项目实训总结报告

项 目 名 称: 基于机器学习的旅游数据分析与推荐平台

专 业 方 向： 软件工程——数据科学及可视化

项 目 负责人： 米思成

专 业 班 级: 6班

小 组 名 称： 第9组

组 员 姓 名： 杨文涛、张和钰、平云、张瑞成

实训起止日期： 2021.5.6 – 2021.6.26

# **1. 引言**

旅游业是一个综合性很强的信息依赖产业，旅游信息的获取，加工，传播和利用对旅游业的发展起着举足轻重的作用，从旅游者和服务提供方面的需求出发建立旅游信息数据库并进行可视化展示，不仅可以使旅游者能够直观的参考利用已经得出的数据统计结果，还可以进行快速，准确的查找和检索自己需要的旅游信息，促进旅游信息的共享，打破旅游信息的封锁。

# **2. 需求分析**

在爬取携程网站上的游记、旅游线路、景点数据后，本小组计划在本次实训项目中实现如下数据分析和功能：

2.1 基础功能

2.1.1 不同城市的平均评分

在爬取下来的数据里面，有如下一项数值值得分析：一座城市的平均评分，对于游客来说，选择出游的很重要一点就是基于以往旅游者的先验经验。本小组对数据进行处理，得到一座城市的平均评分，因为选取的数据量比较大，一定程度上可以去除个体的差异性，得到一个比较公正可信的结果。

将每座城市的平均评分按照由高到低的顺序进行排序，在前端以柱状图的形式进行展示，横坐标为每个城市对应的分数，纵坐标为城市名称，此种受欢迎度信息可以在一定程度上给使用者在选择出游的时候提供参考

2.1.2 不同城市的行程安排，导游服务，酒店体验评分分析

由于顾客的个体差异性，每个顾客在选择出游时，都有自己的偏好性和口味，对于不同方面的要求不尽相同。所以本小组根据爬取的携程数据，得到每个城市所有的旅行线路中关于评分部分的不同方面，有行程安排，导游服务，酒店体验这三项。这样，就将评分进行了细化，游客在选择出游时参考的因素也会更加全面。

在前端对于每个城市的行程安排，导游服务，酒店体验这三项的平均成绩，进行可视化展示，展现形式为柱状图，横坐标为城市名称，纵坐标为上述三项的平均成绩。

2.1.3 不同城市旅游价位分析

对于游客来说，选择城市旅游的因素除了城市受欢迎指数，服务好坏之外，性价比也是值得考虑的一点。本小组得到每个城市的旅游线路消费平均水平，对其进行高到低的排序，在前端进行展示，展示形式为柱状图，横坐标为城市名称，纵坐标为城市评分。同样地，该分析会为游客出行提供一定的参考

2.1.4 旅游线路点评关键词词云

对于不同的游客对于旅行侧重的方面不同，通过旅游路线中的数据，可以分析出游客旅游更注重什么因素，例如行程，导游，性价比，风景等因素。根据爬取的旅游线路，提取数据中的关键词这一项，统计所有关键词的出现次数，选取出现次数大于一百次的关键词，在前端以词云的形式展示出来。

2.1.5 旅行线路统计分析

根据爬取的旅行线路，统计所有从北京出发到各个热点城市旅游线路数量，进行可视化展示。在展示形式上，采取了绘制地图的方式，地图的展示方式较为直观，观者可以较为清晰地感受到热门城市的分布。除此之外，本小组绘制了柱状图，展示从北京出发到各个城市线路的数量。

2.1.6 出游伙伴的分析

旅游业主要是凭借旅游资源和设施，专门或者主要从事招徕、接待游客、为其提供交通、游览、住宿、餐饮、购物、文娱等环节的综合性行业。而游客选择的出游伙伴不同，势必面临着在服务提供方面的差异性。因此对于商家或者服务方来说，分析游客出游伙伴的类型是很有必要的。根据出游伙伴地不同，商家或者服务方可以更加有针对性地提供服务，提高该城市的竞争力。

本小组将爬取的不同的旅游路线作为整体分析，对出游伙伴类型计次，在前端以饼图的形式展示。

2.2 拓展功能

2.2.1 基于游记的旅游线路推荐功能

以游记文本作为主体，通过相关方式获取相应景点的热门程度，并基于此生成一条由热门景点构成的推荐的旅游线路，帮助用户在有限的时间内规划更能展现城市魅力、更有意义的行程。

通过爬虫获取游记文本，对文本进行预处理，包括设定自定义词表、分词和去停用词等。然后对游记文本进行词频统计，并基于景点的热门程度，设计相应的代表度计算方式，得到热门景点的代表度排行。并基于此，给出代表度排行前五的景点作为该城市的推荐线路。

2.2.2 基于LSTM的景点名赋诗功能

经历一天充实的旅行后，许多人都迫不及待地想将自己的所见所想分享给身边的人。但你一定遇到过这样的情况：别人的朋友圈文案都是“落霞与孤鹜齐飞，秋水共长天一色”，而自己却感受到“只能意会，无法言传”的词穷。

本小组的旅游文案生成功能就是瞄准了这一痛点，使用唐诗（五言、七言）作为数据集，构建长短期记忆神经网络（LSTM）进行训练，生成模型。用户可以通过输入正在旅行的景点名称来得到数句自动生成的“古诗”文案，为词穷的燃眉之急提供灵感，为旅行途中增添乐趣。

2.2.3 基于余弦相似度的旅游线路聚类分析

现如今，各大旅游网站都有海量的旅游线路供用户选择，但过多的线路也对用户造成了困扰：信息量过大，常常混淆不同线路，无从下手选择，为旅行增添了不必要的麻烦。本小组通过对大量数据的观察、比对，发现导致这一通点的主要原因是存在大量的相似线路，它们分布零散，不易记忆。

本小组通过对旅游线路的行程信息分词处理，再与爬取的景点名称匹配，得到具体旅游线路的景点，将其聚类分析，将相似的旅游线路归为一类，帮助用户解决困扰。

2.2.4 基于LDA的主题游记功能

旅游网站的游记数量纷繁冗杂，用户想要从中找到值得自己参考的游记犹如大海捞针。因此，我们希望能够通过相关的工作，以城市作为单位，对其游记进行聚类、筛选，得到该城市具有参考意义的游记，供用户参阅。

通过爬虫获取游记文本，对文本进行预处理，包括设定自定义词表、分词和去停用词等，并从中获取相关类型词作为数据集。然后将数据集划分，不断训练模型，得到最佳聚类效果的的主题数目，并以此作为基准训练LDA模型，获得每个城市的相关主题。最后基于获得的主题关键词，使用tf-idf算法匹配相关的游记，得到该主题的下的代表性游记。

## **3. 运行环境**

3.1 操作系统要求

Windows 10 和 Mac系统下均可运行

3.2 浏览器要求

最新版Chrome浏览器

# **4. 概要设计**

4.1 基础功能

4.1.1 不同城市的平均评分

该图表主要展示不同城市的平均评分

4.1.2 不同城市的行程安排，导游服务，酒店体验评分分析

该图表主要展示不同城市的行程安排，导游服务，酒店体验评分

4.1.3 不同城市旅游价位分析

该图表主要展示不同城市的旅游价位分析

4.1.4 旅游线路点评关键词词云

该图表主要展示了旅游线路点评关键词

4.1.5 旅行线路统计分析

该图表主要展示了各个旅游线路

4.1.6 出游伙伴的分析

该图表主要展示了出游伙伴所占比例

4.2 拓展功能

4.2.1 基于游记的旅游线路推荐

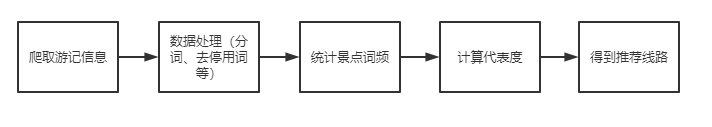
4.2.1.1 设计思路

对大量的游记，进行数据处理。保留其中有利于分析的词，形成数据集。以城市为单位，提取游记中所记录的景点名。然后基于景点设计一个合理的权重机制，来表示每个景点的代表度，并基于代表度给出推荐线路。

4.2.1.2 模块功能介绍

基于游记数据给出城市旅游推荐线路，帮助用户在有限的时间内规划更能展现城市魅力、更有意义的行程。

4.2.1.3 程序流程图



4.2.2 基于LDA的代表性游记提取

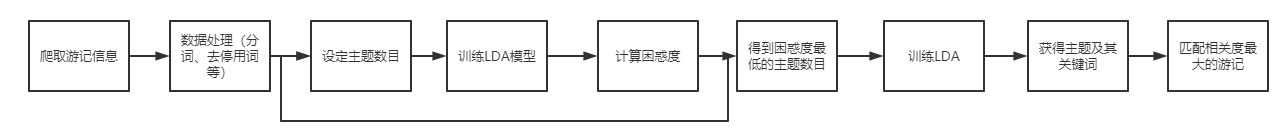
4.2.2.1 设计思路

对大量的游记，进行数据处理。保留其中有利于分析的词，形成数据集。以城市为单位，通过LDA模型获得该城市游记的主题。并使用主题的前十个关键词去匹配相关度最高的游记，作为代表性游记。

4.2.2.2 模块功能介绍

筛选出当前城市游记中最具代表性的游记，帮助用户定位有价值的游记信息。

4.2.2.3 程序流程图



4.2.3 基于LSTM的景点名赋诗功能

4.2.3.1 设计思路

对大量的诗句数据集进行数据处理，筛去宋词的部分，只保留五言、七言的诗句，形成数据集。使用tensorflow构建LSTM神经网络，训练生成模型文件。

4.2.3.2 模块功能介绍

通过用户输入的景点名，使用训练好的模型生成藏头诗。

4.2.4 基于余弦相似度的旅游线路聚类分析

4.2.4.1 设计思路

对旅游线路中的行程数据进行结巴分词，再将结果与爬取的景点名进行匹配，得到list1。将每条线路的list1按照景点名列表进行映射，得到一个向量。重新编写kmeans中的距离函数为余弦相似度（原为欧氏距离）。使用新的cos\_kmeans聚类，得到聚类中心向量，反向映射为景点名，再与list1进行匹配，匹配度高的为一类。

4.2.4.2 模块功能介绍

通过用户输入的城市名，聚出不同的类别，将聚类中的具体线路展示给用户。

# **5. 详细设计**

5.1 基础功能

5.1.1 不同城市的平均评分

在爬取下的旅游路线信息里有描述城市优劣程度的指标，如：钻数，总评分，好评率，为了更加综合地表示这一个城市的受欢迎指数，不能仅仅采取单独某一项数值来表示。一方面是为了更加充分有效全面地利用爬取下来的数据，另一方面是为了使城市优劣程度的指标更加有说服力，经过数据对比，决定采取钻数和总评分两项做加权计算来作为数据来源。

在前端的表现方式上，采用横向柱状图的方式展示数据。我们展示的数据维度分类不多，而且维度字段名称又不会太长时，所以很适合用柱形图来表示数据。将每座城市的平均评分按照由高到低的顺序进行排序，在前端以柱状图的形式进行展示，横坐标为每个城市对应的分数，纵坐标为城市名称。考虑到页面的布局，采用每两秒切换展示的方法。

