自然语言处理报告

课程：自然语言处理
姓名：杨程锦
学号：2021214710
班级：计科 21-1 班
日期：2023.12.8

实验一、语料库的收集与整理

一、 研究背景

语料库是一种大规模电子文本库，经过科学取样和加工，其中包含在实际语言使用中真实出现的语言材料。它不仅是语料库语言学研究的基础资源，也是支持经验主义语言研究方法的主要依托。语料库的应用涵盖词典编纂、语言教学、传统语言研究以及基于统计或实例的自然语言处理等多个领域。根据研究目的和用途，语料库可分为多种类型，这一分类通常在语料采集的原则和方式上得以反映。

有人将语料库分为四种类型：⑴异质的（Heterogeneous），即广泛收集并原样存储各种语料，没有特定的语料收集原则；⑵同质的（Homogeneous），只收集同一类内容的语料；⑶系统的（Systematic），根据预先确定的原则和比例收集语料，使其具有平衡性和系统性，能够代表某一范围内的语言事实；⑷专用的（Specialized），只收集用于特定用途的语料。

此外，根据语料的语种，语料库还可分为单语的（Monolingual）、双语的（Bilingual）和多语的（Multilingual）。按照语料的采集单位，语料库可分为语篇、语句和短语。双语和多语语料库按照语料的组织形式，可分为平行（对齐）语料库和比较语料库。前者的语料构成译文关系，主要用于机器翻译、双语词典编纂等应用领域，而后者则将表述同一内容的不同语言文本收集到一起，多用于语言对比研究。

语料库具有三个特征：1.存放的是在语言的实际使用中真实出现过的语言材料，因此例句库通常不应被算作语料库； 2.是承载语言知识的基础资源，但并非等同于语言知识；3.真实语料需要经过加工（分析和处理）才能成为有用的资源。

二、模型方法

本次实验使用了两个中文语料库，分别是宋词语料库和人民日报语料库。鉴于它们的主体均为中文，我们选择采用了 n-gram 模型，具体而言，是使用了二元的 Bi-Gram 模型。这一模型是基于统计语言模型的算法，其基本思想是通过按字节进行大小为 N 的滑动窗口操作，形成长度为 N 的字节片段序列，即 gram。每个 gram 都代表一个特定的词组合，而对所有 gram 的出现频度进行统计后，通过设定的阈值进行过滤，形成关键 gram 列表，构建文本的向量特征空间。在这个列表中，每种 gram 即成为一个特征向量维度。

该模型建立在一个假设基础上，即第 N 个词的出现只与前面 N-1 个词相关，而与其它任何词都不相关。整句的概率可以被表示为各个词出现概率的乘积。为了实现这一假设，我们选择了二元的 Bi-Gram 模型，即每个词依赖于其前一个词，符合 first order 的马尔科夫假设。

$$\begin{aligned} p(s) &= p(w_1)p(w_2|w_1)p(w_3|w_2)...p(w_n|w_{n-1}) \\ &= p(w_1)\prod_{i=2}^{n} p(w_i|w_{i-1}) \end{aligned}$$

在统计阶段，我们只需要考虑单字词以及相邻的两个单字词所组成的双字词（即 2-gram 所需的词）。这样，我们能够明确代码实现的思路。

三、系统设计

1.打开开发环境，根据自己熟悉的语言，确定开发环境。本次实验我所采用的开发环境是 IDEA 2021.3.1，编程语言是 JAVA1.8。

2.下载语料库（ci.txt 和新闻语料库）到特定目录下。

3.根据文本编码加载语料库文本。宋词文本的格式遵循以下规则：

（1） 每首词以一行词牌名开头，接着一行正文。

（2）有些词在开始前会有单独一行的作者名。

（3）每首词的每一句间用中文标点符号“，”、“。”、“、”隔开。

（4）部分词中含有虚缺号。

而新闻文本则以以下方式呈现：

（1）每句新闻以时间开头。

（2）每句话被分为若干个单词，每个词及其词性由一个“/”分隔。

（3）每个词中可能包含空格或其他非汉字符号。

在加载文本时，所采用的编码是 gb18030。这是因为在处理过程中发现 utf-8 在读取一些较为生僻的汉字时会出现错误（illegal multibyte）。通过查询发现，gb18030 能够有效解决这一问题。

