**自然语言与机器翻译**

**作业**



**学 院 计算机科学与工程学院**

**专 业 计算机科学与技术**

**班 级 计硕2005班**

**学生姓名 张靖男**

**学 号 2001865**

**二Ο二Ο年十二月十七日**

摘 要

中国古典诗歌作为传统文化的一大瑰宝，以其独特的魅力吸引着越来越多的人, 藏头诗作为其中的一种，既实用又有娱乐性，能够在日常生活广泛使用。然而，由于诗歌在平仄押韵等方面的要求，让无数普通人对于藏头诗的学习甚至创作只能扼腕叹息。为此，本文以LSTM模型为基础，设计并实现了藏头诗自动生成系统，为评价生成效果，设计实现了古诗自动评价系统。

本文的主要工作是诗歌数据集的构建，藏头诗生成系统的实现以及古诗自动评价的实现。数据集的搭建，包括获取网页资源、数据清洗以及基于编辑距离对诗歌数据进行去重，最终获得371,883首唐代至当代的五言、七言诗。生成系统的实现，则是运用LSTM模型，完成五言或七言、藏头或藏尾诗的生成，且每次生成的诗歌不同的功能。评价系统的实现，是运用AHP模型，结合诗歌的押韵、平仄、流畅度、内容丰富度以及主题鲜明度各方面，对诗歌进行评价打分。本文还基于Tornado框架和vue.min.js框架，设计实现了用于展示藏头诗自动生成系统的web界面。

本文设计完成的藏头诗自动生成系统，以及为评价生成结果，实现的古诗自动评价系统，能够丰富人们的文化生活，并为以后的诗歌自动生成与评价的学习与研究工作提供参考。本系统将在数据集的扩建、更多模型的尝试以及古诗自动评价方面的扩展等方面，进一步完善。

关键词：LSTM；藏头诗；古诗自动评价；诗歌自动生成

**目录**

[摘 要 I](#_Toc59269276)

[1 作业要求 5](#_Toc59269277)

[1.1 TASK 5](#_Toc59269278)

[1.2 本文选题 5](#_Toc59269279)

[2 需求分析 6](#_Toc59269280)

[2.1 系统实现目标 6](#_Toc59269281)

[2.2 系统功能性需求 6](#_Toc59269282)

[2.2.1 数据采集需求 6](#_Toc59269283)

[2.2.2 藏头诗生成系统需求 7](#_Toc59269284)

[2.2.3 古诗自动评价系统需求 8](#_Toc59269285)

[2.3 系统非功能性需求 9](#_Toc59269286)

[2.4 本章小结 9](#_Toc59269287)

[3 系统设计 10](#_Toc59269288)

[3.1 系统设计原则 10](#_Toc59269289)

[3.2 总体设计 10](#_Toc59269290)

[3.3 模块设计 11](#_Toc59269291)

[3.3.1 诗歌数据集模块 11](#_Toc59269292)

[3.3.2 诗歌训练模块 13](#_Toc59269293)

[3.3.3 诗歌生成模块 15](#_Toc59269294)

[3.3.4 诗歌自动评价模块 15](#_Toc59269295)

[3.3.5 系统web服务模块 20](#_Toc59269296)

[3.4 本章小结 21](#_Toc59269297)

[4 系统实现 22](#_Toc59269298)

[4.1 系统开发环境 22](#_Toc59269299)

[4.2 诗歌数据集模块实现 22](#_Toc59269300)

[4.2.1 诗歌数据抓取 23](#_Toc59269301)

[4.2.2 诗歌数据清洗 23](#_Toc59269302)

[4.3 模型训练与诗歌生成实现 24](#_Toc59269303)

[4.3.1 LSTM模型训练 24](#_Toc59269304)

[4.3.2 诗歌生成实现 26](#_Toc59269305)

[4.4 诗歌自动评价实现 27](#_Toc59269306)

[4.4.1 AHP模型 27](#_Toc59269307)

[5.4.2 平仄、押韵评价实现 28](#_Toc59269308)

[4.4.3 流畅度评价实现 29](#_Toc59269309)

[4.4.4 关键词和主题鲜明度评价实现 29](#_Toc59269310)

[4.5 Web展示系统实现 30](#_Toc59269311)

[4.5.1 前端实现 31](#_Toc59269312)

[4.5.2 后端实现 31](#_Toc59269313)

[4.6 本章小结 32](#_Toc59269314)

[5 系统测试 33](#_Toc59269315)

[5.1 生成诗歌效果测试 33](#_Toc59269316)

[5.1.1 藏头诗生成测试 33](#_Toc59269317)

[5.1.2 藏尾诗生成测试 34](#_Toc59269318)

[5.2 诗歌自动评价测试 34](#_Toc59269319)

[5.3 Web端测试 36](#_Toc59269320)

[5.4 测试结论 37](#_Toc59269321)

[5.5 本章小结 37](#_Toc59269322)

[6 总结与感想 39](#_Toc59269323)

[6.1 总结 39](#_Toc59269324)

[6.2 感想 40](#_Toc59269325)

[附录A 诗歌生成结果 41](#_Toc59269327)

# **1 作业要求**

本章节的主要内容为本课程项目作业要求重述。

## **1.1 TASK**

可选任务：

1. 体验NiuTensor.NMT, 尝试利用模型进行翻译。
2. 编写代码，实现任务一中的batch训练
3. 比比谁的模型更强，使用同样的数据，将前馈网络语言模型的ppl降到更低
4. 自选其他任务，如：分词，机器翻译，问答，阅读理解，命名实体识别等，最好fork NiuTensor项目完成

要求：将代码上传到github中，在项目的ReadMe.md文件中说明做的任务，达到了某种程度（如语言模型使用PPL指标），使用的框架不限。

## **1.2 本文选题**

本课题将设计实现一个藏头诗自动写作系统，研究如何将深度学习技术用于藏头诗的自动生成，主要研究工作如下：

1. 收集藏头诗的相关数据，清洗后用于模型训练。
2. 开发藏头诗自动写作系统，做到输入关键字，输出藏头诗。
3. 对藏头诗自动写作系统进行优化和改进，提升系统性能。
4. 开发基于Web的展示系统，可以把系统封装成api接口，对外提供服务。

# **2 需求分析**

## **2.1 系统实现目标**

本系统的实现目标是实现藏头诗和藏尾诗的自动生成，并最终将生成的诗歌结果通过界面美观、操作便捷的web平台展示给使用者。

## **2.2 系统功能性需求**

功能性需求，指的是开发人员必须在产品中实现的软件功能。本系统的功能性需求主要包括以下三方面：数据采集需求、藏头诗生成系统需求以及古诗自动评价需求。数据采集需求是实现本系统的基础，藏头诗生成系统需求是完成本系统藏头诗生成的要求，而古诗自动评价需求则是对生成诗歌进行评价的实用要求。

### 2.2.1 数据采集需求

数据采集是本系统的一个非常关键的步骤，数据量的大小，数据质量的好坏将直接影响后续藏头诗生成质量。

诗歌这一类的数据集，看似简单易得，但是很难获得较全面的数据集。经过调查发现，在github上有已整理好的古诗词数据集，其中包含大量的古诗数据，但是其中也包含大量残句和缺字诗，且朝代仅有唐、宋两代，还需要进行大量补充。在一开始，受到五言诗起源于汉乐府的影响，收集的是汉代至近现代的诗歌。后发现，古体诗（唐代前）和近体诗（唐代及之后）在格律要求方面有较大差异，及时调整为采集唐代至当代的五言、七言诗。

在诗歌自动评价中，对于诗歌主题的评价中，需要对诗歌主题进行预测，这需要采集按主题分类的诗歌数据。经调查发现，没有整理好的诗歌数据集，且诗歌网站上按诗歌主题分类的数据量较少，这对接下去工作的进行提出了较大的挑战。

正因为较全面的诗歌数据集稀少，本文所搭建的唐代至当代的诗歌数据集，为将来进行诗歌相关研究的人提供便利。

本系统在数据准备阶段，需要的数据有：藏头、藏尾诗模型训练数据和测试集。为保证生成模型的质量，本系统的数据量初步要求为至少三十万首唐代至当代的五言、七言诗。

### 2.2.2 藏头诗生成系统需求

藏头诗生成系统的开发过程中，主要分为三部分内容：一是诗歌数据集的建立，二是诗歌生成模型，三是古诗自动评价。其中，诗歌数据集的建立，包括数据采集工作以及数据清洗工作，数据采集包括从开源数据集获取以及网页爬取诗歌数据，数据清洗操作则包括去脏数据以及对清洗后的数据集进行去重；诗歌生成模型中，包括模型训练，诗歌生成以及web平台搭建，诗歌的生成，需要满足藏头或藏尾的选择、五言或七言的选择；古诗自动评价部分，则包括综合评价和五方面评价，五方面分别是平仄、押韵、流畅度、关键词以及主题鲜明度评价。