5.1.2 不同城市的行程安排，导游服务，酒店体验评分分析

在爬取下的旅游路线信息里有描述该路线各方面服务的指标，如：行程安排，导游服务，酒店体验评分等。但是每段路线的各方面评分是不一定的，在仔细筛查对比数据后，决定使用行程安排，导游服务，酒店体验评分这三项。这三项是大部分路线都具有的分方面指标，决定将这三项作为数据来源。

在前端的表现方式上，采取了柱状图的形式，横坐标为城市名称，纵坐标为上述三项的平均成绩，可以直观地看出每个城市每个方面地评分高低。

5.1.3 不同城市旅游价位分析

在爬取下的旅游路线信息里有描述每段路线花销的指标，是价格这个因素。在外出旅游中，价格分析是可取的，但是以路线为单位，对价格做分析没有太大价值。于是把同一个城市的旅游路线看做一个整体，计算它们的平均价格，这样对于游客的出行时比较有价值的。所以决定把每个城市的平均旅游线路价格作为数据来源。

在前端的表现方式上，采取了条形图的形式，条形图具有直观、形象、生动、具体等特点。该类统计图可以使复杂的统计数字简单化、通俗化、形象化,使人一目了然,便于理解和比较。可以直观展示统计信息属性,是一种对知识挖掘和信息直观生动感受起关键作用的图形结构。为了直观地反映价钱，我们选择了该种图形。横坐标为城市名称，纵坐标为城市评分。

5.1.4 旅游线路点评关键词词云

在爬取下的旅游路线信息里有描述每段路线游客对于其直接感受的评价，比如说，"导游服务好"，"行程棒"，"风景美"，"导游讲解详细"这样的关键词。外出旅游是一件具有主观性的事件，所以对该关键词的分析也具有价值。将所有城市所有路线出现的关键词计次，并去掉出现出现次数100次以下的关键词，将其视为“不重要“的信息而不予以展示。对于在所有旅游路线中出现100次极其以上的关键词，把这个当作数据来源。

在前端的表现方式上，采取了词云的表现形式，词云在信息图表中是吸引眼球的元素，相比其他如条形图更能吸引注意力，它提供了某种程度的‘第一印象’，最常使用的词会一目了然，并且它可以从不同维度展现数据：词汇本身、频率、以及词汇的颜色。

5.1.5 旅行线路统计分析

对于爬取下来的信息，将每座城市的路线视为一个整体，统计每座城市的路线的数据，把这个当作数据来源，并且选取出游频次最高的几座城市。

在前端的表现方式上，采取了地图的表现形式。地图在展示路线上具有得天独厚的优势， 用户可以直观地看出游客去往哪些城市游玩。

5.1.6 出游伙伴的分析

在爬取下的旅游路线信息里有描述出行伙伴的信息，这是一个比较重要的信息。对出游伙伴的分析不应该区分城市，而是要把所有旅游路线作为一个整体分析。将每一条旅游路线里的出游伙伴类型做一个计次，统计所有出游伙伴类型的次数，将其作为数据来源。

在前端的表现方式上，采取了扇形图的表现形式。扇形统计图能够准确的反应出各组数据所占的百分比,各个部分数量与总数之间的关系，可以清楚直观地表示出各部分数量与总数之间的关系，不同的出游伙伴类型可以被直观表现出来。

5.2 拓展功能

5.2.1 基于游记的旅游线路推荐

1. 原理

旅游线路的最终形式为多个景点的组合，因此，其中最重要的是景点的信息处理。

而我们能够获取到的景点的信息由两部分组成：

1. 爬虫获取的景点信息
2. 从游记中提取得到的

爬虫能得到的信息有很多，能够帮助建立相关度的信息项也很多，但考虑到要与游记中提取到的信息相关联，因此在这里我们选择了景点的热门榜单排行作为我们判定景点代表度的依据之一。

为了强调景点的重要程度，我们选择从游记中提取景点名词出现的数量作为景点的另一个代表度的依据。

而代表度的设计，本身是为了规避一些普遍的休闲娱乐场所，诸如人民公园、超市、KTV等。然而在。因此在设计权重的计算方式时，需要加大景点在热门榜单中排名的优势，同时也要削弱（不是消除）同层级景点中排名对代表度的影响。

综上所述，经过多次尝试，最终设计的代表度计算方式如下：

|  |  |  |
| --- | --- | --- |
| **排名** | **初始权重** | **衰减** |
| 1-20 | 1 | 0.001 |
| 21-50 | 0.98 | 0.005 |
| 51-100 | 0.83 | 0.01 |
| 101-500 | 0.33 | \ |
| >500 | 0.1 | \ |

代表度 = 权重 \* 频次

1. 算法

if rank in range(1, 21):  
 rank\_score = 1-0.001\*rank  
elif rank in range(21, 51):  
 rank\_score = 0.98-0.005\*(rank-20)  
elif rank in range(51, 101):  
 rank\_score = 0.83-0.01\*(rank-50)  
elif rank in range(101, 501):  
 rank\_score = 0.33  
else:  
 rank\_score = 0.1  
return count \* rank\_score

1. 流程

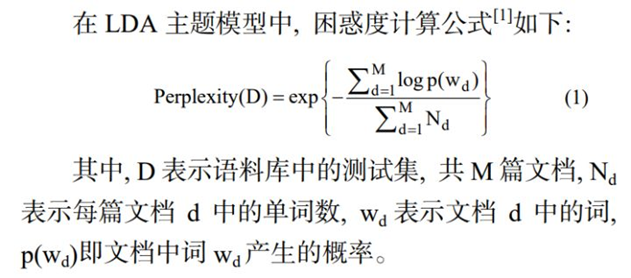
(1) 基于景点在热门榜单中的排名计算其权重值；

(2) 将权重值与景点在游记中的频次相乘，得到景点的代表度

5.2.2 基于LDA的代表性游记提取

1. 原理
2. 游记主题数目确定

正如分类问题用Accuracy来评价，LDA也自己评价标准叫Perplexity。Perplexity可以粗略的理解为“对于一篇文章，我们的LDA模型有多不确定它是属于某个topic的”。topic越多，Perplexity越小，但是越容易overfitting。



我们利用Model Selection找到Perplexity又好，topic个数又少的topic数量。可以画出Perplexity vs num of topics曲线，找到满足要求的点。

1. LDA模型

关于LDA有两种含义，一种是线性判别分析（Linear Discriminant Analysis），一种是概率主题模型：隐含狄利克雷分布（Latent Dirichlet Allocation，简称LDA），这里用到的是后者。

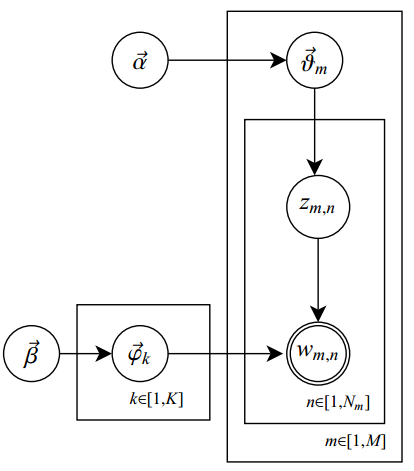
LDA由Blei, David M.、Ng, Andrew Y.、Jordan于2003年提出，是一种主题模型，它可以将文档集中每篇文档的主题以概率分布的形式给出，从而通过分析一些文档抽取出它们的主题（分布）出来后，便可以根据主题（分布）进行主题聚类或文本分类。同时，它是一种典型的词袋模型，即一篇文档是由一组词构成，词与词之间没有先后顺序的关系。此外，一篇文档可以包含多个主题，文档中每一个词都由其中的一个主题生成。

在LDA模型中，一篇文档生成的方式如下：

* 从狄利克雷分布中取样生成文档 i 的主题分布；
* 从主题的多项式分布取样生成文档i第 j 个词的主题；
* 从狄利克雷分布中取样生成主题对应的词语分布；
* 从词语的多项式分布中采样最终生成词语。

其中，类似Beta分布是二项式分布的共轭先验概率分布，而狄利克雷分布（Dirichlet分布）是多项式分布的共轭先验概率分布。

此外，LDA的图模型结构如下图所示（类似贝叶斯网络结构）：

  
如果语料中一共有 N 个词，则上帝一共要抛 2N次骰子，轮换的抛doc-topic骰子和 topic-word骰子。但实际上有一些抛骰子的顺序是可以交换的，我们可以等价的调整2N次抛骰子的次序：前N次只抛doc-topic骰子得到语料中所有词的 topics,然后基于得到的每个词的 topic 编号，后N次只抛topic-word骰子生成 N 个word。此时，可以得到：

1. 算法

**困惑度计算**：

**LDA 模型的 Gibbs Sampling 公式为：**

根据公式，我们的目标有两个：

1. 估计模型中的参数 和 ；
2. 对于新来的一篇文档，我们能够计算这篇文档的 topic 分布

训练的过程：

1. 对语料库中的每篇文档中的每个词汇 $\omega$ ，随机的赋予一个topic编号z；
2. 重新扫描语料库，对每个词 $\omega$ ，使用Gibbs Sampling公式对其采样，求出它的topic，在语料中更新；
3. 重复步骤2，直到Gibbs Sampling收敛；
4. 统计语料库的topic-word共现频率矩阵，该矩阵就是LDA的模型；

根据这个topic-word频率矩阵，我们可以计算每一个p(word|topic)概率，从而算出模型参数 , 这就是那 K 个 topic-word 骰子。而语料库中的文档对应的骰子参数 在以上训练过程中也是可以计算出来的，只要在 Gibbs Sampling 收敛之后，统计每篇文档中的 topic 的频率分布，我们就可以计算每一个 概率，于是就可以计算出每一个 。

由于参数 是和训练语料中的每篇文档相关的，对于我们理解新的文档并无用处，所以工程上最终存储 LDA 模型时候一般没有必要保留。通常，在 LDA 模型训练的过程中，我们是取 Gibbs Sampling 收敛之后的 n 个迭代的结果进行平均来做参数估计，这样模型质量更高。

有了 LDA 的模型，对于新来的文档 doc, 我们只要认为 Gibbs Sampling 公式中的 部分是稳定不变的，是由训练语料得到的模型提供的，所以采样过程中我们只要估计该文档的 topic 分布 就好了. 具体算法如下：

a. 对当前文档中的每个单词 , 随机初始化一个topic编号z;

b. 使用Gibbs Sampling公式，对每个词 , 重新采样其topic；

c. 重复以上过程，知道Gibbs Sampling收敛；

d. 统计文档中的topic分布，该分布就是 。

1. 流程

(1) 读取文本数据集，并按照8：2分为训练集和测试集

(2) 定义主题数量范围，循环训练lda模型，并计算其困惑度，画出困惑度与主题数量的折线图

(3) 从中筛选出使得困惑度最低的主题数目，然后根据获得的最佳主题数目，训练LDA模型

(4) 筛选出每个主题的前10个关键词，作为提取代表性游记的依据

5.2.3 基于LSTM的景点名赋诗功能

1. 原理

LSTM算法全称为Long short-term memory，最早由 Sepp Hochreiter和Jürgen Schmidhuber于1997年提出，是一种特定形式的RNN（Recurrent neural network，循环神经网络），而RNN是一系列能够处理序列数据的神经网络的总称。