4.为了分别统计 n-gram（n=1，2）的词频并存储到相应的数据结构，我们可以按照以下步骤进行处理：

宋词处理环节：

A. 打开宋词语料库文本，按照规律进行分割：
   - 每首词的第一行为词牌名或作者名，若分割后长度为 1，则跳过。
   - 对于分出的词的正文，去除非汉字部分，提取纯汉字内容。

B. 统计单字词频：
   - 逐行读取提取完毕的宋词正文。
   - 对每个单字进行频率统计，使用词典存储，键为单字词，值为频率。

C. 统计双字词频：
   - 将每一行的字符列表化。
   - 遍历字符列表，对相邻两个字符组成的双字词进行频率统计，使用词典存储。

新闻处理环节：

A. 打开人民日报语料库，按照规律进行分割：
   - 每一句新闻以时间开头，去除时间和非汉字部分，提取含有汉字的词及其词性。

B. 统计单字词频：
   - 逐行读取提取过单词的文档。
   - 对每个单字进行频率统计，使用词典存储，键为单字词，值为频率。

C. 统计双字词频：
   - 使用一个临时列表保存所有单词。
   - 设置一个计数器 count，每当 count=2 时，合成两个单字词为一个双字词，并进行频率统计，使用词典存储。

通过以上步骤，你可以得到分别统计了单字和双字词频的词典，其中包括词和频度的信息。

5.将数据结构保存到文本文件，以备后续加载。通过遍历词典，将键和值以“:”分隔的形式写入文档。

流程图如下：