对于开发人员，图2.1展示的是藏头诗自动生成系统的用例图。

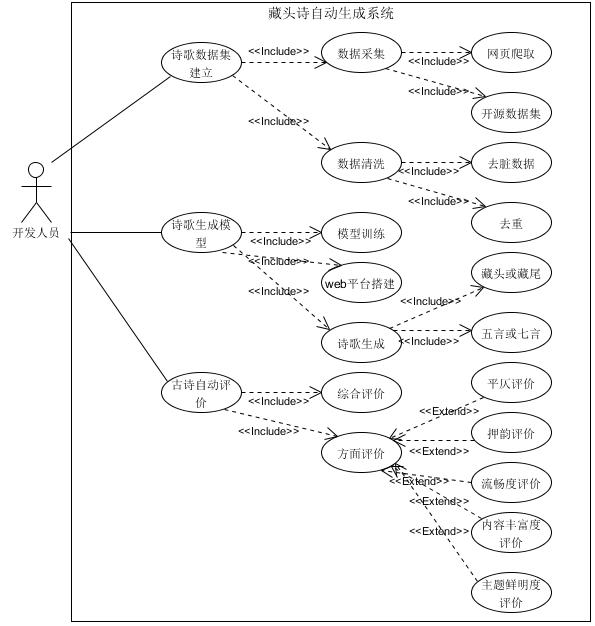


图2.1 藏头诗自动生成系统用例图

本系统主要功能是实现藏头诗与藏尾诗的自动生成，并能自行选择五言或七言。在设计阶段将本系统分为以下两个部分：

第一部分为客户端，用户在提供的界面上进行操作，界面主要包括以下几项：

(1) 两个选择按钮，让用户选择生成藏头或藏尾诗，五言或七言诗。

(2) 一个文本输入框，用来输入藏头或藏尾词。

(3) 一个提交按钮，让用户提交信息，并从客户端传输到服务器端。

(4) 一个文本显示区，用来显示生成的藏头或藏尾诗。

第二部分为服务器端，后端在系统运行中会实时接受前端传送过来的数据，并对数据进行输入预处理，将处理后的数据送入生成模型中，最后将输出的结果返回到客户端并显示。这两块内容包括：

(1) 输入预处理：将从客户端获得的藏头词，判断是否有无效字符，长度是否符合要求，是否有未出现在字符库的汉字，并进行相应的处理，将结果传给模型。

(2) 生成部分：接受处理后的数据，用事先训练好的模型，生成要求的诗作，并传给客户端。

### 2.2.3 古诗自动评价系统需求

目前对古诗生成进行评价的方式，往往是人工打分，专家评价，缺乏对古诗自动评价的系统，无法像机器翻译任务的BLEU，有一个为大众接受的主要评价指标。为对自动生成系统生成的藏头诗、藏尾诗进行自动评价，需要一个自动评价机制。

由于对于古诗的评价，较为复杂，涉及押韵、平仄、流畅度、是否言之有物以及主题的集中程度多个方面，且这些方面难以定量进行描述。参考数学建模中的评价类模型——层次分析法模型。AHP算法在诗歌质量评价中的层次结构图如图2.2所示。

图2.2 AHP层次结构图

在图2.2中，目标层是判断诗歌质量，准则层是押韵、平仄、流畅度、内容丰富度以及主题集中程度。押韵、平仄是诗歌在格律方面的要求，流畅度、内容丰富度以及主题集中程度则是对诗歌内容方面的要求。

## **2.3 系统非功能性需求**

系统除了功能性需求外，为满足客户需求，还有一些非功能性需求，以保证系统的流畅使用和健壮性。本小节通过介绍藏头诗自动生成系统中部分模块的非功能性需求，来展示本系统的可靠性、健壮性和易用性。

(1) 可靠性：可靠性是指整个系统能够平稳运行，且极少出现不稳定的情况。本系统使用的是技术成熟的LSTM模型，搭建模型时使用的是开源的tensorflow库函数。开发的Web网页不涉及注册登录等涉及用户权限的问题，可见本系统的可靠性。

(2) 鲁棒性：本系统有针对输入进行预处理的部分，对用户的不同输入，若是规范要求以外的输入，将判断错误类型，并进行相应处理后，再送入生成模型中，这大大提高了本系统的鲁棒性。

(3) 易用性：本系统针对的用户是对藏头诗感兴趣的普通人，其对算法技术等不了解，不会操作命令行等工具，因而简洁的操作界面是非常重要的。本系统不仅界面友好，而且操作方便，这保证了系统的易用性。

## **2.4 本章小结**

本章首先介绍了该系统的实现目标，然后从系统功能性需求和非功能性需求出发，分析了藏头诗自动生成系统。其中，功能性需求从数据采集需求、藏头诗生成需求和古诗自动评价需求三部分，来实现从采集原始数据，到生成藏头、藏尾诗，并对生成诗歌进行自动评价的过程。非功能性需求是保证系统平稳运行的必要条件，主要从系统的可靠性、鲁棒性以及易用性进行了阐述。

# **3 系统设计**

本章的主要内容是根据需求分析进行的系统总体设计，先介绍系统设计原则，然后分为总体设计和模块设计两部分进行阐述。根据整体架构进行每个模块的功能分解，提供实现每个功能点的算法理论和技术方法。

## **3.1 系统设计原则**

系统设计原则主要包括以下几点：

(1) 实用性：藏头诗自动生成系统的设计目的是使用户能够使用该系统进行藏头诗、藏尾诗的生成，实用性是系统最基本的要求；

(2) 可靠性：服务的质量需要可靠性来保证，在设计过程中，应该使用可靠的技术，增强系统鲁棒性，降低出现错误的风险。在执行诗歌生成任务时，模型往往是24小时运行在服务器上，这要求更具可靠性的系统。

(3) 可操作性：藏头诗自动生成系统充分考虑用户的使用习惯，操作简单；将各个模块分别封装，用户使用时不需要考虑实现细节，可操作性强。

(4) 可扩展性：系统在设计时，要考虑未来可能的扩展需求，尽量降低各模块之间的耦合程度。具体来说，在诗歌数据集的搭建方面，将诗歌按朝代进行统计；在诗歌生成方面，用户可以添加藏字的其他位置。

(5) 可维护性：系统在设计时，要考虑将来扩展、更新功能时进行维护的需求。在藏头诗自动生成系统中，训练模型的参数以及训练数据的变化，需要及时记录相关信息，以便进一步进行调整。

(7) 先进性：系统在开发过程中，为提高系统的扩展性以及系统的开发效率，应选用先进的、主流的、成熟的技术。其中，数据爬取使用的是Beautiful Soup库，藏头诗自动生成系统使用的是LSTM模型，都是目前非常成熟的技术。

## **3.2 总体设计**

藏头诗自动生成系统包括诗歌数据集模块、诗歌训练模块、诗歌生成模块、诗歌自动评价模块以及藏头诗自动生成系统web服务模块。

其中，诗歌语料库模块主要负责建立诗歌语料库，其首先从各大古诗歌网抓取诗歌数据，然后按照规则提取诗歌内容，并将清洗后得到的诗歌数据根据Levenshtein Distance计算字符串相似度来进行去重，得到诗歌数据集；诗歌训练模块主要负责从整理好的诗歌数据集中提取训练数据，运用LSTM算法，进行诗歌向量从当前状态到下一状态的模型训练，并调节参数；诗歌生成模块主要负责根据已经训练好的模型，从初始状态，一步步得到藏头或藏尾诗；诗歌自动评价模块主要负责对生成的诗歌进行打分评价，和其他诗歌进行对比；藏头诗自动生成系统web服务模块是提供一个web服务框架，以便系统对外提供服务。

如图3.1展示的是系统的总体架构图。



图3.1 系统总体架构图

## **3.3 模块设计**

在整个系统中，各个模块是较为独立的。其中，诗歌数据集模块是提供诗歌语料给诗歌训练模块使用；诗歌训练模块的输出是诗歌生成模块和诗歌自动评价模块中流畅度分析的模型文件；诗歌自动评价模块是对诗歌生成模块的结果进行评价；最后，诗歌生成模块和诗歌自动评价模块的结果，会在系统web服务中展示。所以，整个系统也是一个分层的架构。

### 3.3.1 诗歌数据集模块

诗歌数据集模块主要工作包括采集数据和清洗数据。

#### 3.3.1.1 采集数据

采集数据，从两个方向入手，一是调查已有的较权威的数据集，例如前面提到的github上的唐宋代的诗歌数据集，二是从网页上爬取诗歌。下面主要介绍从网页爬取诗歌数据。

要构建诗歌数据集，需要获取大量诗歌资源。使用搜索引擎搜索关键词，如：“古诗”、“诗词”、“古诗文”等，搜索引擎反馈给我们含有诗歌资源的网站。最终我们选择诗词学习网(https://www.shicixuexi.com)，古诗文网（https://www.gushiwen.org），以及古诗文网（https://www.gswen.cn）三个网站作为数据来源。选择理由如下：

(1) 这三个网站的诗歌可以按照朝代进行搜索，这大大方便了我们后续按朝代进行分类去重等操作。

(2) 这三个网站的html语句简洁明了，这为我们大批量地爬取数据提供了便利。

(3) 这三个网站的诗歌数据量较大，资源相对集中。

接下来，以诗词学习网为例，介绍本系统从网页爬取诗歌数据的设计。爬取数据的流程图如图3.2所示。



图3.2 采集数据流程图

#### 3.3.1.2 清洗数据

从网页采集到的诗歌数据中，含有一些html语句、注释、残句、词等脏数据。本系统使用正则表达式的方式来消去数据中的如“<br>”、“注：”、“①”等特殊符号，括号内解释内容，题记等信息，删去含有残句、体裁为词的作品。词，又称长短句，判断时，先将各句长度不等，不是五言或七言的删去，对于《生查子》、《木兰花》等词牌名的作品，各句字数相等，且均为五言或七言，再通过标题匹配删去。