一般地，RNN包含如下三个特性：

a）循环神经网络能够在每个时间节点产生一个输出，且隐单元间的连接是循环的；

b）循环神经网络能够在每个时间节点产生一个输出，且该时间节点上的输出仅与下一时间节点的隐单元有循环连接；

c）循环神经网络包含带有循环连接的隐单元，且能够处理序列数据并输出单一的预测。

LSTM就是门限RNN，其单一节点的结构如下图所示。LSTM的巧妙之处在于通过增加输入门限，遗忘门限和输出门限，使得自循环的权重是变化的，这样一来在模型参数固定的情况下，不同时刻的积分尺度可以动态改变，从而避免了梯度消失或者梯度膨胀的问题。

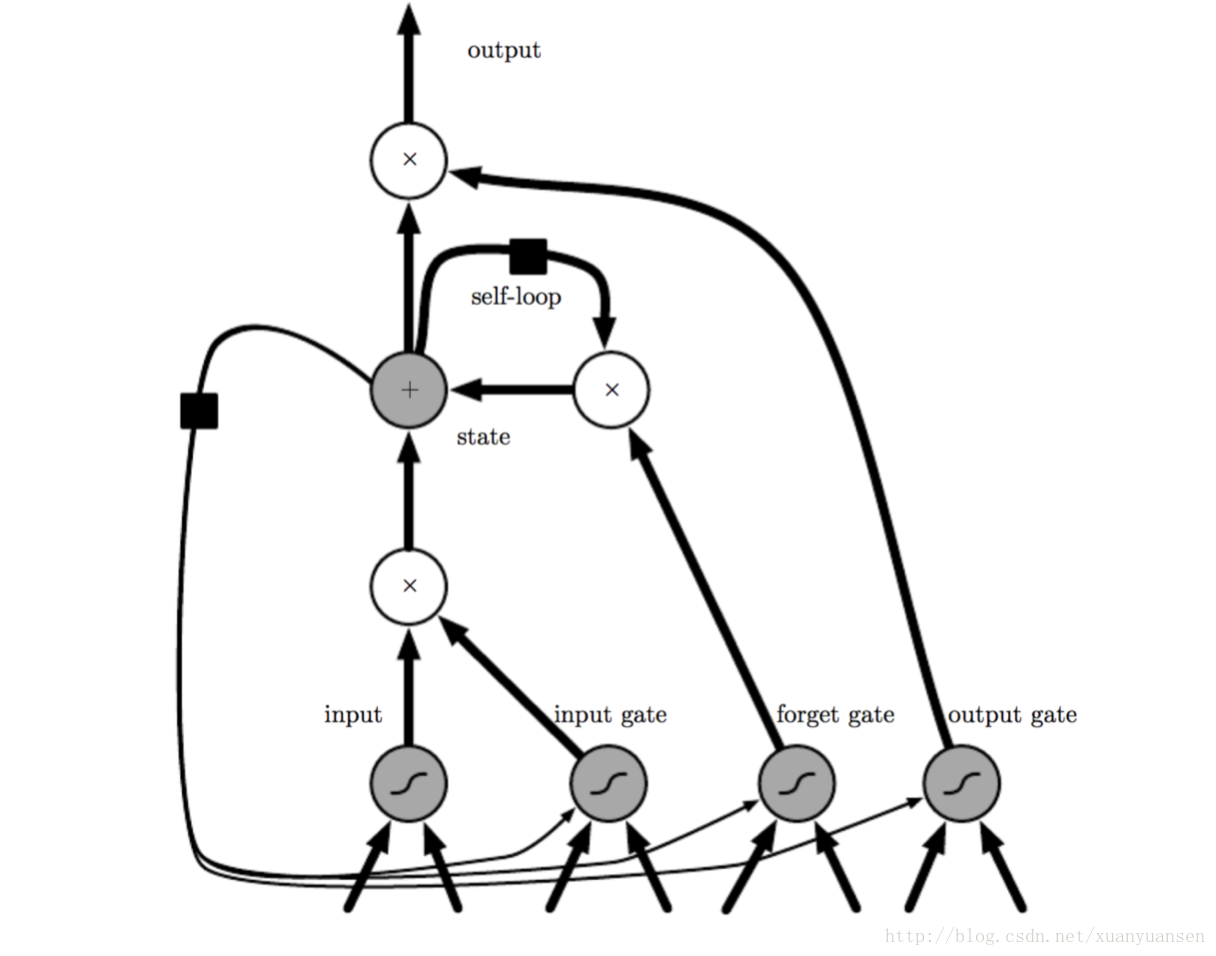


图1 LSTM的CELL示意图

与前馈神经网络类似，LSTM网络的训练同样采用的是误差的反向传播算法（BP），不过因为LSTM处理的是序列数据，所以在使用BP的时候需要将整个时间序列上的误差传播回来。LSTM本身又可以表示为带有循环的图结构，也就是说在这个带有循环的图上使用反向传播时我们称之为BPTT（back-propagation through time）。下面我们通过图2和图3来解释BPTT的计算过程。从图2中LSTM的结构可以看到，当前cell的状态会受到前一个cell状态的影响，这体现了LSTM的recurrent特性。同时在误差反向传播计算时，可以发现h(t)的误差不仅仅包含当前时刻T的误差，也包括T时刻之后所有时刻的误差，即back-propagation through time的含义，这样每个时刻的误差都可以经由h(t)和c(t+1)迭代计算。

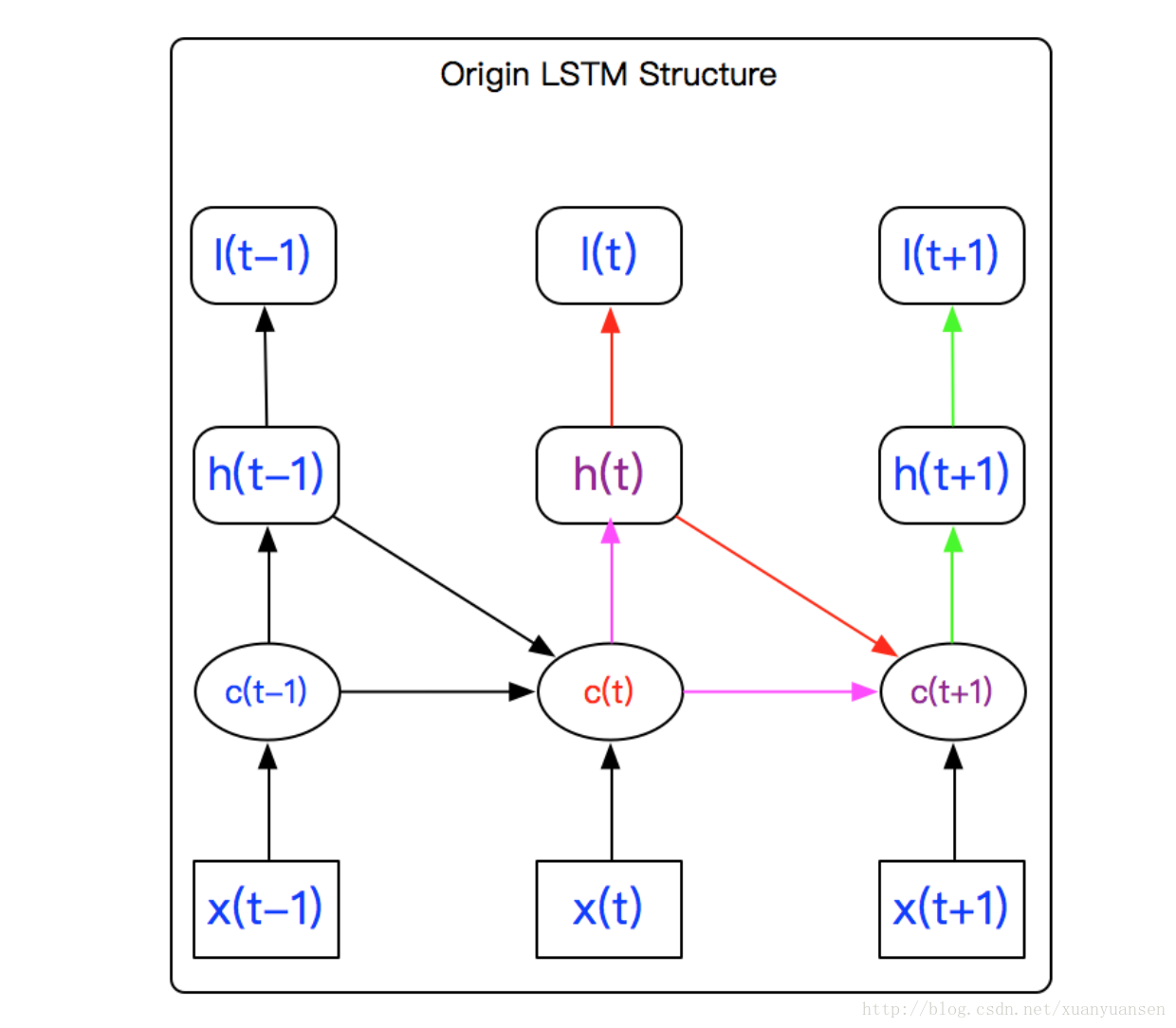
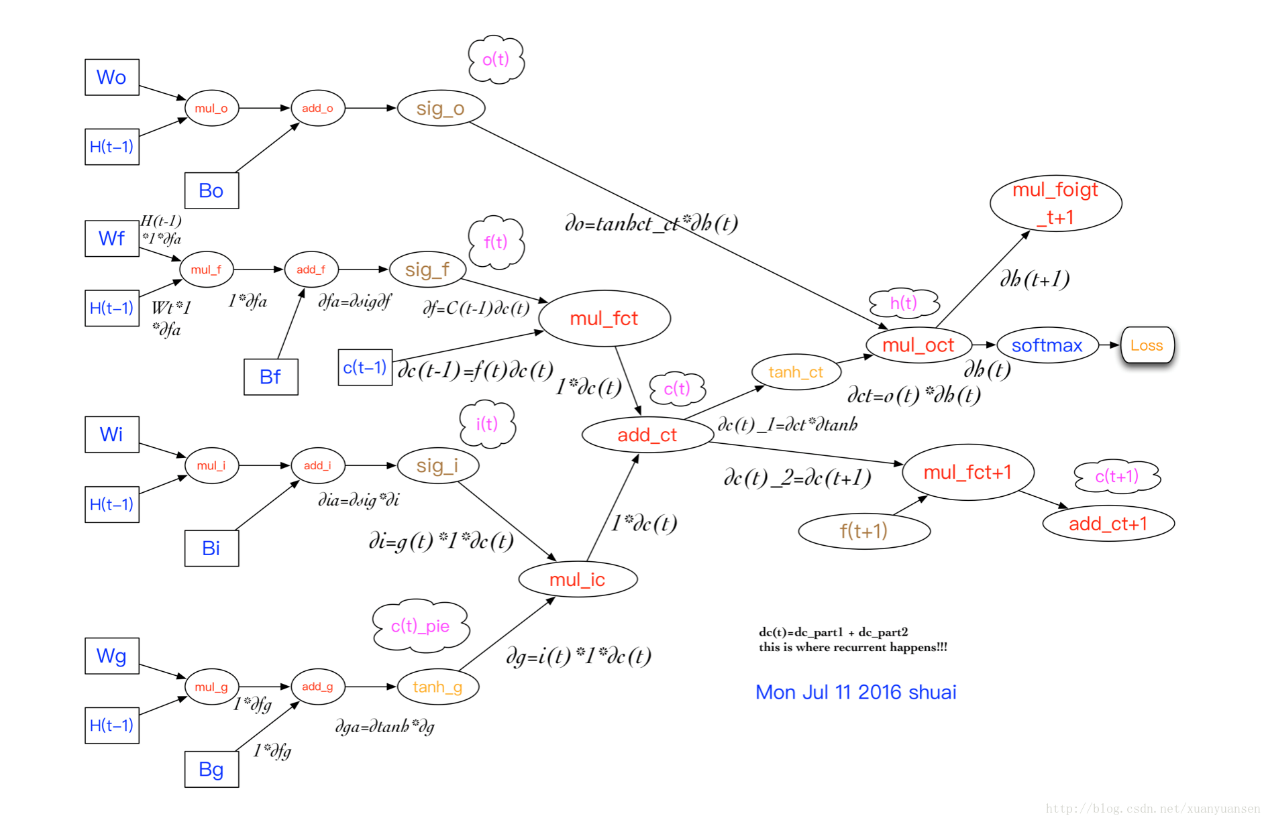