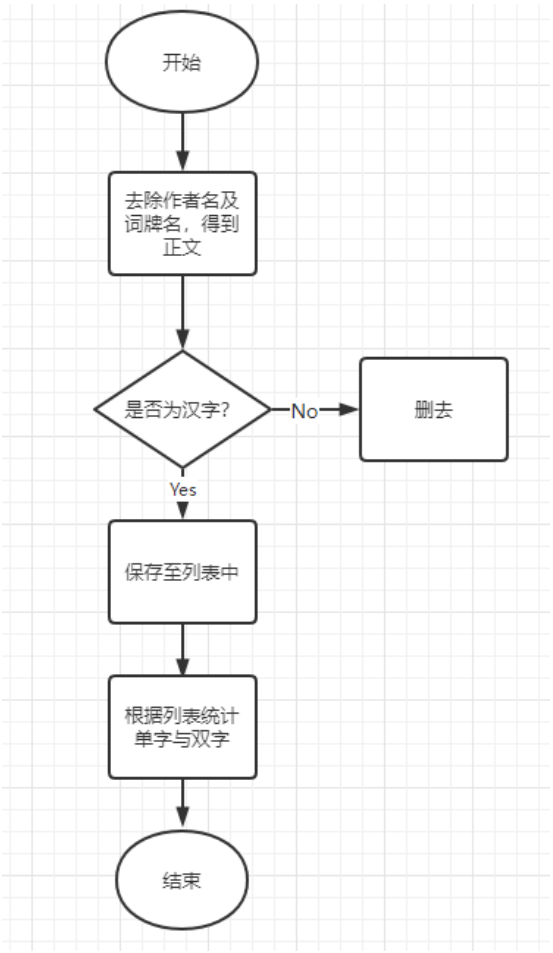

图 1 宋词词频统计系统流程

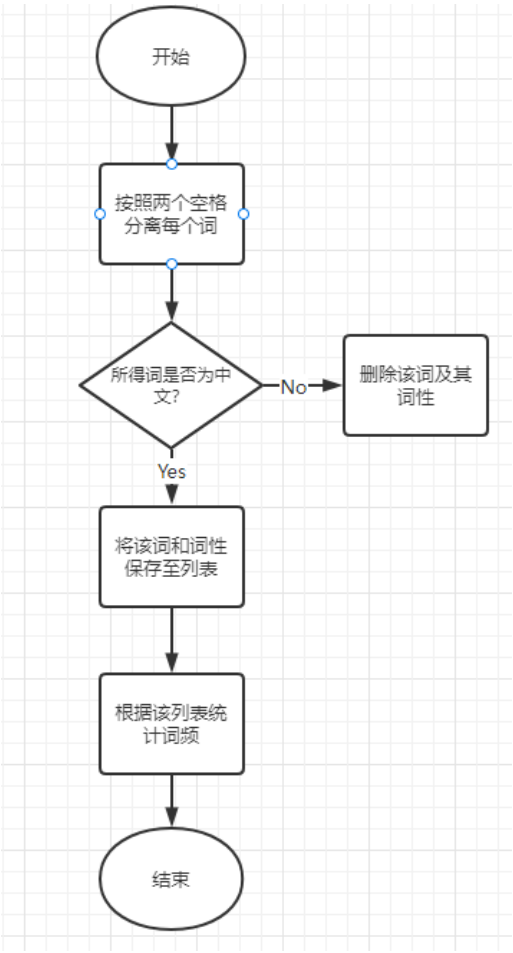

图 2 新闻文本词频统计系统流程

四、系统演示与分析

宋词文本词频统计系统：

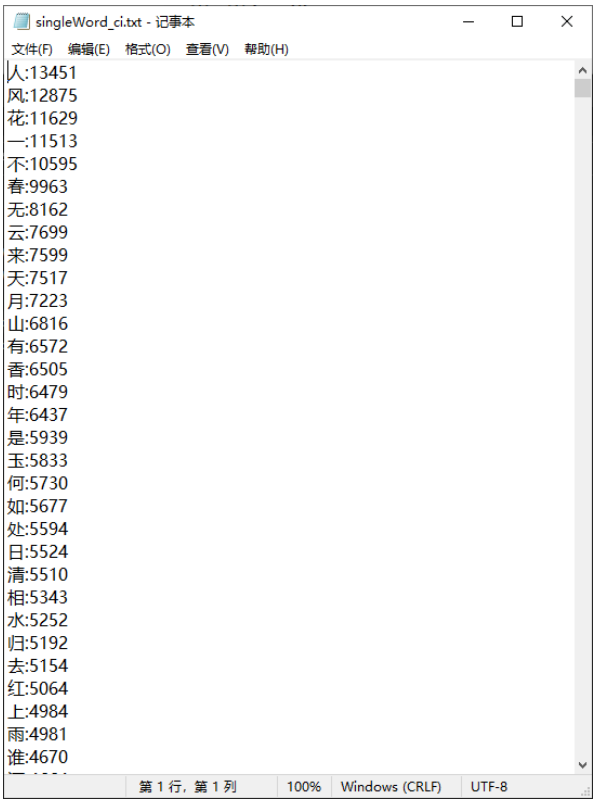

图 3 宋词文本 1-gram 词频统计

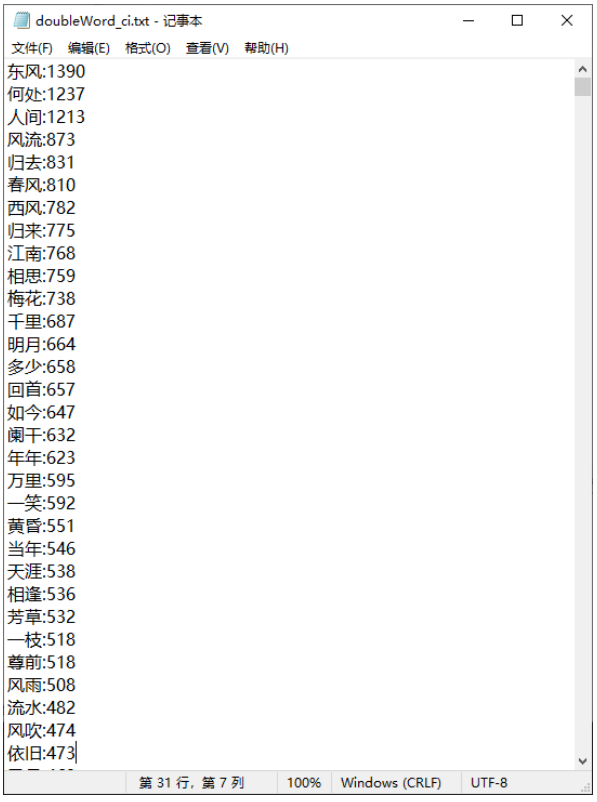

图 4 宋词文本 2-gram 词频统计

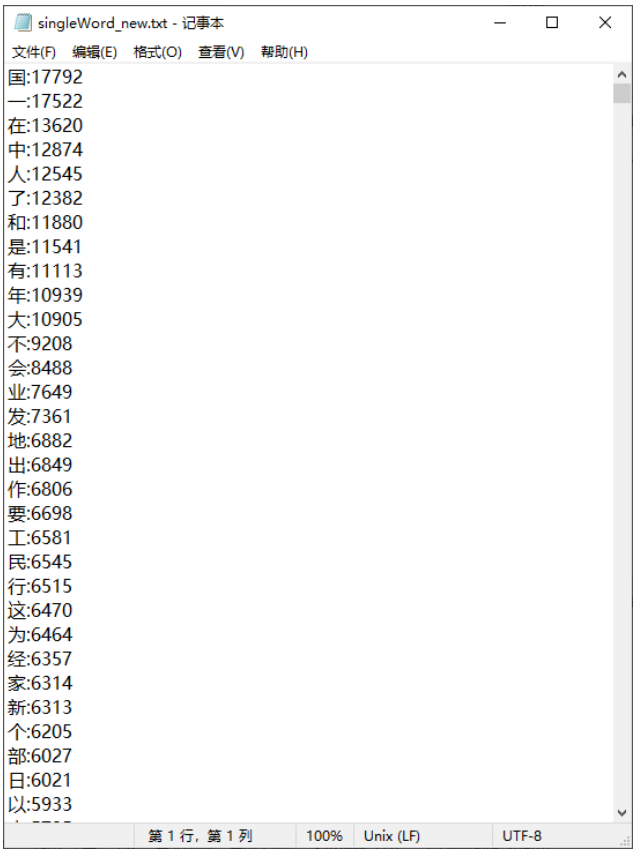

图 5 新闻文本 1-gram 词频统计

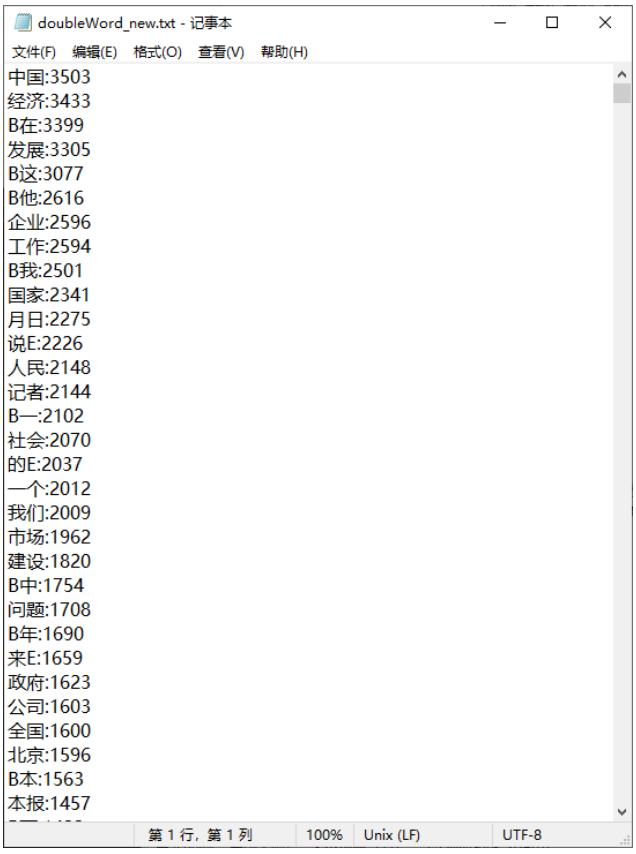

图 6 新闻文本 1-gram 词频统计

分析：能够准确地提取出所需统计的文本并对要统计的词频等进行了准确的统计。在进行统计的过程中主要遇到的问题：

1.在打开文件时的编码一开始使用的是“utf-8”,然后在执行的过程中发现报错：‘utf-8’ codec can’t decode bytes in position xxxx: illegal multibyte sequence。意思就是说某些汉字较为生僻以至于 utf-8 无法读取。

2.对于文本中的某些格式上需要避免统计的地方，在使用语句时不能够很好地将这些不符合要求的地方完全去除。