由于数据的来源不同，难免有重复，去重是必不可少的一步。经清洗后的数据集，利用Levenshtein库函数，来度量两首诗的相似程度，从而对搜集到的诗歌数据进行去重。下图3.3是数据清洗的流程图。



图3.3 数据清洗流程图

### 3.3.2 诗歌训练模块

诗歌训练模块，主要包括定义输入数据的input和target，划分训练数据和测试数据，以及构建LSTM模型。

#### 3.3.2.1 训练数据划分及定义

由于收集的诗歌长度不一致，有绝句、律诗以及长诗。一方面为统一诗向量长度，另一方面为更充分利用收集到的诗歌数据。将诗歌按每四句一首进行划分，原371,883首诗的数据集，扩展成743209首四句诗的数据集。

定义input及target。input是当前状态的诗向量,target是下一个状态的诗向量。举例来说，如果input是“窗前明月光，疑是地上霜。举头望明月，低头思故乡。”，那么target则是“前明月光，疑是地上霜。举头望明月，低头思故乡。]”其中，“]”为终止补充字符。

划分训练集和测试集。由于数据量较大，我们从原数据集中随机抽取10000首作为测试集，其余部分作为训练集。使用的是train\_test.py，其中为保证各数据集的互异性，需要将抽取出的测试集数据从整个数据集中删去。

#### 3.3.2.2 LSTM模型应用部分

在获得诗歌数据集，并定义input和target，以及划分完训练集和测试集后，我们构建LSTM模型进行训练。下图3.4是模型训练的流程图。



图3.4 模型训练的流程图

如图所示，图的左半边是准备数据的操作，先生成字表，记录字-数字ID的对应关系，然后根据这个对应关系，将汉字形式的诗歌，转化为诗向量的形式，在生成训练集和测试集的诗向量后，按batch\_size大小划分batch，从而进行分批次训练。数据准备的一系列操作为接下来的模型训练做准备。图右边是模型训练部分，先准备训练环境，如果有已训练过的模型，先载入，然后开始训练，训练过程中，可以通过tensorboard查看训练结果，最后，当loss值稳定后，训练完成。

### 3.3.3 诗歌生成模块

诗歌有五言及七言之分，在藏头诗的生成中，也需要可以指定生成五言或七言诗；藏头诗可以藏头，也当然可以藏其他位置，藏头和藏尾也是需要完成的功能；一次生成多首诗以供选择，进一步，每次生成不同的诗歌。下表3.1介绍了预计实现的功能和采用的方法。

表3.1 实现功能及方法

|  |  |  |
| --- | --- | --- |
| 预计功能 | 实现方式 | 补充 |
| 五言或七言 | 新生成的字符为逗、句号时，判断当前句子长度是否符合要求，若不符合，重新生成该句。 | 可在数据中添加五言或七言信息，如五言诗开头加“5”；或分为两个不同模型进行训练 |
| 藏头或藏尾 | 分成两个模型，分别训练 | 无 |
| 每次生成诗歌不同 | 生成下一个字符时，在生成概率最大的一百个中，随机选择，作为下一个字符。 | 无 |

### 3.3.4 诗歌自动评价模块

诗歌自动评价模块选用的是AHP算法。

首先需要确定诗歌质量有哪些影响因素，押韵、平仄这两点是一首诗的基本要求，接下去决定一首诗的质量的便是诗歌的流畅度以及诗歌的内容和主题。

从诗歌的内容角度而言，一首诗歌是否言之有物，是否内容丰富，可以通过提取的诗歌关键词，来进行判断。诗歌讲究“文以意为主”，虽然不同主题的优劣总是“仁者见仁，智者见智”，不能笼统地下结论，但一个好的作品，应该有鲜明的主题，即主题应当集中。

因此，决定将押韵、平仄、流畅度、内容丰富度以及主题鲜明度，作为诗歌自动评价中考虑的因素。

古代汉语分平、上、去、入四声，平仄中的“平”指四声中的平声，也就是现代汉语中的第一和第二声，“仄”指四声中的上、去、入三声，入声在现代普通话中已经消失，上声和去声分别是现代汉语中的第三和第四声。

入声，属于特殊情况，需要单独考虑。查阅资料，发现对入声规律的研究，构造入声的判断函数。了解押韵的规则，考虑特殊情况入声的存在。如图3.5所示，描述的是诗歌押韵评价的流程图。

平仄描述的是古代汉语的声调，对于诗歌而言，五绝、七绝有着不同的平仄简表，也就是诗歌创作的平仄要求规范，有着4类：平起首句押韵、平起首句不押韵、仄起首句押韵以及仄起首句不押韵，分别对应五绝和七绝，共有了8种平仄简表。根据押韵时判断的首句是否押韵的结果，对应相应的平仄简表，比对符合情况，从而对诗歌平仄，做出评价。

图3.6描述的是对诗歌平仄评价的流程图。



图3.5 押韵评价的流程图 图3.6 平仄评价流程图

我们认为，对于诗歌，它的流畅度越高，那么其生成的概率也越大。那么可以通过生成概率的比较，来判断诗歌流畅度的大小。流畅度的计算是对训练模型的使用的另一种方式，由当前状态，推出下一个字符的可能情况和对应概率，根据实际下一字符，得知该字出现的概率大小。每个出现字符的概率，连乘的积就是当前诗的生成概率，也就是我们所求的流畅度大小。如图3.7描述的是流畅度的计算方式。

对于一首诗歌是否内容丰富，我们通过提取诗歌的关键词进行判断。通过查阅资料，得知两种关键词提取算法，分别是TF-IDF算法和TextRank算法。

经实验后，发现TextRank算法提取的关键词质量较高，且多为意象词，但对于一些诗歌，提取不出关键词；TF-IDF算法提取的关键词存在部分整句被提取的现象，但提取的数量较TextRank算法不同。

于是，决定取两种算法获得关键词的并集，作为诗歌的关键词集合。提取的关键词中，含有部分停用词，包括没有什么实际意义的功能词和应用广泛无法保证结果的词汇词，如“不见”、“何处”等。所以，需要对关键词，根据停用词表进行过滤，得到更适用的关键词表。关键词的具体提取流程图如图3.8所示。



图3.7流畅度评价的流程图 图3.8关键词提取流程图

对诗歌的主题鲜明程度判断，是通过对诗歌主题进行预测，预测得到的各标签的概率值。由于一首诗歌的主题不一定唯一，结合采集数据的标签数量，决定以前二的概率值之和定为主题鲜明程度的判断依据。

为对诗歌主题进行预测，我们收集各主题诗歌数据，根据收集到的数据量进行筛选，最终标签定为8个，分别为爱国、悼亡、读书、山水、思念、思乡、送别以及田园。为预测诗歌主题标签，初步决定两种方法，方法一是神经网络预测模型，方法二是诗歌的关键词-频率字典方法。详细方法介绍如表3.2所示。

表3.2 预测诗歌主题的尝试方法介绍

|  |  |
| --- | --- |
| 方法 | 使用模型、步骤 |
| 神经网络模型 | CNN\_Model |
| KMax\_CNN\_Model |
| BiGRU\_Model |
| BiLSTM\_Model |
| 关键词-频率字典 | 根据关键词-频率字典，计算诗歌属于各主题标签的概率值，从而进行标签预测。 |

神经网络模型方法，主要通过kashgari框架来实现。关键词-频率字典方法，则是通过不同主题诗歌的关键词-频率字典，计算诗歌属于各主题标签的概率值，具体流程如图3.9所示。



（a）生成关键词-频率字典流程图 （b）判断主题鲜明度流程图

图3.9 根据关键词-频率字典判断诗歌主题鲜明度

其中，左图（a）是根据不同主题诗歌，生成关键词-频率字典的流程图，右图(b)是诗歌根据关键词-频率字典计算不同属于诗歌类别的概率值，从而判断诗歌主题鲜明度的流程图。

AHP算法的第一步是建立层次分析结构模型，然后构造比较矩阵，由于我们是运用AHP算法是为求得各个因素的影响权重值，所以下一步，也就是最后一步是进行层次单排序及一致性检验。计算矩阵的特征值和特征向量，取最大特征值及其对应的特征向量，并进行一致性检测。若能通过一致性检测，则该矩阵的最大特征值对应的特征向量，进行归一化后的结果，就是我们所求的各个因素的影响权重值，否则需重新构造比较矩阵，并进行重新计算，直到满足一致性要求。

如图3.10是AHP算法的程序流程图。



图3.10 AHP算法程序流程图

### 3.3.5 系统web服务模块

web界面主要靠html开发，展示系统的后端运用tornado框架。

对于前端的设计，使用了一些模板框架，如vue.js框架，一方面时更容易上手，另一方面也简化了设计的工作。设计的使用界面主要包括两个单选框，用于选择五言或七言，藏头或藏尾，一个输入文本框，用于输入藏头或藏尾词，一个提交按钮以及一个文本框，用于显示两首藏头诗内容。