图2 LSTM网络示意图

为了直观地表示整个计算过程，在参考神经网络计算图的基础上，LSTM的计算图如图3所示，从计算图上面可以清晰地看出LSTM的forward propagation和back propagation过程。如图，H(t-1)的误差由H(t)决定，且要对所有的gate layer传播回来的梯度求和，c(t-1)由c(t)决定，而c(t)的误差由两部分，一部分是h(t)，另一部分是c(t+1)。所以在计算c(t)反向传播误差的时候，需要传入h(t)和c(t+1)，而h(t)在更新的时候需要加上h(t+1)。这样就可以从时刻T向后计算任一时刻的梯度，利用随机梯度下降完成权重系数的更新。



LSTM“四部曲”：

第一步：

第一步要决定我们要从细胞状态中丢掉什么信息。LSTM使用“遗忘门”，本质是一个sigmoid层。它通过输出一个0-1的数字来决定丢掉什么信息，“1”代表全部保留，“0”代表全部丢弃。

第二步：

第二步要决定我们在细胞状态中存什么新信息。首先，tanh层创建一个新候选向量。然后“输入门”，本质上就是sigmoid层，决定update什么值。

第三步：

第三步，我们用新的候选值 + ft乘旧状态Ct-1，来得到新状态Ct。

第四步：

最后一步决定我们要输出什么。首先，我们使用一个sigmoid层来决定我们输出细胞状态的什么部分。然后，我们把细胞状态与sigmoid层的输出相乘，再经过一个tanh层，就得到了我们的输出。

2. 模型构建

* 首先使用BasicLSTMCell定义单个基本的LSTM单元。
* 使用dropout方法：网络中每个单元在每次有数据流入时以一定的概率(keep prob)正常工作，否则输出0值，可以有效防止过拟合。
* 使用MultiRNNCell构造多层LSTM，使用zero\_state即可对各种状态进行初始化。

3. 训练及生成

* 增量式训练过程 ：输入原文本{START, X1, X2, …, Xn}，模型输出{Y1, Y2, …, Yn-1, END}。模型输出和原文本{X1, X2, …, Xn, END}。利用交叉熵计算loss。
* 生成过程：输入开始符*START*，预测下一个字，然后将预测结果作为输入，至到预测结束符。
* 数据集：含有3080条古诗句的数据集

5.2.4 基于余弦相似度的旅游线路聚类分析

1. 原理

文本聚类无法采用传统的kmeans或基于密度的dbscan算法，这些算法中计算距离均采用欧式距离，而使用欧式距离衡量文本的相似性显然是极其荒谬的。

本小组使用了余弦相似度来衡量文本的相似性。本小组先将每个线路中的景点按照总景点列表映射成为仅含有0、1的行向量，使用余弦相似度计算向量之间的相似度，其本质是计算向量之间夹角的大小，可以完美避开欧氏距离的缺点，同时不受总景点列表顺序的影响。