五、对这门课的感想、意见和建议

在这次实验中，我设计了一个词频统计系统，通过分析文本格式的特点，成功地统计了宋词文本和新闻文本中的 1-gram（单字）和 2-gram（双字）词的词频。词频统计是自然语言处理中的基础环节，它为分析整个语料库的特征提供了重要数据。通过这个实验，我积累了实际操作的经验，更深入地理解了自然语言处理的知识，并增强了我的编程能力。这次实践为我提供了处理和分析其他语句的基础，对我在自然语言处理领域的发展起到了积极的推动作用。

实验二、词汇知识库使用技术

一、研究背景

知识库是一种描述性方法，用于存储和管理知识，由知识和知识处理机构组成，形成一个特定的知识领域。

短文本自动生成技术属于自然语言生成（NLG）研究的范畴。这项技术使计算机能够根据知识库或逻辑形式的机器表述，生成符合语法和逻辑规则的自然语言文本。相对于长文本，短文本具有内容特征稀疏、噪声大、上下文依赖性强的特点。同时，受网络传播的影响，短文本还具备海量性、实时性和内容多样性等特征。

二、模型方法

本实验旨在通过研究宋词的格律，结合实验一中统计得出的单双字词信息，实现宋词的自动生成。在词的选择阶段，实验采用了轮盘赌方式。以下是轮盘赌选择法的模型和基本原理介绍。

轮盘赌选择法（roulette wheel selection）是一种简单且常用的选择方法。在该方法中，个体的选择概率与其适应度值成正比，即适应度越大，被选中的概率越高。然而，在实际轮盘赌选择中，个体的选择通常基于“累积概率”。