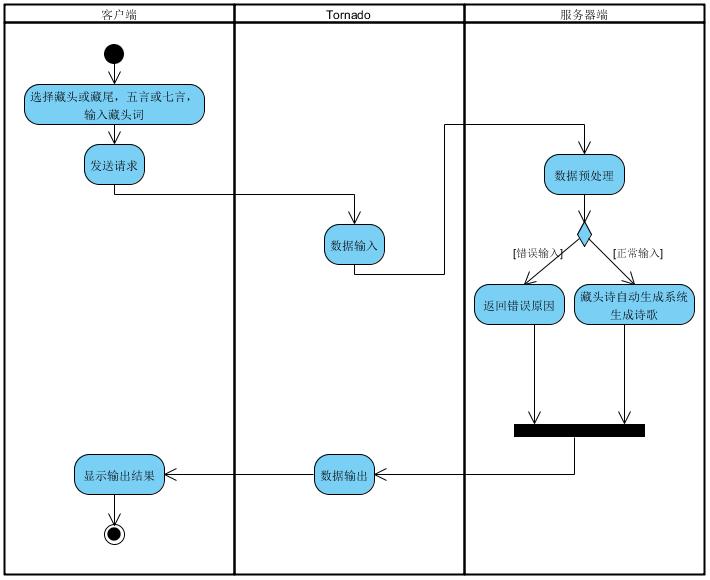
本系统选用tornado作为后端开发框架，通过post和get方法传递参数。下图3.11是web服务的活动图。

图3.11 web服务活动图

如图3.11所示，首先，用户在客户端进行藏头或藏尾、五言或七言的选择，并输入藏头或藏尾词，然后发送请求，数据输入后，到服务器端，对输入的数据，进行数据预处理，如果是错误输入，则返回错误原因，否则通过生成系统生成诗歌，并返回数据，最终在客户端显示输出结果。

## **3.4 本章小结**

本章在开头先介绍了本系统的设计原则，然后分别阐述了本系统的总体设计和模块设计。总体设计中，指出藏头诗自动生成系统包括诗歌数据集模块、诗歌训练模块、诗歌生成模块、诗歌自动评价模块以及系统web服务模块，描述了各个模块间的关系。然后在模块设计中，对各个模块的详细设计进行阐述。诗歌数据集模块中先采集大量诗歌数据，而后对数据进行清洗，得到目标数据集；诗歌训练模块，先定义输入和目标，划分训练集，而后搭建模型，进行训练；诗歌生成模块，运用训练好的模型，生成五言或七言的藏头或藏尾诗；诗歌自动评价模块，运用AHP算法从押韵、平仄、流畅度三个角度对古诗打分评价；系统web服务模块，简要介绍了系统展示界面的设计过程。

# **4 系统实现**

本章将阐述藏头诗自动生成系统的实现过程。先对系统开发环境进行说明，然后对系统中诗歌数据集、诗歌训练、诗歌生成、诗歌自动评价以及系统web服务这些模块的实现进行阐述。

## **4.1 系统开发环境**

藏头诗自动生成系统的开发主要是在Windows 10和CentOS 7.2操作系统上进行，开发过程中使用的集成环境有VS Code，使用的编辑工具为Notepad++，使用的Web框架为Tornado框架。详细信息如表4.1所示。

表4.1 系统开发软件需求表

|  |  |  |
| --- | --- | --- |
| 名称 | 版本 | 功能 |
| Windows | 10 | 操作系统 |
| Linux | CentOS 7.6 | 操作系统 |
| Visio Studio Code | 1.14.1 | 集成开发环境 |
| Notepad++ | 7.8.6 | 文本编辑器 |
| Python | 3.6 | 编程语言 |
| vue.js | 2.2.2 | Web框架 |
| tornado | 5.0.2 | Web框架 |
| tensorflow-gpu | 1.10.0 | 机器学习框架 |
| Tensorboard | 1.10.0 | 可视化工具 |
| CUDA | 9.0 | GPU加速 |
| Levenshtein | 0.12.0 | 字符串相似度计算工具 |
| jieba | 0.42.1 | 中文分词工具 |

自动生成模型训练资源占用量较大，因此模型训练均上传至服务器进行，其系统环境为CentOS 64位。训练模型的硬件资源配置如表4.2所示。

表4.2 系统开发硬件配置表

|  |  |
| --- | --- |
| 硬件资源 | 配置 |
| CPU | 6核 |
| 显存 | 12G |
| 硬盘空间 | 10G |
| GPU | GeForce GTX Titan X |

## **4.2** **诗歌数据集模块实现**

诗歌数据集的构建，是完成藏头诗自动生成系统的基础，需要注意的是数据量的大小以及数据质量。因此要尽可能多地获取诗歌数据，并进行清洗并去重，从而获得更高质量的数据集。

### 4.2.1 诗歌数据抓取

诗歌数据爬取部分的实现，主要采用requests库、Beautiful Soup库以及re库函数实现。主要函数及其功能如表4.3所列。

表4.3 诗歌数据抓取主要函数

|  |  |
| --- | --- |
| 函数 | 功能 |
| askURL(url) | 获取界面的全部内容 |
| getData(html) | 调用Beautiful Soup解析网页内容html，然后通过使用正则表达式，获得诗歌内容，保存到诗歌列表中 |
| saveData(datalist,savepath) | 将诗歌列表datalist中的内容保存到savepath路径中 |

在爬取网页数据部分spider.py中，主要工作是爬取目标网站上的诗歌数据，并保存。其中，askURL(url)函数的作用是获取界面的全部内容，参数url是目标网页的网址；在得到目标网页内容后，调用getData(html)，使用Beautiful Soup解析网页内容html，然后通过正则表达式匹配，获得诗歌内容，保存到诗歌列表中；再调用函数saveData(datalist,savepath)，将诗歌列表datalist中的内容保存到savepath路径中。网页的URL更新在主函数中。根据不同的网页html格式的网页形式不同，在使用正则表达式获取诗歌数据时，需要根据不同网站的规律进行调整。

### 4.2.2 诗歌数据清洗

诗歌数据清洗以及去重部分的实现，主要采用re库和Levenshtein库函数实现。清洗部分的主要函数及其功能如表4.4所列，数据去重合并部分则如表4.5所列。

表4.4 数据清洗主要函数

|  |  |
| --- | --- |
| 函数 | 功能 |
| check\_num\_of\_line(poem) | 去除诗句数不足4句的 |
| check\_num\_of\_words(poem) | 剔除诗句长度不一致，或不是五言或七言诗的 |

在数据清洗部分clean.py中，主要工作是对从网页爬取到的数据进行清洗，去除里面不符合诗歌要求的脏数据。首先，调用read\_file(openfile)，读取待处理的数据，然后利用正则表达式，替换和去除如“<br>”、“注：”、“①”等脏数据部分，并统一标点，再调用check\_num\_of\_line(poem)，去除诗句数不足4句的，之后调用check\_num\_of\_words(poem)，剔除诗句长度不一致，或不是五言或七言诗的，下一步，通过判断诗作标题中是否含有“生查子”、“木兰花”等词牌名，去除满足每句字数统一为五言或七言，但是词的作品，最后将清洗后的诗作进行保存。

表4.5 数据合并主要函数

|  |  |
| --- | --- |
| 函数 | 功能 |
| read\_file(openfile) | 读取待处理的数据 |
| to\_rate(list,line) | 返回当前行的诗歌数据与列表中数据的最大相似度 |

在合并数据集部分he.py中，主要工作是将收集的各个数据集内容合并起来，并通过字符串相似度，进行去重。首先，调用read\_file(openfile)，读取待处理的数据，然后，定义诗歌数据列表，再调用to\_rate(list,line)，返回当前行的诗歌数据与列表中数据的最大相似度，若相似度大于临界值，则剔除，否则，保存该行诗歌数据并添加进诗歌数据列表中，最终获得唐代至当代的五言、七言诗总计371,883首。

## **4.3 模型训练与诗歌生成实现**

模型训练模块包含了LSTM模型训练以及参数调节过程，训练的产出是诗歌生成模块和诗歌自动评价模块中流畅度分析的模型文件。诗歌生成模块，包含实现藏头或藏尾、五言或七言的选择，以及每次生成的诗歌等功能。

### 4.3.1 LSTM模型训练

根据诗歌训练模块的设计中input和target的定义，以及训练集和测试集的划分，得到了模型训练的原料——“数据”。接下来，搭建LSTM模型。如表4.6，列出了模型实现的主要类和主要函数。

表4.6 LSTM模型主要类及主要函数

|  |  |  |
| --- | --- | --- |
| 类名 | 主要函数 | 作用 |
| Param() | \_\_init\_\_() | 参数定义 |
| data() | prepare() | 构建字符-数字ID表 |
| train\_test\_data() | 生成训练集和测试集的诗歌向量集 |
| batch() | 按batch\_size，划分batch |
| load\_prepare() | 读取预处理好的文件 |
| Poetry\_Model() | \_\_init\_\_() | 定义模型结构 |
| load() | 载入已有的模型 |

如表中所示，Param()类中包含大量初始化参数，包括batch\_size，num\_layers，learning\_rate等。data()类的作用是将初始数据向量化，并划分batch，以投入训练的模型中。Poetry\_Model()类是定义模型及相关函数。

train.py中定义模型训练的具体过程。先调用data()类进行数据准备，生成字表，将诗歌转化为诗向量，并划分batch，以备进行分批训练。再调用Poetry\_Model()类，定义模型，若有已训练的模型，使用load()函数载入，继续进行训练，当loss稳定后，训练完成。