2. 算法及步骤

① 获取每条线路的行程 list

② 将每个 list 中的字符串使用结巴分词，得到分词信息

③ 将分词结果与景点名匹配，将匹配到的景点名组合成此线路的景点名 list①，将所有线路涉及到的景点名组合成另一个总景点名 list②

④ 将每条线路的景点名 list① 按照总景点名 list② ，映射成向量

⑤ 使用重写的 cos\_kmeans 聚类（将 kmeans 中的欧氏距离换成余弦相似度，以用于特定的文本聚类）

⑥ 将每一类的中心坐标反向映射为景点名 list③

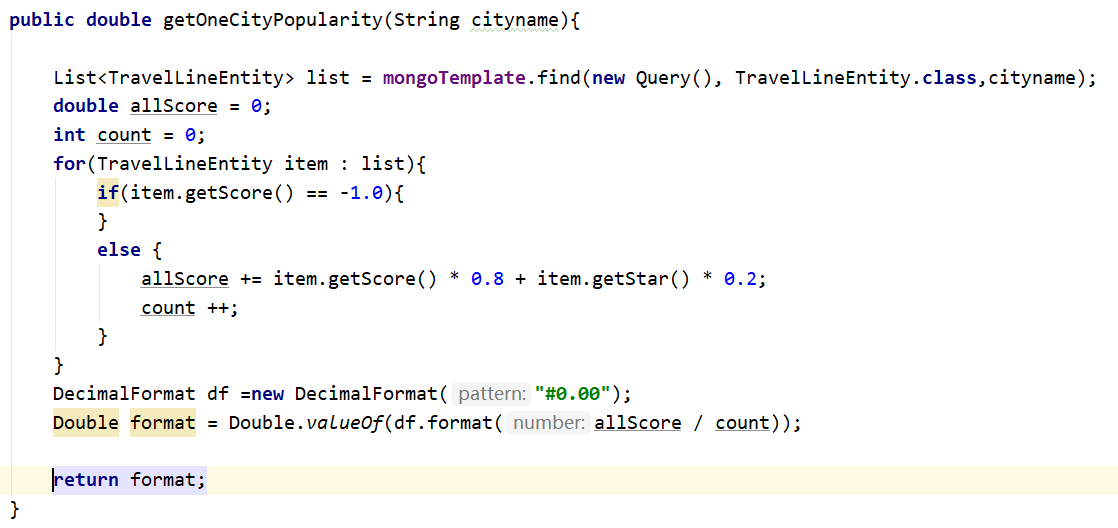
⑦ 将每条线路的景点名 list① 与 景点名 list③ 匹配度高的为一类

# **6. 系统实现**

6.1 基础功能

6.1.1 不同城市的平均评分

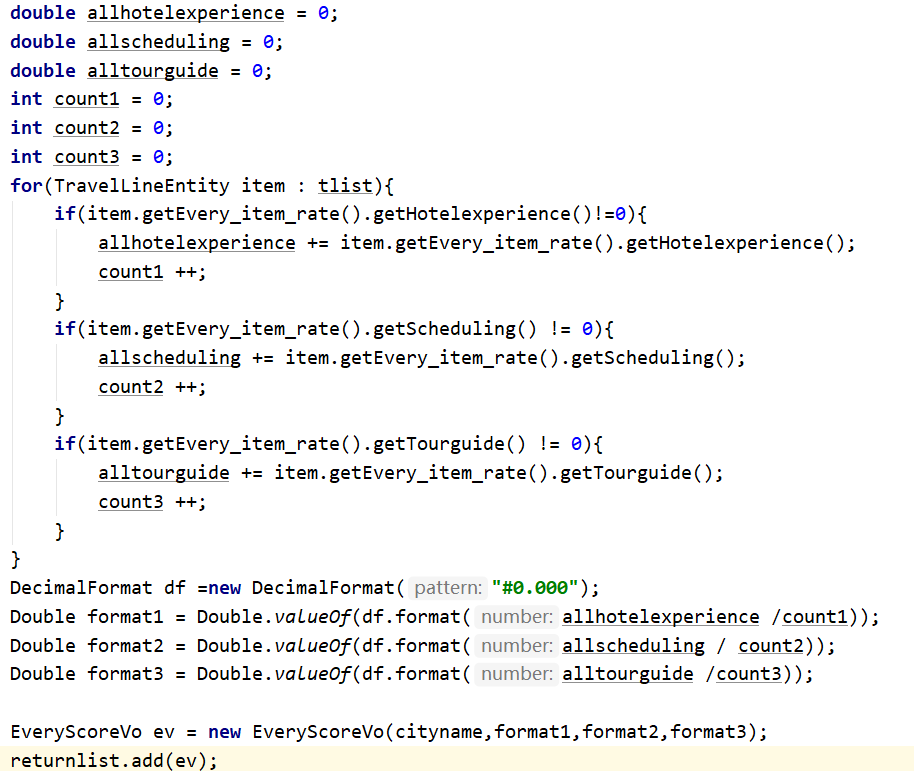
得到单个城市的平均评分



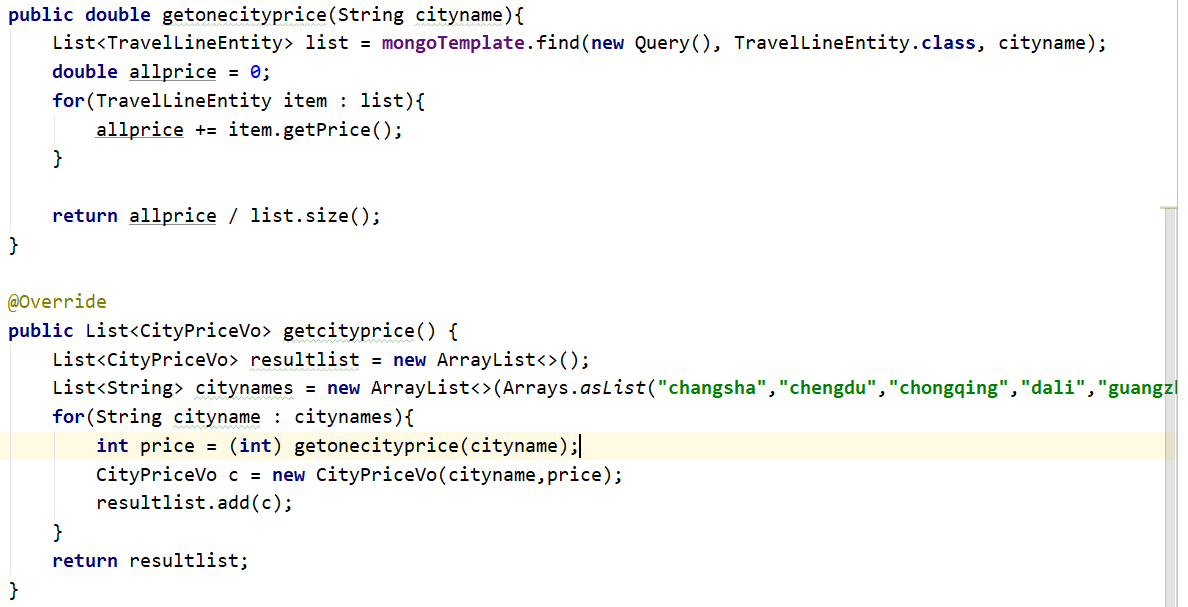
调用得到城市的单个评分的函数后，返回全体城市的评分



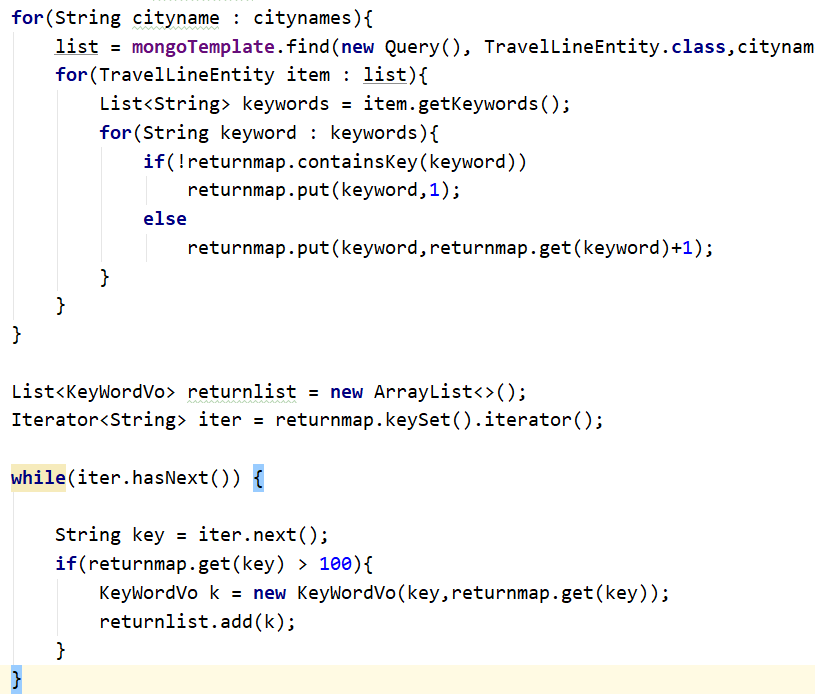
6.1.2 不同城市的行程安排，导游服务，酒店体验评分分析



6.1.3 不同城市旅游价位分析



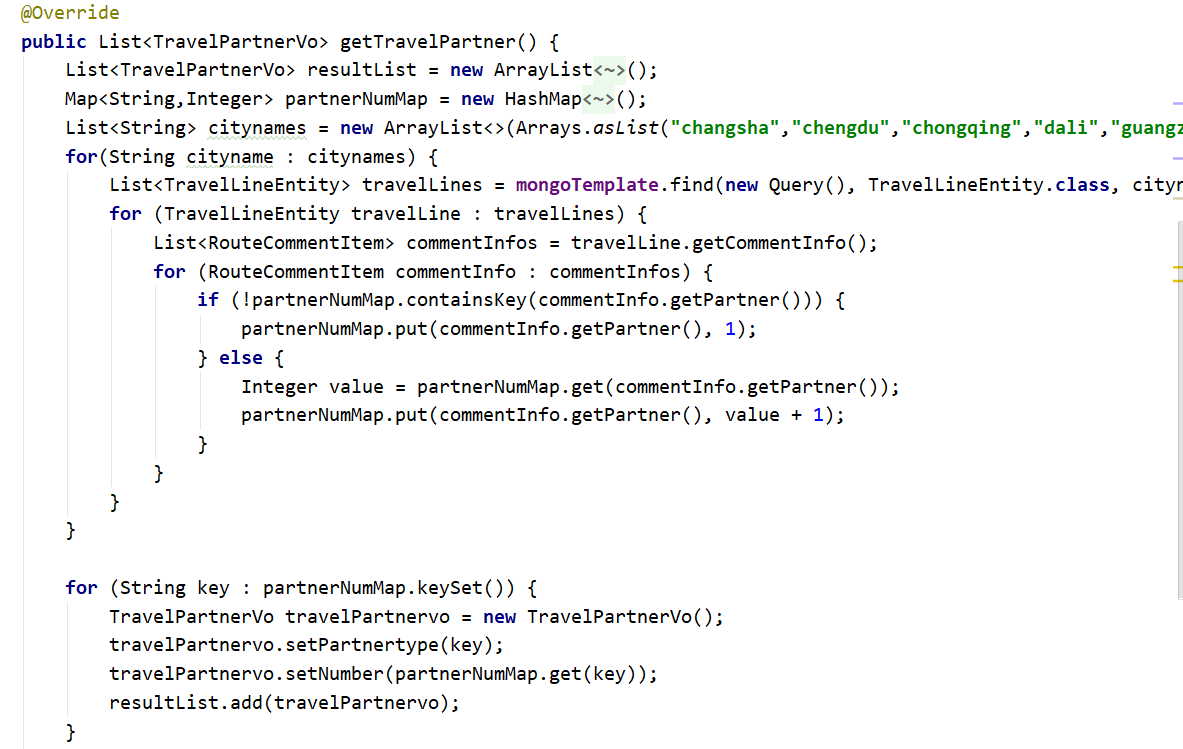
6.1.4 旅游线路点评关键词词云



6.1.5 旅行线路统计分析



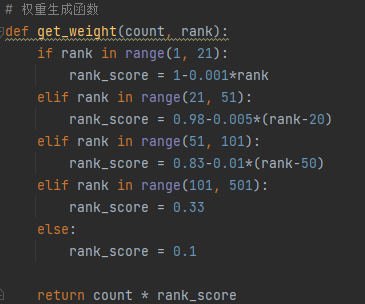
6.1.6 出游伙伴的分析

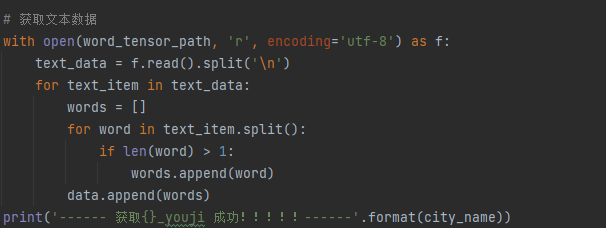


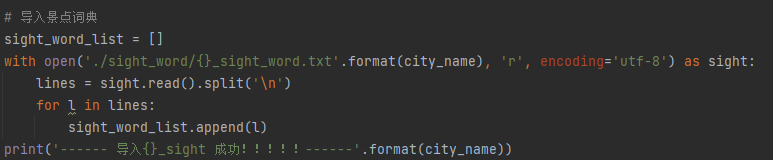
6.2 拓展功能

6.2.1 基于游记的旅游线路推荐

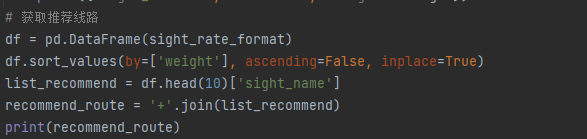
1. 实现过程
2. 导入处理好的文本数据集。
3. 导入景点数据。
4. 筛选并统计景点词频。
5. 计算景点的权重。
6. 比较权重值，筛选出，最佳的5个景点，作为推荐线路。
7. 核心代码











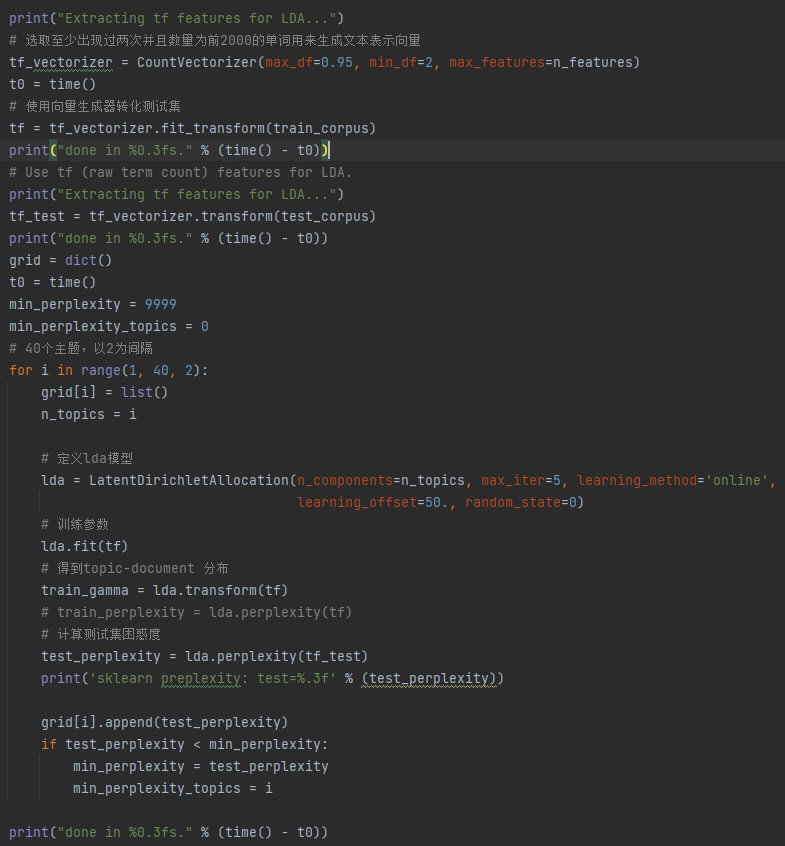
6.2.2 基于LDA的代表性游记提取

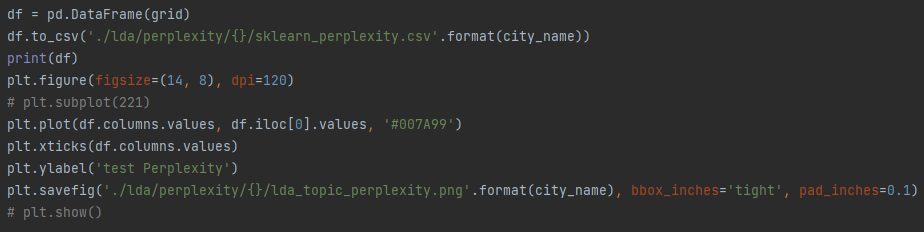
1. 实现过程
2. 读取文本数据集，并按照8：2分为训练集和测试集
3. 定义主题数量范围，循环训练lda模型，并计算其困惑度，画出困惑度与主题数量的折线图
4. 从中筛选出使得困惑度最低的主题数目，然后根据获得的最佳主题数目，训练LDA模型
5. 筛选出每个主题的前10个关键词，作为提取代表性游记的依据
6. 核心代码