轮盘赌选择法的操作过程如下：

1. 计算每个个体的适应度值，并计算总适应度。
2. 根据每个个体的适应度值计算其选择概率，即个体适应度值除以总适应度。
3. 计算累积概率，即将各个个体的选择概率累加，形成一个累积分布。
4. 生成一个随机数，然后根据这个随机数在累积分布上的位置确定选择的个体。

通过这个过程，适应度更高的个体有更大的概率被选择，但所有个体都有一定的机会被选中，以保持选择的多样性。这种方法有效地模拟了自然选择中适者生存的原则，用于生成符合特定要求的宋词文本。

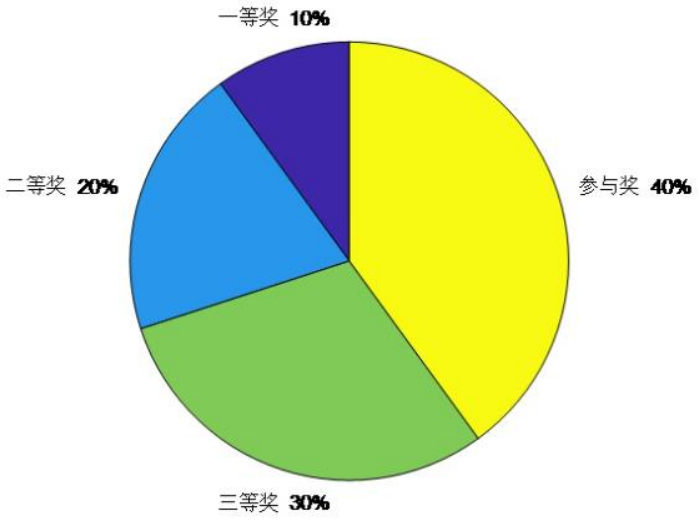

图 7 一个轮盘赌的模型

当我们直接转动轮盘时，抽中“参与奖”的概率显然最大，因为它在总体中

占比最高。这体现了“轮盘赌选择法”中占比越大，被选中概率越高的原则。然而，我们通常不使用抽中“几等奖”的概率等定性指标来表述每个部分被选中的概率。相反，我们引入了“适应度”和“累积概率”的概念。适应度表示个体在选择中的相对优势，与问题的优化目标相关。在这个情境中，适应度可理解为每个部分被选中的相对概率，与“几等奖”相对于总体的比例有关。累积概率是适应度值按比例累加形成的概率分布。通过计算累积概率，我们将每个部分的选择概率映射到一个区间，这个区间的长度与适应度值成正比。生成随机数后，根据其在累积概率分布上的位置确定选择的个体，以实现按适应度值比例选择的目的。引入适应度和累积概率的概念使得轮盘赌选择法更加灵活，能够更准确地控制不同个体被选中的概率，使其更贴近实际问题的优化需求。设某一部分 x(i)的适应度值表示为 f(xi)，该部分被选中的概率为 p(xi)，累积概率为 q(xi)，对应的计算公式如下：

$$p(x_i) = \frac{f(x_i)}{\sum_{j=1}^{N} f_j}$$

$$q(x_i) = \sum_{j=1}^{i} p(x_j)$$

在上述公式中，累积概率表示每个个体之前所有个体的选择概率之和，类似于转盘上的“跨度”，其中“跨度”越大，被选中的概率越高，这相当于概率论中的概率分布函数 F(x)。轮盘赌选择法的具体过程如下：

1. 计算每个个体被选中的概率，记为 p(xi)。
2. 计算每个部分的累积概率，记为 q(xi)，它是前 i 个个体被选中的累积概率之和。
3. 随机生成一个数组 m，数组中的元素取值范围在 0 和 1 之间，并按从小到大的方式进行排序。
4. 逐个比较累积概率 q(xi)与数组元素 m[i]，如果 q(xi)大于 m[i]，则个体 x(i)被选中；如果小于 m[i]，则继续比较下一个个体 x(i+1)，直至找到被选中的个体。
5. 若需要选择 N 个个体，则重复步骤（3）和（4）N 次，以得到所需数量的个体。

这个过程确保了个体被选中的概率与其适应度值成比例，同时通过随机性保持了一定的多样性。这是轮盘赌选择法在选择个体时的基本原理。

三、系统设计

1.打开开发环境，根据自己熟悉的语言，确定开发环境。本次实验我所采用的开发环境是 IDEA 2021.3.1，编程语言是 JAVA1.8。

2.将实验一中生成的词典到特定目录下。该过程在实验一中已完成，在此不再赘述。

3.将词典加载到内存中，主要包括词和词频。从文件中将单双字词及其出现频率重新读取，并将其保存至词典中以备使用。

4.采用随机生成，或者 n-gram 等算法，生成宋词。

这一过程，我们首先重新打开原本的宋词文本，研究了宋词的格律，由于宋词文本中所含的宋词数量众多，因此我编写了一段程序来先行对词牌名及其格律进行分类与保存，该程序的大致流程如下：