Param()类中有不少初始化参数，需经过大量实验调参，才能够达到更好的效果。主要调节的参数及含义如下表4.7所示。

表4.7 调整的主要参数及含义

|  |  |
| --- | --- |
| 参数 | 含义 |
| num\_layers | LSTM层数 |
| learning\_rate | 初始学习率 |
| batch\_size | batch的大小，即每个batch中有几组input和target |

当batch\_size=64，n\_epoch=15，rnn\_size=128，decay\_steps=1000，decay\_rate=0.97，learning\_rate=0.02时，调节num\_layers的过程如下图4.1所示。

图4.1 num\_layers与每轮训练时间关系

观察每轮增加时间折线，发现当num\_layers=8时，即图中标粗点，是图像中的极小点，表明，随着num\_layers的增加，每轮训练的时间持续增加，但增加的速度，先上升，后下降，再上升的趋势，当num\_layers=8时，每轮训练的时间增加量达到极小值点。并进一步结合训练代价的考虑，最终对参数num\_layers，选择num\_layers=8。

对其他参数的调节过程如图4.2。从图中可以看出，图中的极小值点有A、B两点，如图中所示。经对比分析后，发现当选择参数为B点，即参数batch\_size=64，rnn\_size=128，num\_layers=8， learning\_rate=0.04时，训练效果能够在可接受的训练代价中达到最佳。

B

A

图4.2 learning\_rate、batch\_size与loss的关系

### 4.3.2 诗歌生成实现

关于诗歌生成的实现，预期实现的功能包括：诗歌能够选择生成五言或七言，藏头或藏尾诗，且每次生成的诗歌不同。以藏头诗为例，其生成流程图如图4.3所示。



（a）藏头诗生成主程序流程图 （b）藏头句生成流程图

图4.3 藏头诗生成流程图

从上图左边的藏头诗生成主程序流程图中，藏头诗的生成，先经过藏头词的检查，判断为4字，不足则补充，不符合要求，则重新补充；然后由当前诗poem内容和当前藏头字，生成当前藏头句，一步步生成整首诗。

在生成当前藏头句的子程序流程图中，先不断生成下一字符，知道下一字符为标点，然后判断句长是否满足当前要求，满足则返回，否则重新生成该句。同时，为了实现每次生成的诗歌不同的功能，在生成下一字符时，引入随机数，随机在概率最大的100个可能字符中，选择其一。

这样生成的藏头诗，满足五言或七言的选择，以及每次生成的诗歌不同的要求。

## **4.4 诗歌自动评价实现**

诗歌自动评价的实现，主要包括通过AHP算法，得到各影响因素的权重值，对各影响因素，进行具体分析评价，从而利用AHP算法求得的权重向量，得到诗歌自动评价的结果。

### 4.4.1 AHP模型

首先，建立层次分析结构模型。目标层为评判古诗质量，准则层为在诗歌评价模块的设计中确定的影响因素，分别是平仄、押韵、流畅度、关键词和主题鲜明度，由于我们的目标是确定各个影响因素权重，因而不需要方案层。

接下来，构造成对比较矩阵。根据Santy1-9标度方法，构造出的比较矩阵如下式4.1所示。

|  |  |  |
| --- | --- | --- |
|  |  | （4.1） |

下一步，层次单排序及一致性检验。先对比较矩阵求出特征值，

为选择最大的特征值，这里实数特征根有，，，且对应特征向量为实数向量，含有虚部的复数无法比较大小。在三个实数特征向量中，我们选择最大特征值。

其对应的特征向量为：

再对该比较矩阵进行一致性检验，若能通过，则该特征向量就是所求的权值向量。

求得一致性指标，如式4.2所示。

|  |  |  |
| --- | --- | --- |
|  |  | （4.2） |

通过查表，我们得到当n=5时，RI=1.12。

则一致性比率，如式4.3所示。

|  |  |  |
| --- | --- | --- |
|  |  | （4.3） |

因此满足一致性检验。

最后，对权值向量进行归一化，得到最终的权值向量为

### 5.4.2 平仄、押韵评价实现

诗歌平仄、押韵的评价部分的实现，主要采用re库和pypinyin库函数实现。平仄、押韵评价主要函数及其功能如表4.8所列。

表4.8 平仄、押韵评价主要函数及其功能

|  |  |
| --- | --- |
| 函数 | 功能 |
| is\_rusheng(word) | 判断输入字是否为入声字 |
| ping\_or\_ze(word) | 判断输入字是平还是仄 |
| rhyme(word1,word2) | 判断字1和字2是否押韵 |
| score\_pingze() | 对输入诗在平仄方面进行打分 |
| score\_rhyme() | 对输入诗在押韵方面进行打分 |

在诗歌平仄、押韵部分score.py中，主要工作是对输入诗歌的平仄、押韵部分进行评价，并给出分数。首先调用score\_rhyme(),对输入诗在押韵方面进行打分，其中，2次调用了rhyme(word1,word2)，分别检验颔联和尾联的末尾字是否押韵，以及首联是否和颔联、尾联押韵，并根据结果，返回对平仄评价时比照的平仄规则选择。其中，押韵判断时，还需调用is\_rusheng()，来判断是否压入声韵。然后调用score\_pingze()，根据押韵判断时的返回值，进行判断，对不同字进行平还是仄判断时，调用ping\_or\_ze(word)函数。其中，需要注意的是入声字属于仄声。

### 4.4.3 流畅度评价实现

诗歌流畅度的评价部分的实现，主要采用tensorflow-gpu库函数和data()类，Poetry\_Model()类实现。

流畅度评价主要函数及其功能如表4.9所列。

表4.9 流畅度评价主要函数及其功能

|  |  |
| --- | --- |
| 函数 | 功能 |
| fluent() | Poetry\_Model()类中的函数，用来计算输入诗的生成概率 |
| score\_fluence() | 对生成概率，进行分级打分 |

在诗歌评价部分check\_quality.py中,对诗歌流畅度进行评价的函数flence()中，先调用Poetry\_Model()类中的fluent()函数，计算得到输入诗的生成概率。因为生成概率的数量级相差实在过大，为使尽量统一在较小范围内，经过多次实验，最终对生成概率开二次方根，为避免有生产率过大或过小未考虑的情况，再进行分级打分，使分数跨度和前面对平仄和押韵的评价保持一致。

### 4.4.4 关键词和主题鲜明度评价实现

对于诗歌关键词和主题鲜明度的评价，需要额外采集对应不同主题的诗歌数据。类似于诗歌数据集模块，由于没有成形的对应不同主题的诗歌数据集，网页上的分主题的诗歌量较少，爬取了多个网页，共获得8个主题的诗歌数据，共计8210首。

诗歌关键词评价部分的实现，主要采用jieba库中的analyse.textrank()和analyse.extract\_tags()函数实现。关键词评价主要函数及其功能如表4.10所列。

表4.10 关键词评价主要函数及其功能

|  |  |
| --- | --- |
| 函数 | 功能 |
| get\_keywords() | 对输入诗歌运用textrank算法和tf-idf算法分别提取关键词并取并集，过滤停用词。 |
| score\_keywords() | 对关键词集合，进行打分 |

对诗歌主题集中部分的实现，主要分为两部分：一是对诗歌主题标签进行预测，二是根据各主题标签的概率给出主题鲜明度评价。

对于诗歌主题标签预测部分，按照设计部分，尝试了两类方法，正确率比较情况如表4.11所示。

表4.11 诗歌主题标签预测正确率比较情况

|  |  |  |
| --- | --- | --- |
| 方法类别 | 模型或方法 | 正确率 |
| 神经网络模型 | CNN\_Model | 40.22% |
| KMax\_CNN\_Model | 41.32% |
| BiGRU\_Model | 39.34% |
| BiLSTM\_Model | 39.34% |
| 关键词-频率字典 | 取概率第一标签 | 53.92% |
| 取概率前二标签 | 73.45% |

从上表的结果比较中，不难看出，神经网络模型方法，可能囿于数据量的限制，多种模型尝试下，分类结果的正确率普遍在40%左右，无法满足对于分类任务的要求。而基于关键词-频率字典的方法，虽然朴素，但分类效果明显优于神经网络模型的方法，取概率第一的标签时，正确率为53.92%，当取概率前二的标签时，正确率达到73.45%，基本满足对于分类任务的要求。因此，对诗歌主题集中部分的实现，决定使用基于关键词-频率字典的方法。

诗歌主题鲜明度评价部分的实现，主要采用诗歌关键词评价部分的get\_keywords()函数以及operator库函数。主题鲜明度评价主要函数及其功能如表4.12所列。

表4.12 主题鲜明度评价主要函数及其功能

|  |  |
| --- | --- |
| 函数 | 功能 |
| topic\_dict() | 将已保存的关键词-频率字典导入，留待使用。 |
| poem\_topic () | 对输入的诗歌运用get\_keywords()函数，提取关键词，并通过关键词-频率字典，计算得到概率前二的标签和概率值。 |
| score\_topic() | 根据前二的概率值，对主题鲜明度进行打分。 |

在诗歌评价部分check\_quality.py中,对诗歌主题鲜明度进行评价的函数score\_topic()中，先调用topic\_dict()函数，将data/label文件夹中保存的不同主题的关键词-频率字典导入，保存为字典形式，留待使用。然后调用poem\_topic ()函数，对输入的诗歌，运用textrank算法和tf-idf算法分别提取关键词，取并集，过滤停用词，然后对比不同主题的关键词-频率字典,得到概率值前二的大小。最后根据得到的前二的概率值，对主题鲜明度进行打分。