**困惑度计算**

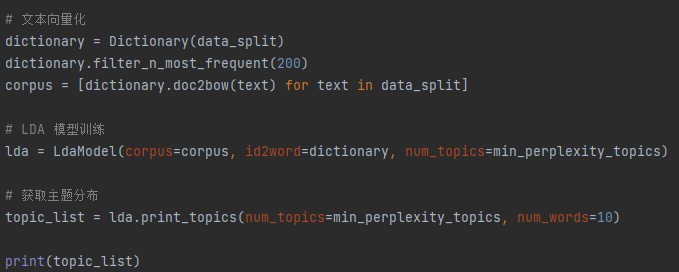








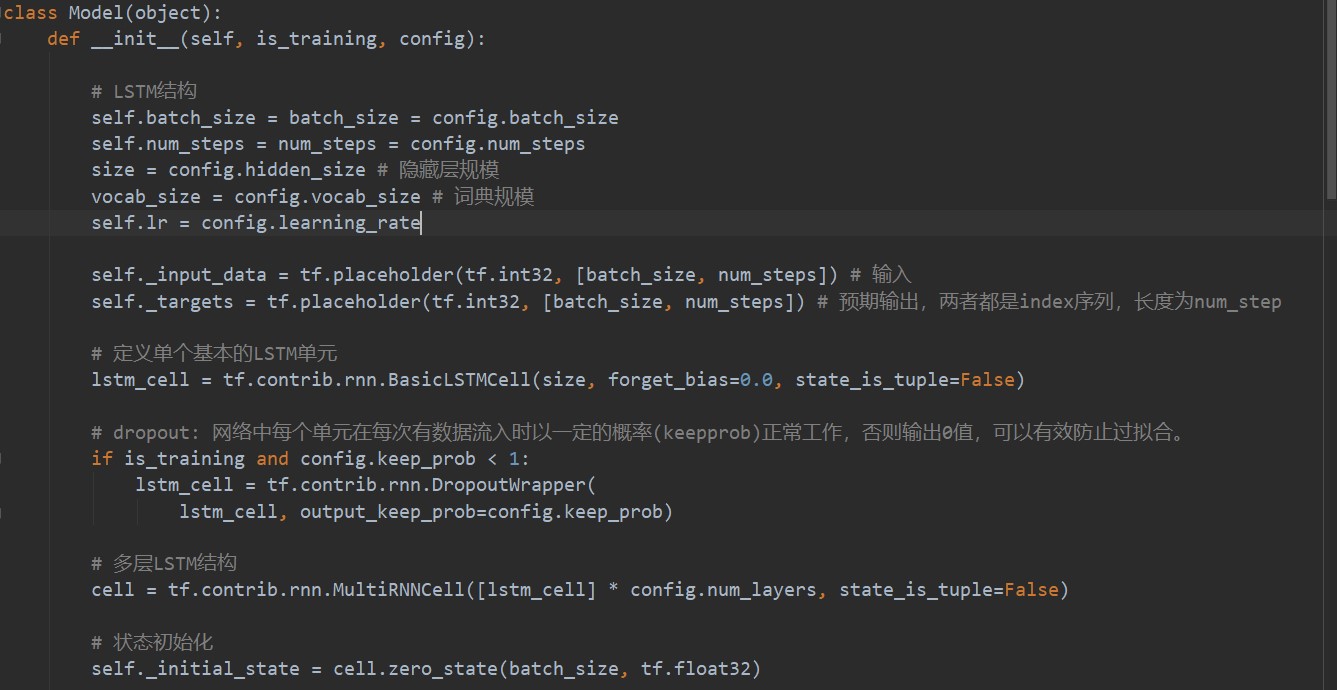
**LDA模型训练**

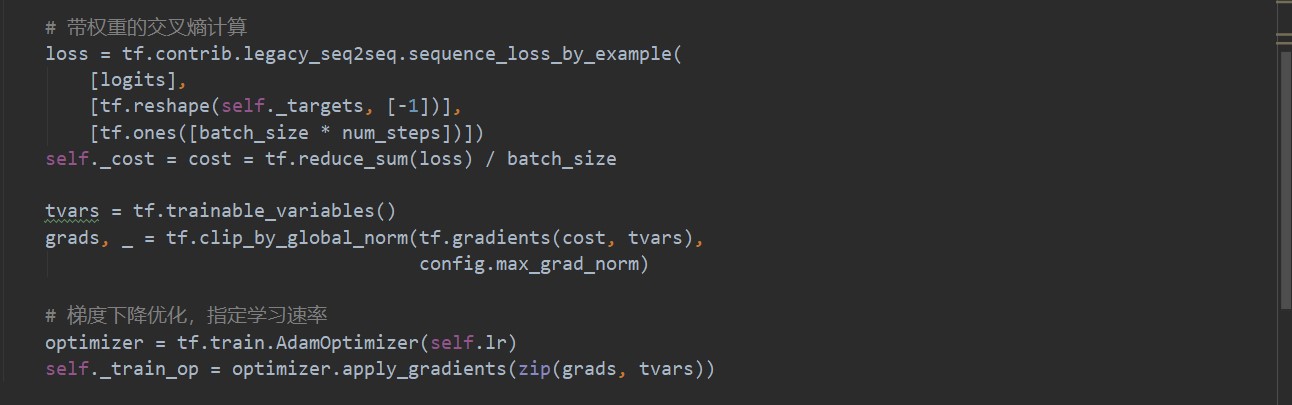


**主题游记提取**

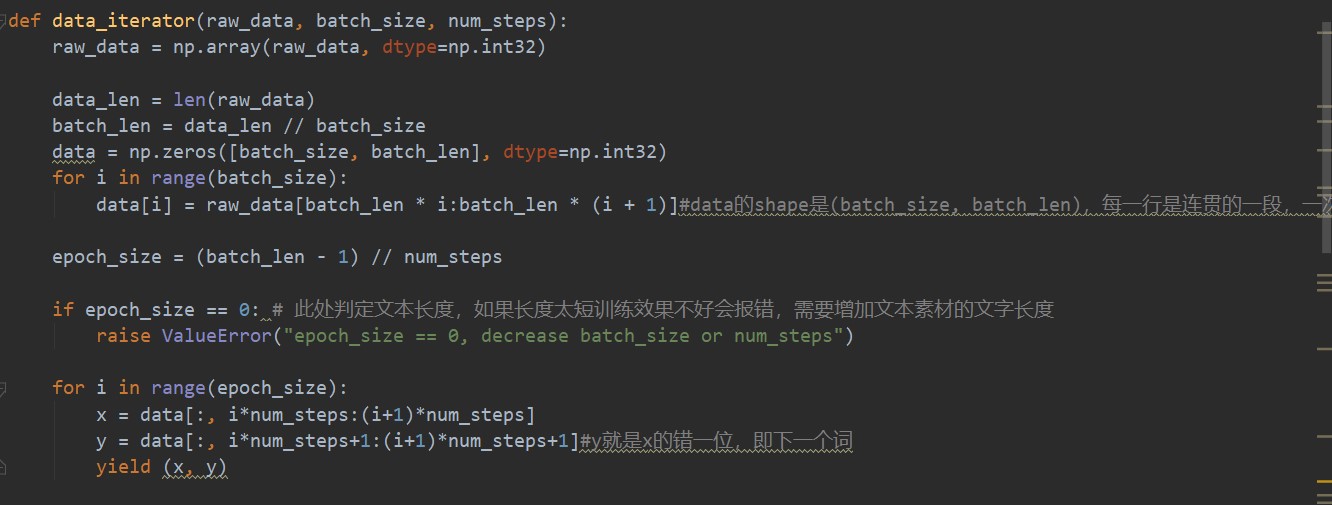
6.2.3 基于LSTM的景点名赋诗