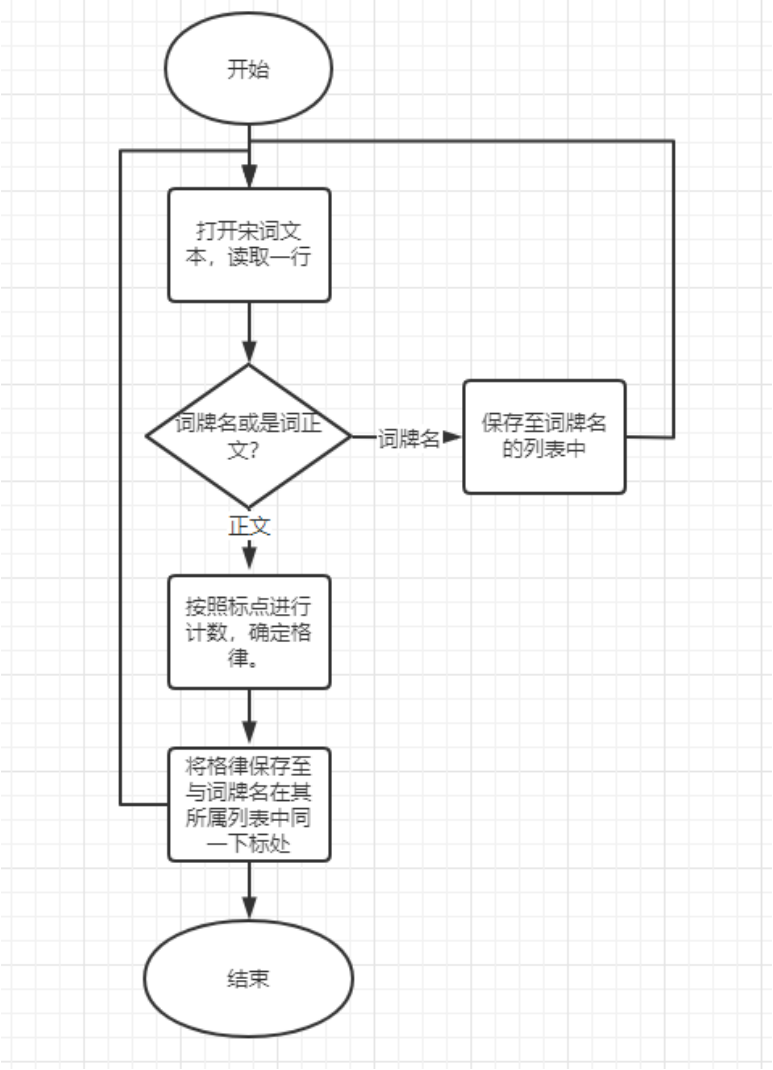

图 8 词牌名及其格律分类流程

我们根据每首词的词牌名获取其对应的格律。针对格律中的每一句，我们使用双字词进行填充。若该句字数为奇数，我们在最后选择一个单字进行填充。在选择词的基本方法上，我们采用轮盘赌策略。具体而言，我们通过计算每个词的累积概率来进行选择，以构建宋词的文本。这一过程通过不断重复，以确保生成的文本符合所选格律，并且整体风格自然流畅。