## **4.5 Web展示系统实现**

展示系统的前端主要靠html开发，运用vue.js框架，而展示系统的后端则运用tornado框架。

### 4.5.1 前端实现

对于前端的设计，使用了一些模板框架，如：vue.js框架，一方面时更容易上手，另一方面也简化了设计的工作。使用界面主要包括两个单选框，一个输入文本框，一个提交按钮以及一个文本框显示两首藏头诗内容。藏头诗自动创作系统的界面实现如图4.4所示。



图4.4 藏头诗自动创作系统界面

### 4.5.2 后端实现

本系统选用tornado作为后端开发框架，通过post和get方法传递参数。

在数据处理部分，通过handle\_poem函数对输入的参数进行合法性判断，主要判断关键词长度以及关键词是否符合规范。由于模型的输出是以字符串的形式，需要对其进行分句，从而便于显示。

后端实现的主要函数如表4.13所示。

表4.13 后端实现主要函数及其功能

|  |  |
| --- | --- |
| 函数 | 功能 |
| pinyin\_2\_hanzi(pinyinList) | 对输入参数中的拼音，给出十个分数最高汉字 |
| handle\_poem() | 对输入数据进行预处理，并生成诗歌，或返回相应错误 |

pinyin\_2\_hanzi(pinyinList)用在对字表中不存在的字，对其进行同音替换。其中handle\_poem()对输入数据进行预处理，对错误输入返回错误提示，经处理后得正确输入，生成诗歌，并输出。其处理过程的流程图如下图4.5所示。



图4.5 输入合法性判断及处理流程图

上图中，key\_words表示输入的藏头词或藏尾词，错误1表示输入的key\_words有效长度为0，错误2指输入的key\_words中含有非中文字符，均无法生成诗歌。

## **4.6 本章小结**

本章首先介绍了系统的开发环境，然后对系统设计部分的各模块内容进行阐述。对于诗歌数据集模块，介绍了诗歌数据的抓取以及数据清洗工作，对模型训练与诗歌生成部分，阐述了LSTM模型，利用训练好的模型，生成要求的诗作，并对比了尝试的其他模型方法；诗歌评价模块，从AHP模型出发，根据平仄、押韵、流畅度的影响因素，构造诗歌自动评价机制；Web展示系统实现则从前后端分别介绍了展示系统的搭建过程。

# **5 系统测试**

## **5.1 生成诗歌效果测试**

为测试藏头诗生成效果，由于目前没有公认的较好的自动化测评方式。在很多相关研究中，都是通过人工测评的方式对生成诗歌进行评价，且主要从押韵程度、流畅性、主题相关性以及内容意义四个方面展开。

本测试方案将由本系统生成10首藏头诗，10首藏尾诗，从以上四个角度来进行打分，每一项的满分是十分。

另外，为了比较生成结果，另设置两个对照组，分别是本系统、清华九歌诗歌生成系统(<http://jiuge.thunlp.org/>)以及在线藏头诗系统（<http://www.227g.com/shi/>）。

针对诗歌生成效果人工评测的详细方案如下表5.1所示。

表5.1 诗歌生成效果的详细测试方案

|  |  |
| --- | --- |
| 实验目的 | 测试模型生成诗句的效果 |
| 实验数据 | 10首藏头诗，5首五言，5首七言，藏头词字数四字：三字：两字=2:2:1 |
| 10首藏尾诗，5首五言，5首七言，藏尾词字数为四字 |
| 实验对象 | 本系统、清华九歌诗词生成系统、在线藏头诗系统 |
| 评价方面 | 藏头诗从以下四项打分：押韵程度、流畅程度、主题相关度、内容是否有意义 |
| 藏尾诗则从流畅程度、主题相关度、内容是否有意义三项打分。 |
| 评分标准 | 将生成的诗歌打乱，让20个人对生成的诗歌进行评判打分，最终取平均值。每一项的分数是 10 分，最后计算每一项总得分的平均分。其中关于押韵的评分标准为：假如不押韵就是0分，一对押韵为5分，两对押韵为10分，其它三个指标以5分作为一个平均值，若效果差，则5分以下，若效果较好，则在5分基础上酌情加分。 |

由于在线藏头诗系统生成句数由藏头词字数决定，所以为保证测试的公平性，对该系统只进行四字生成测试。

所以藏头诗的实验对象为本系统和九歌系统；由于九歌没有藏头诗生成的功能，所以藏尾诗的实验对象为本系统和在线藏头诗系统。

因为用户可能会输入一些现代词，所以在测试用例里，平均的设置了几组古代词和现代词，进行实验。

藏头诗、藏尾诗具体生成效果见附录所示。

### 5.1.1 藏头诗生成测试

藏头诗生成评分，对藏头诗测试的样例进行人工打分，得分情况如表5.2所示。

表5.2 本系统与清华九歌系统生成藏头诗打分表

|  |  |  |
| --- | --- | --- |
| 指标 | 本系统 | 清华九歌 |
| 押韵程度 | 7.0 | 6.5 |
| 流畅程度 | 7.3 | 7.5 |
| 主题相关度 | 6.8 | 6.6 |
| 内容是否有意义 | 6.5 | 6.7 |

从表5.2中，可以发现，人工评测的结果中，从押韵程度、主题相关度两方面，本系统略优于清华九歌系统，而从流畅程度以及内容是否有意义来看，则略显不足。

总体来看，本系统与清华九歌系统在藏头诗生成方面的生成效果，在具体各方面上，各有优缺，但总体上，生成效果相近。

### 5.1.2 藏尾诗生成测试

藏尾诗生成评分，对藏尾诗测试的样例进行人工打分，得分情况如表5.3。

表5.3 本系统与在线藏头诗系统生成藏尾诗打分表

|  |  |  |
| --- | --- | --- |
| 指标 | 本系统 | 在线藏头诗系统 |
| 流畅程度 | 4.5 | 3.8 |
| 主题相关度 | 5.2 | 5.4 |
| 内容是否有意义 | 6.5 | 6.2 |

从表5.3中，可以发现，对藏尾诗的人工评测结果中，押韵程度由于藏尾词统一，未进行评价外，流畅程度方面，本系统优于在线藏头诗系统，而在主题相关度和内容是否有意义两方面，则略有不足。

从总体来看，本系统与在线藏头诗系统在藏尾诗的生成效果相比，略胜一筹。

## **5.2 诗歌自动评价测试**

为检验诗歌自动评价系统的有效性，测试方案为对人工评价的数据进行自动评价，并比较人工评测和自动评测的得分情况。由于人工评价按各方面指标给分，因此在自动评价中，将各项分数列出，以供比较。

其中，内容是否有意义与内容丰富度相比较，主题相关度与主题集中度，即主题鲜明度相比较。

首先，对诗歌数据进行自动评价，对藏头诗的评价结果如表5.4所示。藏头诗的人工与自动评价对比如表5.5所示。

表5.4 藏头诗自动评价结果表

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 系统评价 | | 诗1 | 诗2 | 诗3 | 诗4 | 诗5 | 诗6 | 诗7 | 诗8 | 诗9 | 诗10 | 均分 |
| 本系统 | 押韵 | 10.0 | 8.0 | 8.0 | 10.0 | 10.0 | 10.0 | 8.0 | 8.0 | 10.0 | 10.0 | 9.2 |
| 平仄 | 9.6 | 8.4 | 9.2 | 9.1 | 8.8 | 8.8 | 9.6 | 8.8 | 7.1 | 8.4 | 8.8 |
| 流畅度 | 10.0 | 10.0 | 10.0 | 9.6 | 10.0 | 10.0 | 10.0 | 10.0 | 8.2 | 10.0 | 9.8 |
| 丰富度 | 7.0 | 9.0 | 10.0 | 8.0 | 6.0 | 10.0 | 9.0 | 8.0 | 8.0 | 10.0 | 8.5 |
| 主题 | 4.3 | 4.5 | 5.0 | 4.7 | 3.5 | 3.6 | 4.5 | 4.1 | 5.3 | 3.0 | 3.8 |
| 清华九歌 | 押韵 | 8.0 | 8.0 | 8.0 | 10.0 | 10.0 | 8.0 | 8.0 | 10.0 | 8.0 | 10.0 | 8.8 |
| 平仄 | 10.0 | 9.1 | 8.8 | 9.4 | 8.3 | 8.8 | 9.2 | 8.8 | 9.6 | 8.9 | 9.1 |
| 流畅度 | 10.0 | 10.0 | 10.0 | 10.0 | 10.0 | 10.0 | 10.0 | 10.0 | 10.0 | 10.0 | 10.0 |
| 丰富度 | 8.0 | 10.0 | 8.0 | 10.0 | 7.0 | 10.0 | 8.0 | 10.0 | 8.0 | 10.0 | 8.9 |
| 主题 | 3.4 | 3.9 | 3.2 | 4.2 | 3.8 | 4.5 | 4.2 | 3.3 | 3.5 | 3.4 | 3.7 |

表5.5 藏头诗的人工与自动评价对比表

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 评价方式 | 本系统 | | | | | 清华九歌 | | | | |
| 押韵 | 平仄 | 流畅度 | 内容 | 主题 | 押韵 | 平仄 | 流畅度 | 内容 | 主题 |
| 人工评价 | 7.0 | 无 | 7.3 | 6.5 | 6.8 | 6.5 | 无 | 7.5 | 6.7 | 6.6 |
| 自动评价 | 9.2 | 8.8 | 9.8 | 8.5 | 3.8 | 8.8 | 9.1 | 10.0 | 8.9 | 3.7 |
| 自动评价综合得分 | 8.8 | | | | | 8.9 | | | | |