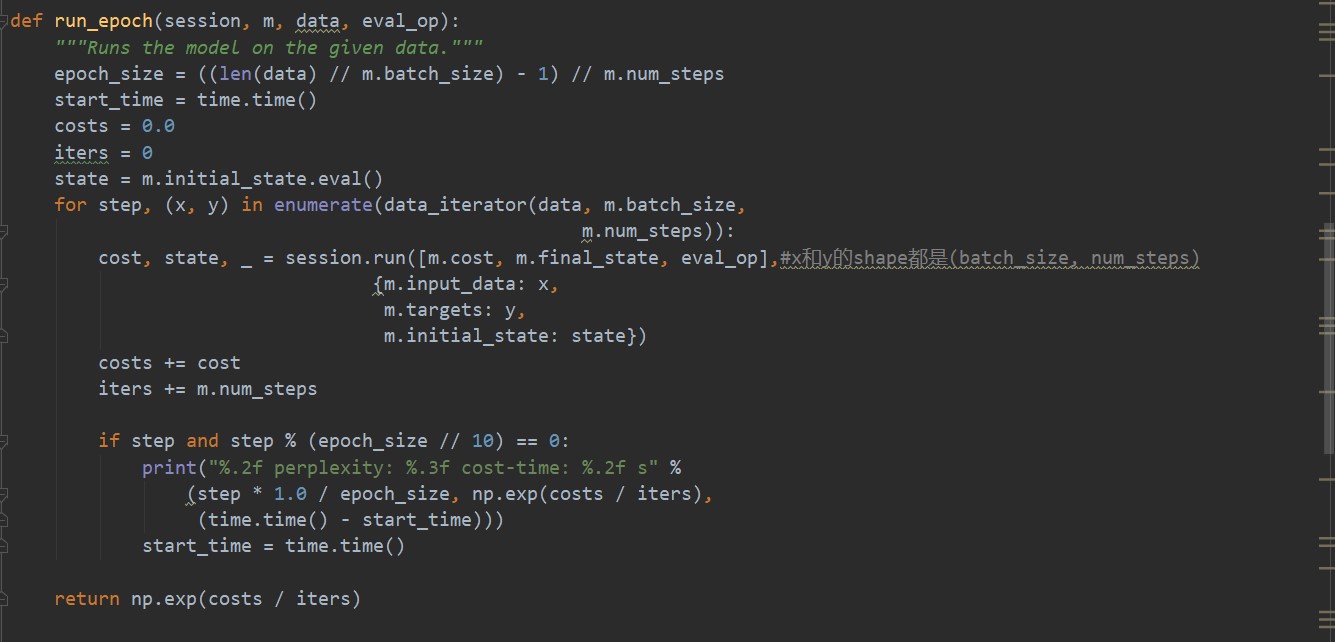
Model：



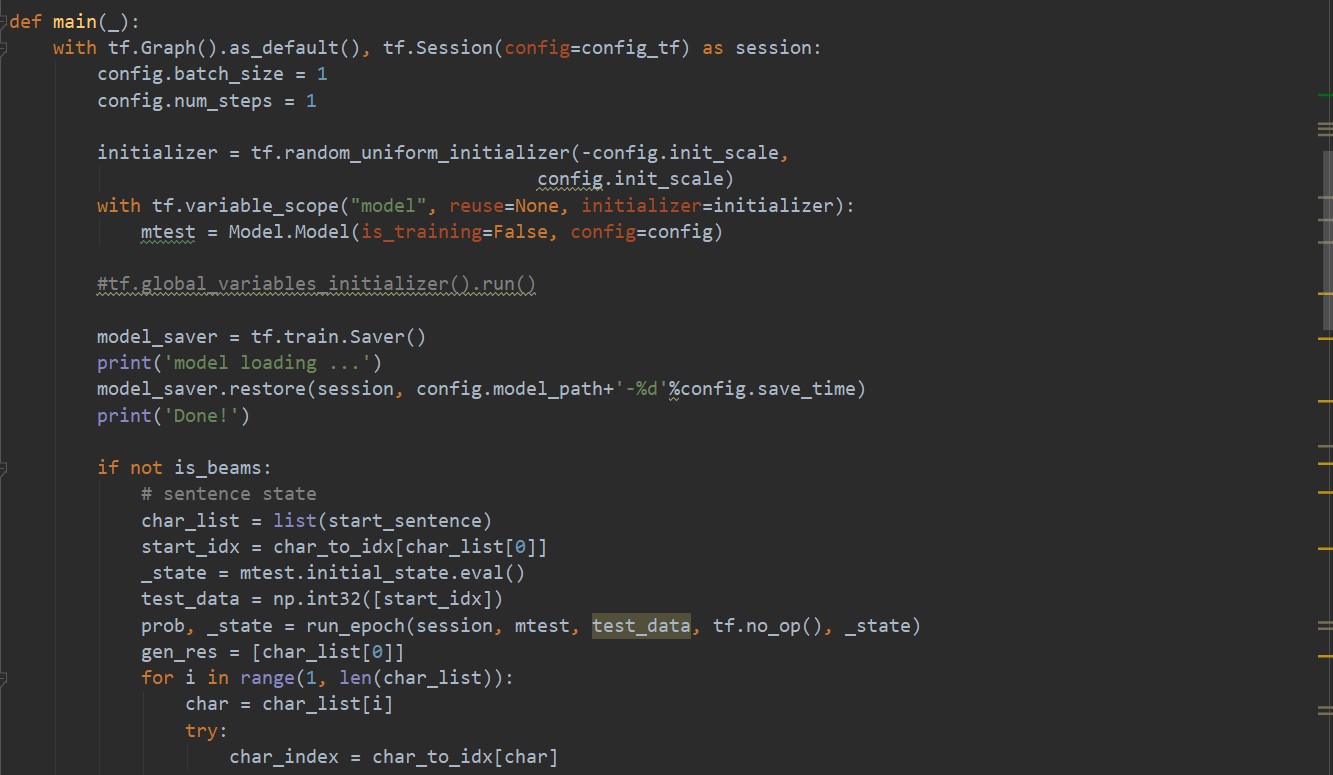


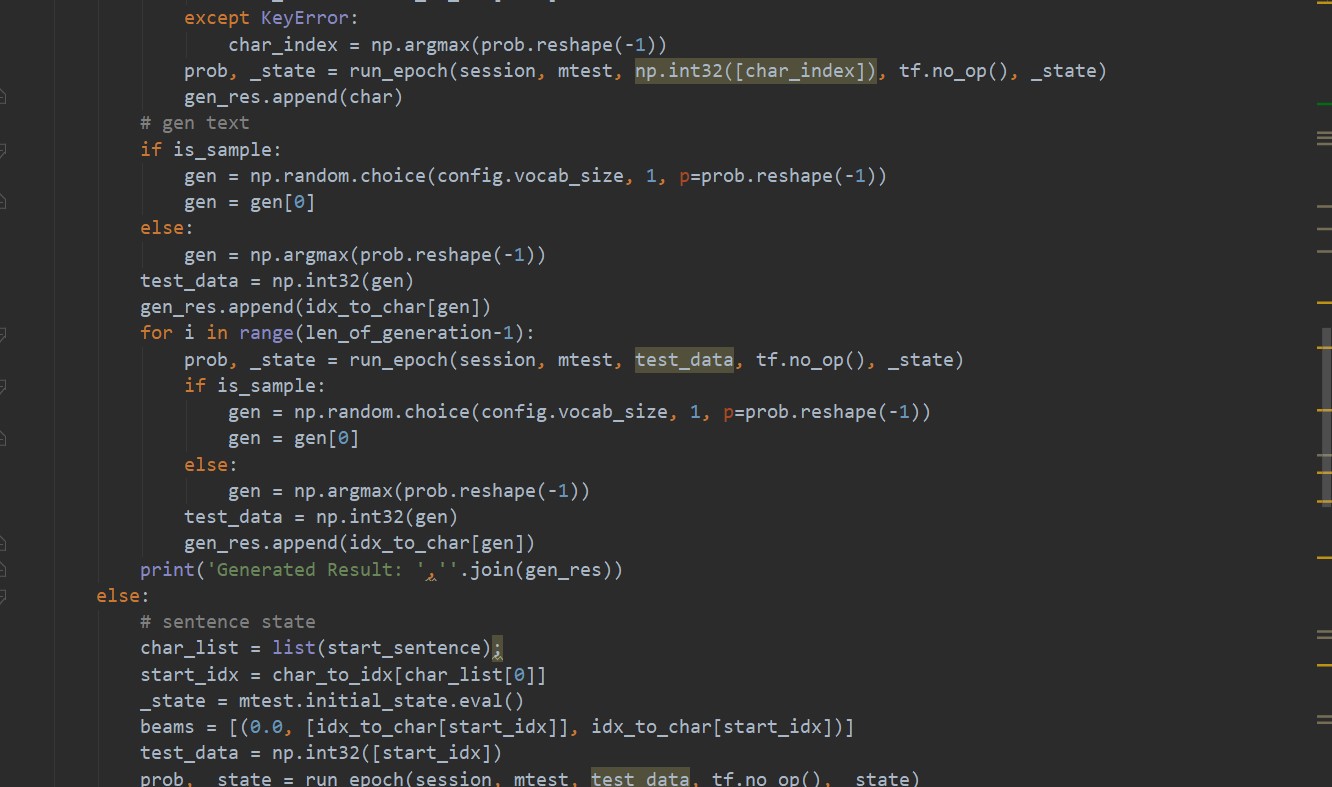
Train:

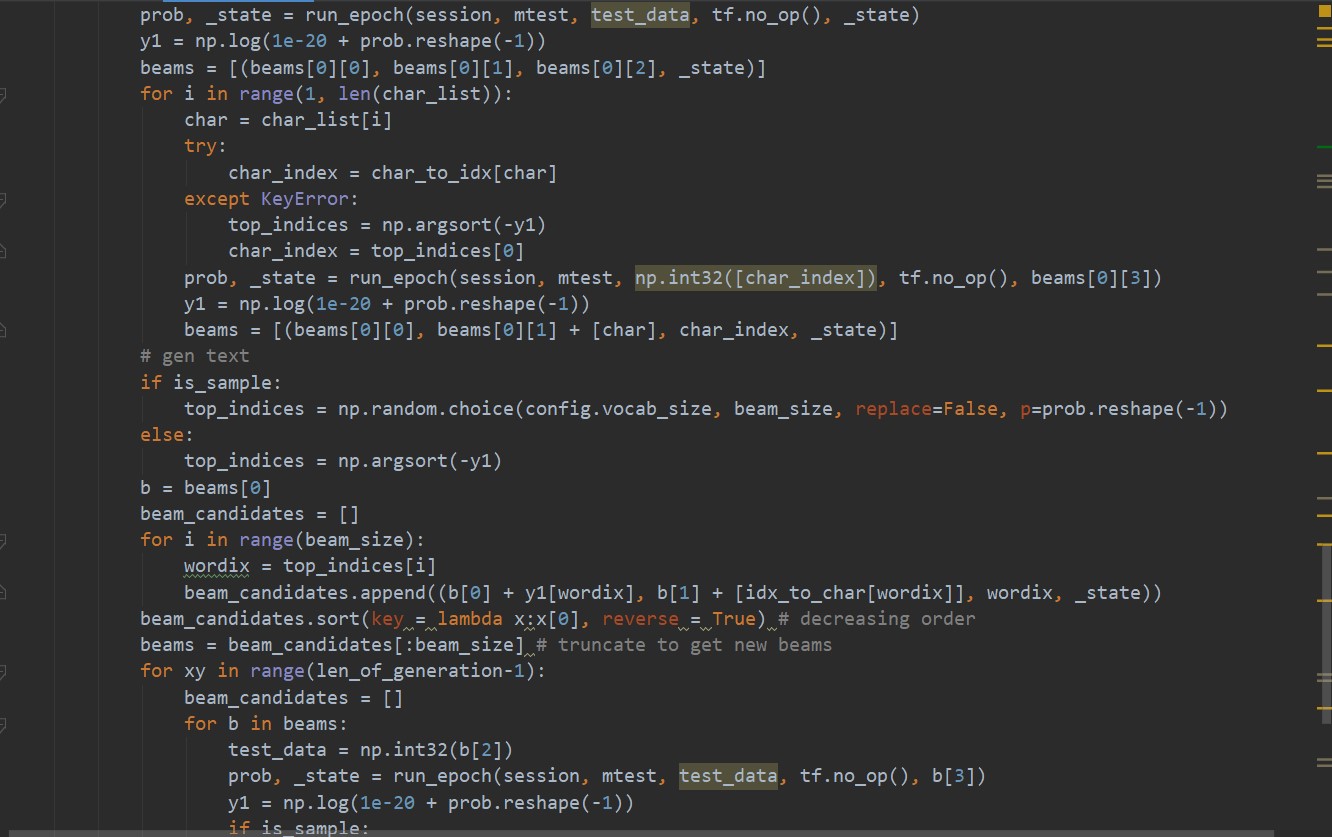


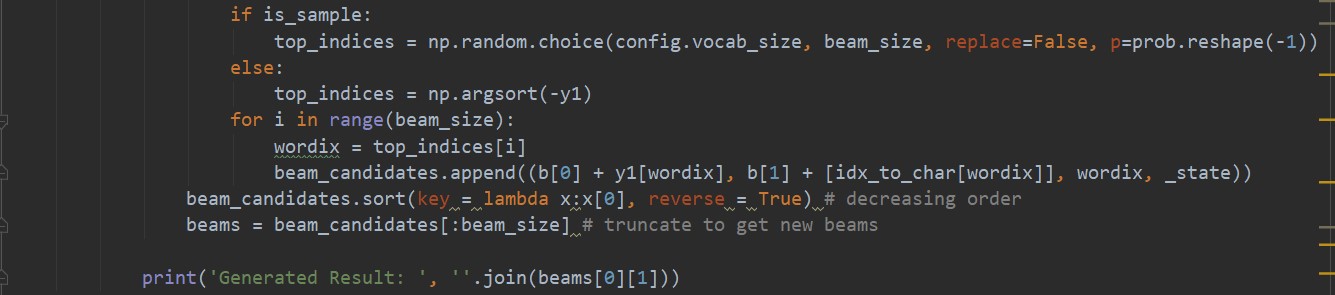


Generate:

