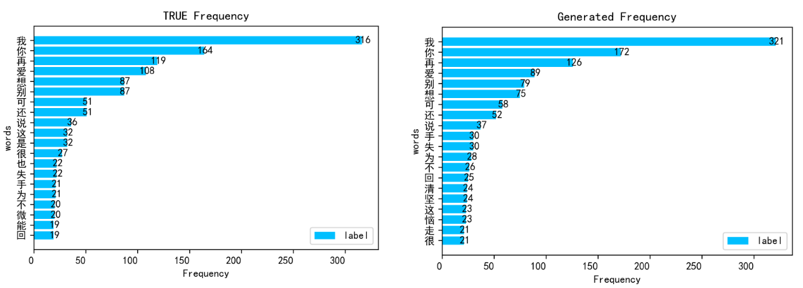


图35. 用字分布比较(左真实、右生成)

通过上图可以直观看出，生成器生成词的分布情况与真实情况大致相近，”我”、“你”、“再”、“爱”等出现次数靠前的汉字按频数顺序两幅图几乎是一致的，其余汉字的频数排序略微发生变动，这说明训练好的生成器在一定程度上学习到了周杰伦歌词的字的分布规律。

1. **总结与不足**

本项目爬取了周杰伦的部分歌词数据，并可视化了数据，展示了周杰伦歌词的用字、用词风格与习惯；之后对歌词进行情感分析，分别利用DNN和LSTM构建了歌词情感分类器，并比较了两者的性能；最后利用LSTM构建了歌词自动生成器，且训练好的生成器在一定程度上学习到了周杰伦歌词的用字习惯。

本次项目主要的不足点和待完善的地方主要有三处：

1. 数据量不足，应扩充样本量，大数据更适合深度学习模型；
2. 数据标注虽然参考百度歌曲鉴赏内容，但仍具有主观性；
3. 在本报告的第三部分，只是建立了基于字符嵌入的LSTM，因此无法避免预测的歌词出现不成词的情况，因此可以进一步考虑建立类似于第二部分的基于词嵌入的LSTM，在结果上可能有所改善，两个模型可以对比比较性能差异。

**五、组员分工**

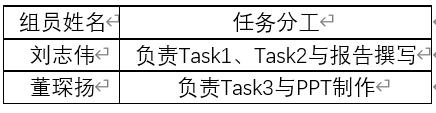


图36. 组员分工

**六、代码展示**