结合表5.4和表5.5，可以发现，人工评价与自动评价的给分方式不同，具体分值有所不同，但是分数的相对情况则保持一致。人工评价中，清华九歌生成诗的流畅程度、内容丰富度略优于本系统，自动评价中的结果也是类似，该项得分上，清华九歌优于本系统。同样，在押韵程度和主题鲜明度方面，比较人工评价和自动评价的结果，发现分数情况类似，本系统略优于清华九歌系统。从综合评价的给分结果，本系统与清华九歌的生成效果相近。

从各方面评价，以及综合评价中，自动评价与人工评价给出的结果相似。

接下来，对生成的藏尾诗进行自动评价，由于藏尾诗的尾已定，讨论押韵，没有意义，因而省去该项，评价结果如下表5.6所示。

表5.6 藏尾诗自动评价结果表

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 系统评价 | 类别 | 诗1 | 诗2 | 诗3 | 诗4 | 诗5 | 诗6 | 诗7 | 诗8 | 诗9 | 诗10 | 均分 |
| 本系统 | 平仄 | 7.9 | 8.4 | 8.8 | 8.4 | 8.3 | 9.4 | 8.3 | 9.4 | 8.8 | 8.8 | 8.7 |
| 流畅度 | 5.4 | 10.0 | 2.1 | 1.0 | 1.0 | 10.0 | 1.0 | 1.0 | 1.0 | 1.0 | 3.4 |
| 关键词 | 8.0 | 10.0 | 8.0 | 10.0 | 3.0 | 10.0 | 7.0 | 10.0 | 5.0 | 8.0 | 7.9 |
| 主题 | 4.5 | 4.1 | 3.3 | 6.8 | 10.0 | 3.8 | 0 | 5.5 | 7.6 | 4.5 | 5.0 |
| 在线藏头诗系统 | 平仄 | 8.3 | 8.1 | 8.3 | 8.1 | 7.9 | 8.1 | 7.5 | 8.8 | 7.5 | 8.4 | 8.1 |
| 流畅度 | 1.0 | 1.0 | 10.0 | 1.0 | 1.0 | 3.2 | 1.0 | 1.0 | 1.0 | 1.0 | 2.1 |
| 关键词 | 7.0 | 10.0 | 8.0 | 10.0 | 9.0 | 10.0 | 8.0 | 9.0 | 8.0 | 10.0 | 8.9 |
| 主题 | 3.8 | 5.5 | 3.4 | 5.5 | 10.0 | 3.7 | 5.4 | 4.0 | 10.0 | 4.1 | 5.5 |

藏头诗的人工与自动评价对比如表5.7所示。

表5.7 藏尾诗的人工与自动评价对比表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 评价方式 | 本系统 | | | | 在线藏头诗系统 | | | |
| 平仄 | 流畅度 | 内容 | 主题 | 平仄 | 流畅度 | 内容 | 主题 |
| 人工评价 | 无 | 4.5 | 6.5 | 5.2 | 无 | 3.8 | 6.2 | 5.4 |
| 自动评价 | 8.7 | 3.4 | 7.9 | 5.0 | 8.1 | 2.1 | 8.9 | 5.5 |
| 自动评价综合得分 | 3.5 | | | | 3.2 | | | |

结合以上两表，可以发现，在生成的藏尾诗流畅度评价中，人工评价和自动评价的结果相似，本系统略优于在线藏头诗系统；而在内容丰富度以及主题鲜明度中，本系统得分略低于在线藏头诗系统，也与人工评价中结果类似。

综合评价的给分结果，本系统略优于在线藏头诗系统的生成效果。其中，因为藏尾诗藏尾字是限定的，无法实现押韵，该项得分为0，所以藏尾诗的自动评价综合得分明显低于表5.5中藏头诗的自动评价得分。

在藏头诗与藏尾诗的人工评价和自动评价比较中，虽然给分方式不同，具体分数有所差异，但分数的相对情况相同。在本次实验结果中，本古诗自动评价系统在押韵程度、平仄和流畅度方面以，进行评价的结果与人工评价相近；自动综合评价结果也与人工评价相近。

## **5.3 Web端测试**

对于本系统而言，对于web端，主要关注于能否为用户提供稳定可靠且实时有效的藏头诗生成服务，因此测试时，主要考虑以下三个方面：

(1) 对非法输入的处理，

(2) 响应时长

(3) 系统兼容性

对于藏头诗和藏尾诗非法输入检查逻辑是一致的，所以就以藏头诗为例，对系统鲁棒性进行测试，测试结果如表5.8所示。

表5.8 系统鲁棒性测试

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 输入 | 输出 | 是否正常 |
| 1 | 1-4字中文 | 两首诗歌 | 是 |
| 2 | 超过4字 | 截前四字作得两首诗 | 是 |
| 3 | 空格 | 提示藏头词在1-4字 | 是 |
| 4 | 字符串含有空格 | 两首诗歌 | 是 |
| 5 | 字符串含字母或其他非中文符号 | 提示藏头词为中文 | 是 |
| 6 | 字符串含有未出现过的字 | 读音相近字代替，得到两首诗 | 是 |

在表5.8中，对于输入进行合法性检测，对各种非法输入进行相应处理，结果输出均正常，这保证了系统的鲁棒性。

对实时性的测试如表5.9所示。给出8组测试输入，对不同诗歌类型、不同长度输入词的相应时间进行统计。

表5.9 系统实时性测试

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 输入 | 类型 | 响应时间（s） |
| 1 | 春 | 五言藏头诗 | 5.27 |
| 2 | 夏 | 七言藏头诗 | 5.84 |
| 3 | 春分 | 五言藏尾诗 | 4.99 |
| 4 | 谷雨 | 七言藏尾诗 | 5.99 |
| 5 | 芙蓉虾 | 五言藏头诗 | 4.81 |
| 6 | 红烧肉 | 七言藏头诗 | 5.72 |
| 7 | 四喜丸子 | 五言藏尾诗 | 4.92 |
| 8 | 龙井虾仁 | 七言藏尾诗 | 5.32 |

在表5.9中，对于各种不同输入请求，需要的相应时间，均在6秒之内，满足对系统实时性要求。

对兼容性来说，测试对象为主流浏览器，这里测试IE和chrome浏览器，对显示、相应以及服务方面进行测试。测试结果如表5.10所示。

表5.10 系统兼容性测试

|  |  |  |  |
| --- | --- | --- | --- |
| 浏览器 | 显示 | 响应 | 服务 |
| IE | 正常 | 正常 | 正常 |
| chrome | 正常 | 正常 | 正常 |

在表5.10中，测试了系统在各主流浏览器的使用，均显示、相应、服务正常，这满足系统兼容性要求。

## **5.4 测试结论**

在诗歌生成效果测试中，本系统的藏头诗生成效果与清华九歌系统相近，藏尾诗的生成效果则由于在线藏头诗生成系统；在诗歌自动评价测试中，本系统的自动评价结果与人工评价结果在各个细节方面以及总体综合评价方面，评价结果相近，这证明本系统的自动评价机制的有效性；在系统web端测试中，本系统的鲁棒性、实时性以及兼容性均表现良好，满足要求。

本系统在诗歌生成，古诗自动评价，以及web端，均通过测试，且表现不错。

## **5.5 本章小结**

本章分别对诗歌生成效果，包括藏头诗生成和藏尾诗生成，诗歌自动评价以及系统web端进行测试。在诗歌生成效果测试时，采用人工打分方法，从押韵程度、流畅程度、主题相关度、内容是否有意义四个方面进行测试，对比本系统与其他诗歌生成系统效果；在诗歌自动评价测试中，运用本系统的自动评价机制对人工评价的作品进行打分，并从各个细节方面以及总体综合评价方面，对比人工评价结果；在系统web端测试中，对系统的鲁棒性、实时性以及兼容性进行测试。

# **6 总结与感想**

## **6.1 总结**

通过近半年的学习，藏头诗自动生成系统的设计与与实现已经顺利完成，不仅完成了任务书中的要求，而且在系统测试的结果也较为成功，令人满意，尤其是在对诗歌自动评价机制中进行了尝试，测试的效果也符合预期目标。本系统的重点在于诗歌数据集的搭建，藏头诗自动生成系统的构建以及诗歌自动评价系统的实现。对这段时间的学习和工作总结如下：

(1) 在工作前期，收集国内外相关研究现状，并进行分析，了解本课题的研究难点和重点，并发现缺乏诗歌自动评价系统，为接下去的研究进展作铺垫。

(2) 本系统从互联网上获取诗歌资源，为提高诗歌数据集质量，利用正则表达式进行数据清洗，通过Levenshtein Distance计算字符串相似度从而进行去重，得到高质量的唐代至当代五言、七言诗数据集，共计371,883首。

(3) 本系统共训练了两个基于神经网络的生成系统：藏头诗生成系统和藏尾诗生成系统，设计并实现了藏头诗自动生成系统。

(4) 为满足人们对于藏头诗的生成需求，本系统实现了藏头诗以及藏尾诗的生成，五言诗以及七言诗的选择，且在界面中，对同一输入，生成不同诗作，以供用户选择，满足人们对于个性化的追求。