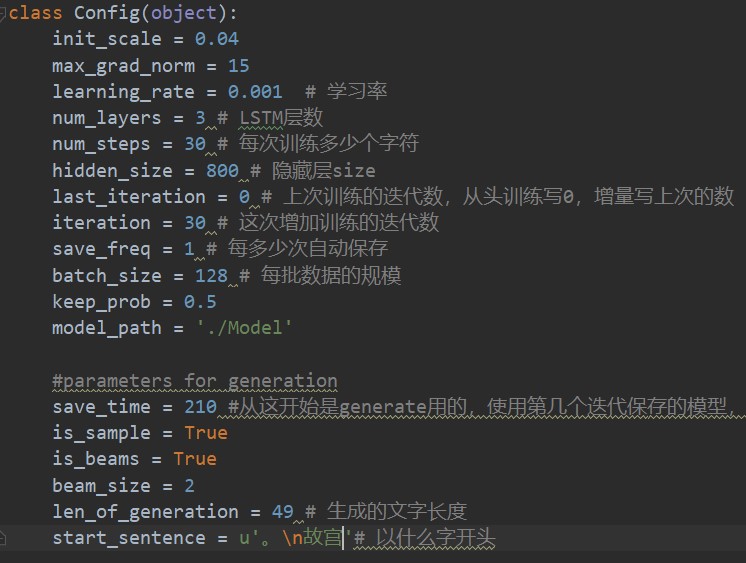






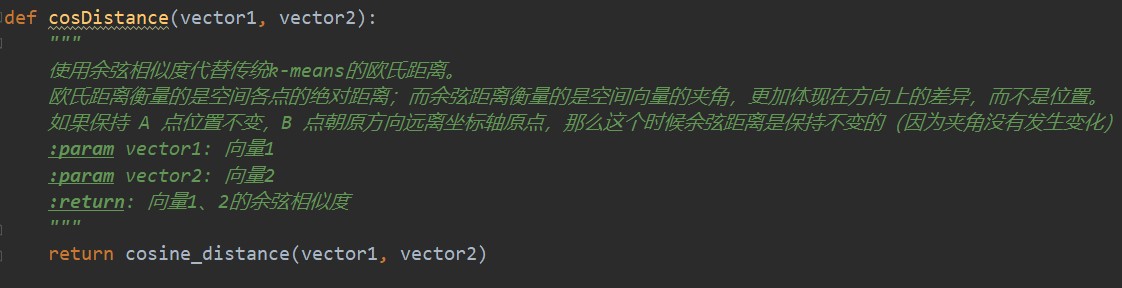


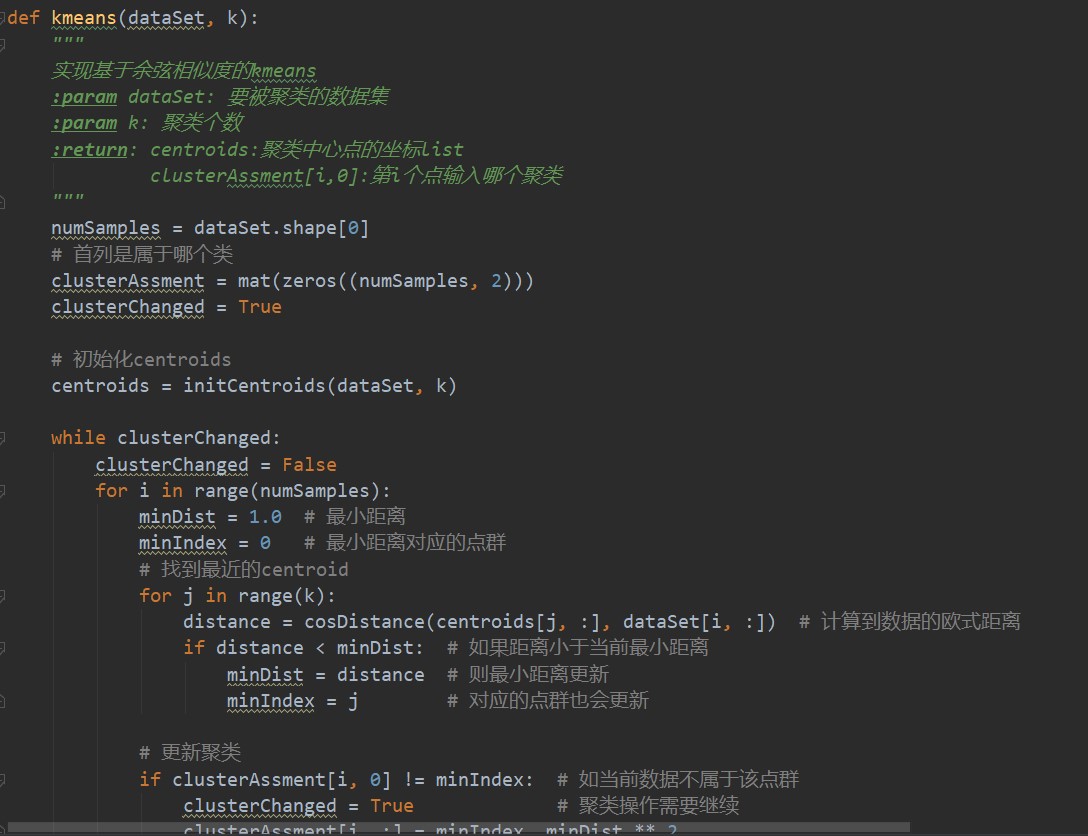
Config:



6.2.3 基于余弦相似度的旅游线路聚类分析

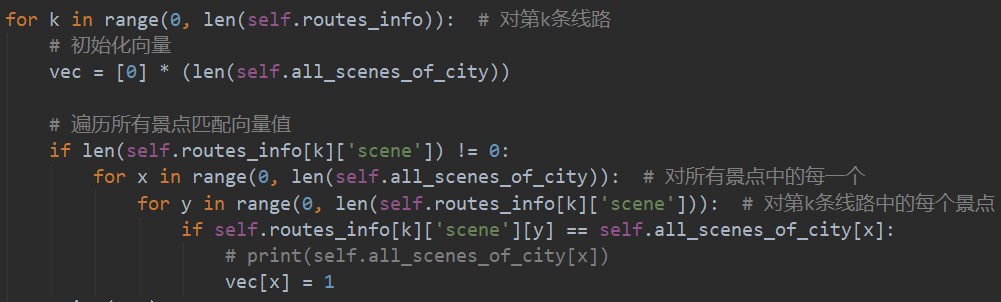
cos\_kmeans:





聚类：

核心代码：映射向量



# **7. 系统测试**

7.1 基础功能数据展示页面

7.1.1不同城市的平均评分

图像以每两秒的速度变化，循环展示



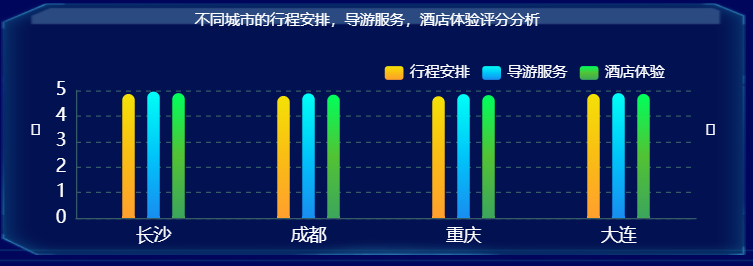
7.1.2出游伙伴分析



7.1.3旅游线路统计分析



7.1.4 不同城市的行程安排，导游服务，酒店体验评分分析



7.1.5不同城市旅游价位分析



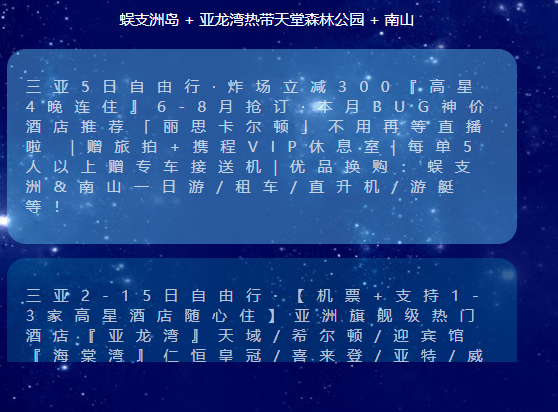
7.1.6 词云



7.2 旅游线路聚类功能数据展示

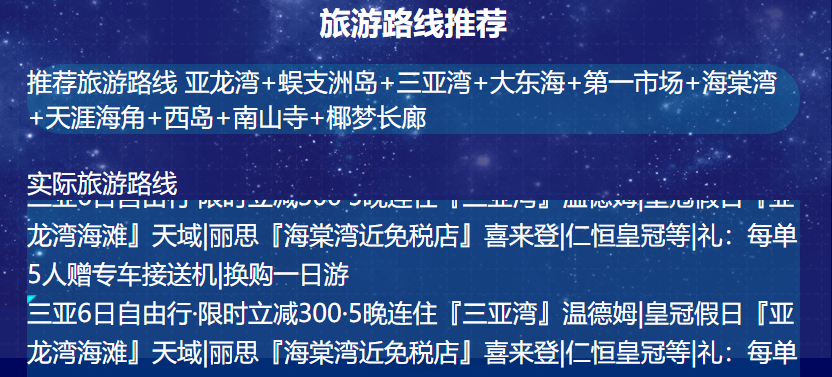
每个聚类内部可以滚动显示更多线路





7.3 对于城市推荐相关内容功能数据展示

7.3.1 旅游路线推荐



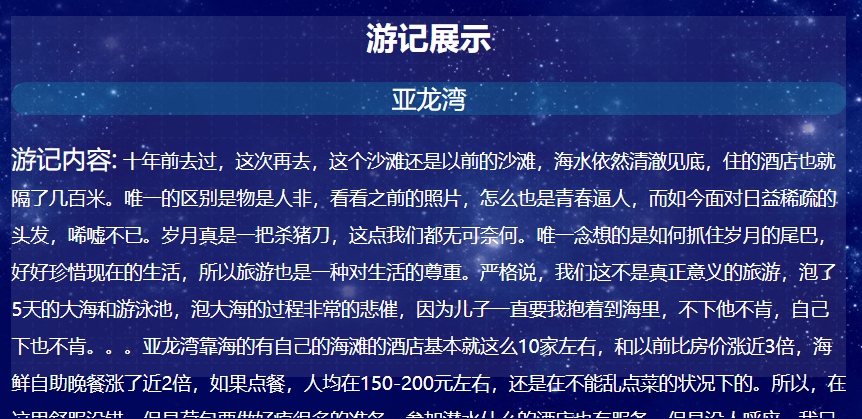
7.3.2 景点推荐



点击景点，可以展示：

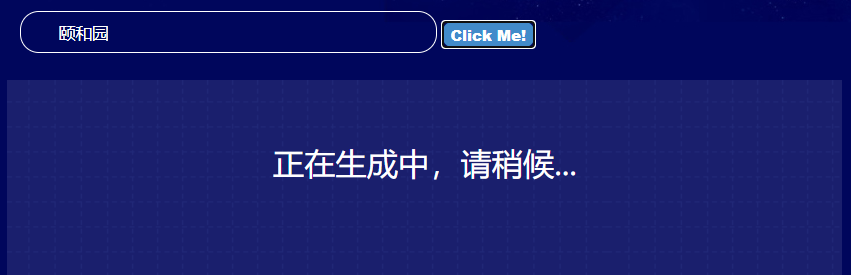


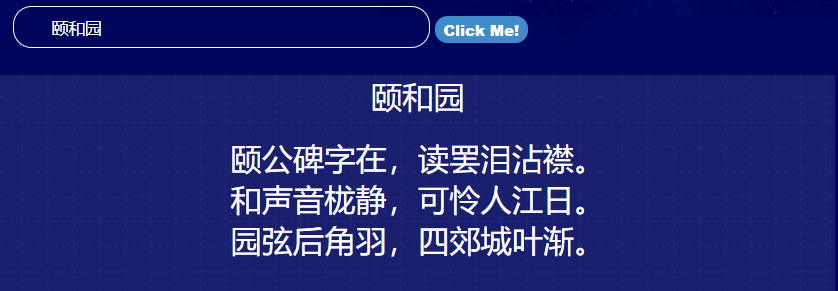
7.3.3 基于LDA的主题游记推荐



7.4 景点诗词生成

输入景点名称，以颐和园为例：





**8. 小组分工**

|  |  |  |
| --- | --- | --- |
| **数据爬取及预处理** | **旅游线路数据爬取** | **米思成** |
| **游记数据爬取** | **杨文涛** |
| **旅游景点数据爬取** | **张瑞成** |
| **功能** | **基础功能数据处理及分析** | **张和钰、平云** |
| **基于LDA的主题游记功能** | **杨文涛** |
| **基于LSTM的景点名赋诗功能** | **米思成** |
| **基于余弦相似度的旅游线路聚类分析** | **米思成** |
| **旅游线路推荐功能** | **杨文涛** |
| **可视化** | **Springboot搭建** | **张和钰、平云** |
| **前端Echarts图表展示** | **张和钰、平云** |