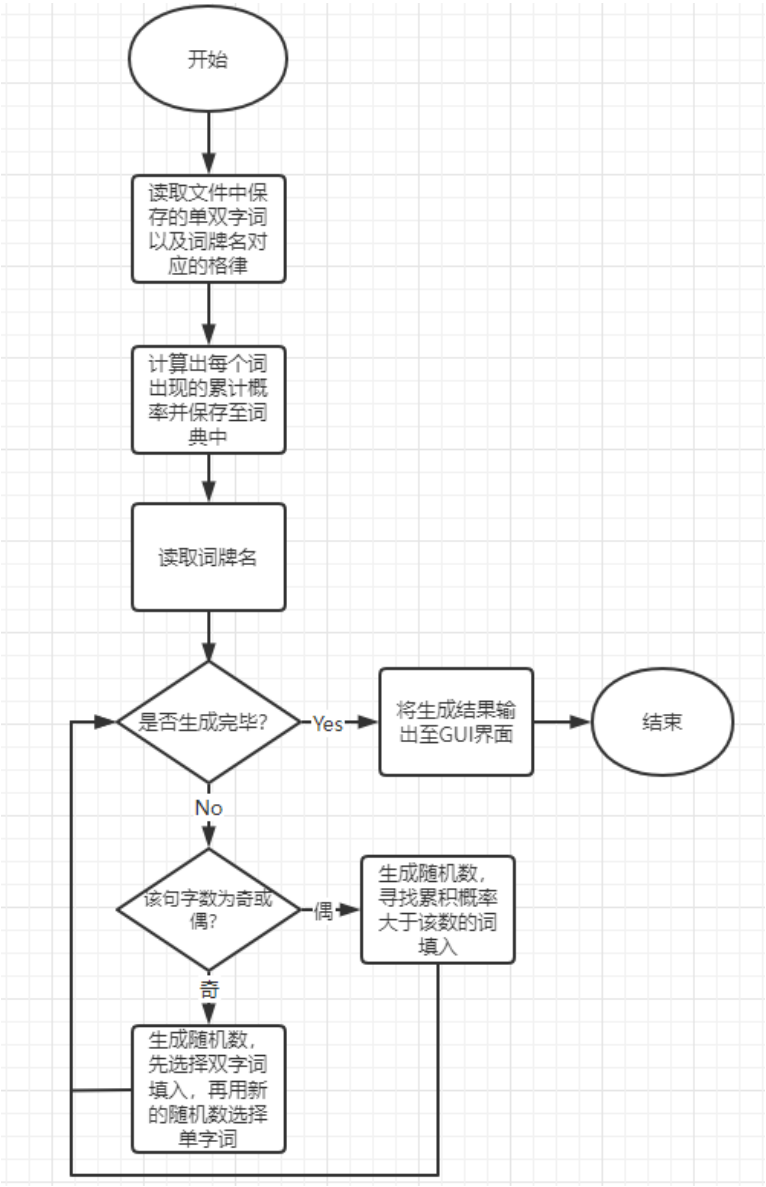

图 9 宋词自动生成系统流程

四、系统演示与分析

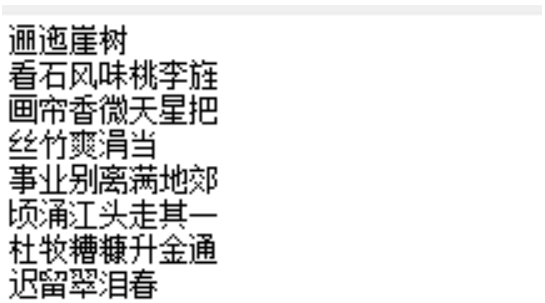

图 10 宋词自动生成的一个例子

该系统在获取用户输入的词牌名后，通过已统计的词牌名确定词的格律，并以此为基础生成宋词。然而，系统在用户点击“生成宋词”按钮后可能出现结果生成时间较长的问题，可能是由于搜索匹配的代码效率不足导致的。为了提高系统的响应速度，可以优化搜索匹配的算法，提高代码执行效率。

另外，系统在生成过程中存在一定概率出现非汉字的符号，如括号和引号等。这可能是由于前期清洗数据时的代码设置遗漏所致。为了改进这一问题，可以先判断每一句中是否含有括号，如果存在，则删除括号及其中的内容。然后再进行清洗和生成的过程，以确保词库中仅包含符合要求的汉字，从而避免非汉字符号出现在生成结果中。这样的改进将提高生成结果的准确性和质量。

五、对这门课的感想、意见和建议

在最终测试生成宋词的过程中，我意识到实验一中存在的清洗漏洞。在实验一中，我们只考虑了中文的逗号、句号、顿号以及虚缺号等特定符号，而未充分考虑其他可能出现的符号。这次发现的漏洞为我提供了一次宝贵的提醒机会，使我更加关注实验细节，以减少未来犯类似错误的可能性。

此外，本次实验基于实验一的基础进行了集中应用和深化。在实验一中，我们初步了解了词频统计等基本处理方式，而在实验二中，我们需要在已有的词库和词频基础上实现宋词的生成。这次实验为我提供了一个实践机会，使我能够进一步学习短文本生成的知识，并深化对轮盘赌算法、n-gram 文法等自然语言处理基本概念的理解。

总体而言，本次实验不仅巩固了之前学到的知识，还拓展了对自然语言处理领域关键概念的理解，为我今后在相关领域的学习和实践奠定了基础。

实验三、中文分词技术的应用

一、研究背景

中文分词是中文文本处理的基础步骤，也是实现中文人机自然语言交互的关键模块。与英文不同，中文句子中没有明确的词边界，因此在进行中文自然语言处理时，必须先进行分词，而分词的效果直接影响到词性、句法树等模块的效果。尤其在人机自然语言交互中，优秀的中文分词算法能够显著提高自然语言处理的效果，有助于计算机更好地理解复杂的中文语言。