(5) 在对生成的诗歌进行评价中，利用AHP算法，从押韵、平仄、流畅度、内容丰富度、主题鲜明度五个方面，实现古诗自动评价系统，填补了诗歌自动评价系统的空白，对诗歌自动评价系统的实现做出尝试，也为后来的研究者提供了一些解决思路。

(6) 本系统以web界面的形式进行展示，利用Tornado后端框架和vue.min.js前端框架进行搭建，展示了藏头诗和藏尾诗自动生成的成果。

本文完成了藏头诗的自动生成，这一核心功能，并添加了藏尾诗的自动生成，在对诗歌评价的过程中，进一步设计完成了古诗自动评价系统。总结全文，本文还有一些可以进一步研究的工作：

(1) 古诗数据集。在搜集诗歌过程中，对大量的残句，残诗等数据，直接剔除，没有进行利用，可以尝试如何利用这一部分数据，做到“变废为宝”；经清洗、去重后的数据集共有371883首唐代至当代的五言、七言诗，虽然在本系统实现中，已经尽可能搜集整理，但由于古时抄录、刊印错误等，有些诗句存在多可能性。之后，还需要更进一步搜集相关资料，提高数据集的准确度。

(2) 对古诗词进行分词。本系统是基于字进行训练，生成效果总体符合预期。诗歌作品中，有些字词的组合有典故或特殊含义，若分开学习，则可能失去整体含义。若能对古诗词等进行分词，生成效果或有较大提升。但现有的汉语分词工具，如jieba，对于古诗词的分词效果不佳，无法进行直接使用。

(3) 古诗自动评价方式。本系统是利用AHP算法，从押韵、平仄、流畅度三个方面实现。由于没有可利用的古诗打分数据，使用AHP算法，构造比较矩阵，可以尝试其他算法。还可以进一步增加评价的角度，从而更全面地对古诗进行评价。

(4) 本文最终使用的是LSTM系统，生成诗歌的效果符合预期，但在训练过程中需要预设较多参数，参数调节的好坏，对诗歌生成模型有着较大的影响。因此，预设的参数，事先需要进行大量对照实验，进行对比测试，从而确定。

## **6.2 感想**

在这半个学期自然语言与机器翻译课程的学习中，我收获了很多，对机器翻译的各阶段历程有了更系统的认识，对一些方法模型也有了进一步的理解。

感谢肖老师对机器翻译的介绍，给我们讲解了机器翻译经历的数个阶段，对统计时代的机器翻译进行了说明，尤其是对IBM模型的叙述，让我们这些看过资料依然困惑的人，终于能够“拨云见日”；对神经元原理的阐述，尤其是“女朋友”例子，让我们知道了高端的知识往往可以用朴素的方式进行解释；对transformer模型的介绍，尤其是K,Q,V的生动描述，让我们能够更进一步理解attention机制。

感谢各位学长对上机实验细致的介绍与无微不至的帮助，让我们能够无障碍地体验在NiuTrans框架下使用统计机器翻译与FNN翻译模型。

老师的课程安排得很棒，由历史切入，对几个时代的机器翻译进行介绍，其中对统计机器翻译以及神经机器翻译模型进行了较详细的介绍，对热门的transformer模型也进行了说明，让我们对机器翻译既有了整体了解，又对其中的重点内容有了进一步的认识。实验的安排很合理，从基础的实验平台构建，到每一步的实现以及作用，从统计机器翻译的实现，到神经机器翻译FNN的实现，从ppt讲解时的细致认真，到上机实验时的耐心讲解，让我们这些入门菜鸟，能够有一个几乎无bug的实验体验，为后面自己项目的进行提供了一定的经验。

附录A 诗歌生成结果

**表A1 藏头诗生成结果**

|  |  |  |  |
| --- | --- | --- | --- |
| 藏头词 | 类型 | 本系统 | 清华九歌 |
| 春风化雨 | 五言 | 春事已求甚，  风霜始在人。  化文黄屋士，  雨后鬓纷纷。 | 春树绿阴深，  风尘自古今。  化工成底事，  雨雪满山林。 |
| 青梅竹马 | 七言 | 青苞怪己应无尽，  梅影初宜自惆怅。  竹堂楼上不能来，  马南何处觅鸳鸯。 | 青春风暖柳条新，  梅雨初晴草色匀。  竹里一枝聊寄傲，  马头三月未归人。 |
| 明天下雨 | 五言 | 明主当观日，  天台隐昔修。  下名虽欲别，  雨意在荒丘。 | 明朝又月华，  天地自生涯。  下界无人到，  雨中有一家。 |
| 东北大学 | 七言 | 东风动夜归秋晚，  北郭仙船定有天。  大物已无多死有，  学来难免翳须肝。 | 东风南海送归舟，  北望天涯忆旧游。  大地山河千万里，  学仙何处觅瀛洲。 |
| 花想容 | 五言 | 花落风枝冷，  想知迁砌凉。  容光遂不浅，  此梦未能央。 | 花落小桃红，  想君笑此翁。  容身无一事，  颜色自春风。 |
| 阳关调 | 七言 | 阳台相对无殊思，  关下悠哉且不归。  调策不缘邀老去，  初经觅句觉高碑。 | 阳月笛声何处是，  关山迢递隔江津。  调和春色无人管，  曲折蘼芜满路尘。 |
| 佛跳墙 | 五言 | 佛家应有地，  跳迹自无田。  墙外鱼鱼暗，  江流柳径眠。 | 佛阁焚香坐，  跳珠出建章。  墙阴新雨露，  下马度潇湘。 |
| 八宝饭 | 七言 | 八月十头都是梦，  宝书上共发飘然。  饭馀散帙诗如此，  一里春风又紫兰。 | 八年十载一经过，  宝刹依然在薜萝。  饭颗山前真乐事，  牛羊何处问渔蓑。 |
| 明月 | 五言 | 明镜露满衣，  月华能如此。  幽梦坐忘机，  流光一千里。 | 明年清光好，  月色满池塘。  照影随流水，  人间有底忙。 |

**续表A1 藏头诗生成结果**

|  |  |  |  |
| --- | --- | --- | --- |
| 藏头词 | 类型 | 本系统 | 清华九歌 |
| 孤烟 | 七言 | 孤负深情无近时，  烟霞摇韵总依然。  犹堪昨日萤棂过，  已识旧功初一番。 | 孤云漠漠水悠悠，  烟雨濛濛一叶舟。  起看江山千万叠，  暮年心事付沧洲。 |

**表A2 藏尾诗生成结果**

|  |  |  |  |
| --- | --- | --- | --- |
| 藏尾词 | 类型 | 本系统 | 在线藏头诗系统 |
| 春风化雨 | 五言 | 兰根一月春，  秋色凌霜风。  我生多造化，  晓凉空尽雨。 | 庄苑丽宜春，  诗酒会春风。  自是无为化，  云行疑带雨。 |
| 青梅竹马 | 七言 | 梦来山色水生青，  又好归时有落梅。  知君忽见荒山竹，  万里青山避甲马。 | 数株桃树药囊青，  细缬全披画阁梅。  会须上番看成竹，  不待诏书行上马。 |
| 明天下雨 | 五言 | 去年已一明，  春风亘四天。  岩禽隔林下，  乃复赛风雨。 | 长门夜月明，  碧嶂插遥天。  昨日上山下，  江前飞暮雨。 |
| 东北大学 | 七言 | 忆我如何老岭东，  百咽从恩指江北。  馀星齐续怨春大，  远击不循严翰学。 | 新妆袨服照江东，  狂夫犹戍交河北。  仰看明星当空大，  山公倒载无妨学。 |
| 诗礼银杏 | 五言 | 危哭士氏诗，  与北酒兴礼。  生物相餍银，  羹筇不如杏。 | 香闻郑国诗，  风行未备礼。  风高浪泼银，  重帷照文杏。 |
| 年年有余 | 七言 | 结乐涪翁无几年，  如今衰后认当年。  身中万事无时有，  待尽凭栏细雨余。 | 水冻草枯为一年，  出入关山十二年。  富贵婚姻古无有，  水洊源流万顷余。 |
| 灯影牛肉 | 五言 | 一楚绿生灯，  九宇宫坡影。  巳尔炊螟牛，  于醅自肌肉。 | 思君如孤灯，  空山朱戟影。  相思望斗牛，  奉公举骨肉。 |
| 鱼香肉丝 | 七言 | 柑子春升引鲤鱼，  谈山培赐露新香。  诸公不尽庭花肉，  待月新园冒晓丝。 | 绿藻还疑宴镐鱼，  至今三载犹闻香。  雠家白官先请肉，  日暮向风牵短丝。 |

**续表A2 藏尾诗生成结果**

|  |  |  |  |
| --- | --- | --- | --- |
| 藏尾词 | 类型 | 本系统 | 在线藏头诗系统 |
| 龙井虾仁 | 五言 | 匪伊摄元龙，  兼之楚得井。  刘侯本匪虾，  着拙继王仁。 | 溥德翊飞龙，  严气消冰井。  虫怜目待虾，  良牧怀深仁。 |
| 莲花血鸭 | 七言 | 指姊园中见墨莲，  可怜儿女自愁花。  却恐移我桃欲血，  着臂投低只双鸭。 | 峰开华岳耸疑莲，  何如江上采莲花。  蛟龙啮尸鱼食血，  新教内人供射鸭。 |