人工智能在构建中文自然语言对话系统时，通过结合语言学知识不断优化，成功训练出一套具有良好分词效果的算法模型，为机器更好地理解中文自然语言打下了坚实基础。

中文分词根据实现原理和特点主要分为以下两个类别：

1. 基于词典分词算法：也称为字符串匹配分词算法。该算法按照一定策略，将待匹配的字符串与一个已建立好的庞大词典中的词进行匹配。若找到某个词条，则匹配成功，识别了该词。常见的基于词典的分词算法有正向最大匹配法、逆向最大匹配法和双向匹配分词法等。这类算法应用广泛，分词速度较快，研究者长期以来在不断优化基于字符串匹配的方法，如最大长度设定、字符串存储和查找方式，以及词表的组织结构，如采用 TRIE 索引树、哈希索引等。

2. 基于统计的机器学习算法：这类算法包括 HMM、CRF、SVM、深度学习等。例如，Stanford 和 Hanlp 分词工具基于 CRF 算法。基本思路是对汉字进行标注训练，不仅考虑了词语出现的频率，还考虑了上下文信息，具备较强的学习能力。因此，这类算法在歧义词和未登录词的识别方面表现出色。

二、模型方法

1. 隐马尔可夫模型（HMM）介绍：

隐马尔可夫模型是一种马尔可夫链的扩展，其状态无法直接观察到，但可以通过观测向量序列观察到。每个观测向量通过概率密度分布表示各种状态，形成状态序列。HMM 在语音识别、计算机文字识别、多用户检测等领域取得了显著成功。模型包括隐含状态集合、可观测状态集合、初始状态概率矩阵 π、隐含状态转移概率矩阵 A、观测状态转移概率矩阵 B。

- 隐含状态集合 S：马尔可夫链中的实际隐含状态。
- 可观测状态集合 O：与隐含状态相关联，可直接观察得到。
- 初始状态概率矩阵 π：描述隐含状态在初始时刻的概率。
- 隐含状态转移概率矩阵 A：描述隐含状态之间的转移概率。
- 观测状态转移概率矩阵 B：描述隐含状态到观测状态的转移概率。

一般用 $\lambda=(A,B,\pi)$表示一个隐马尔可夫模型。

2. 前向最长匹配算法（FMM）：

FMM 是一种基于词典的分词方法，通过从左向右扫描文本，寻找词的最大匹配。例如，词典中包含“钓鱼”和“钓鱼岛”，对于句子“钓鱼岛属于中国”，FMM 会将其分词为“钓鱼岛/属于/中国”。

这种算法的思想简单而直观，能够有效地利用词典信息进行分词，但也存在一些局限性，例如无法处理歧义词和未登录词。 FMM 在中文分词中起到了基础

作用，尤其在早期是一种应用广泛且速度较快的分词算法。

3.逆向最长匹配算法（BMM）：

逆向最长匹配算法是一种基于词典的中文分词方法，与前向最长匹配算法相反，它从右向左扫描文本，寻找词的最大匹配。与前向最长匹配算法类似，BMM也依赖于预先建立的词典，其中包含各种可能的词汇。

算法的基本思想是从文本的末尾开始，选择最长的词进行匹配，并将匹配到的词作为一个词块从文本中去除。然后，重复这个过程，直至处理完整个文本。例如，对于句子“钓鱼岛属于中国”，如果词典中包含“钓鱼”和“钓鱼岛”，BMM 会将其分词为“钓鱼/岛/属于/中国”。

二、 系统设计

本实验采用前向最长匹配算法（FMM）作为中文分词系统的核心算法，系统设计主要包括以下步骤：

1. 选择开发环境：

打开开发环境，根据自己熟悉的语言，确定开发环境。本次实验我所采用的开发环境是 IDEA 2021.3.1，编程语言是 JAVA1.8。

2. 导入分词词典或模型：

为构建分词系统，我们导入相应的分词词典。在本次实验中，我们利用实验一中统计的人民日报语料库的 1-gram 词作为基础。通过读取词频统计结果，将人民日报语料库的所有 1-gram 词加入分词词典。

3. 实现 FMM 算法：

FMM 的实现流程如下：

- 获取用户在文本框内输入的文字内容。
- 将输入内容作为匹配对象，逐一与分词词典中的词语进行匹配。
- 若成功匹配，则保存匹配的词语，并在其后插入“/”作为分词标志。
- 若匹配失败，则从右边删去一个字，继续逐一匹配，直至成功匹配为止。
- 重复上述过程，直到整个语句分词完毕。若最后一个字仍无法匹配，将其分离，并匹配后面的语句。

通过以上流程，实现了基于 FMM 的中文分词系统，该系统能够有效地利用词典信息进行分词，提高对用户输入的文本的处理效率。

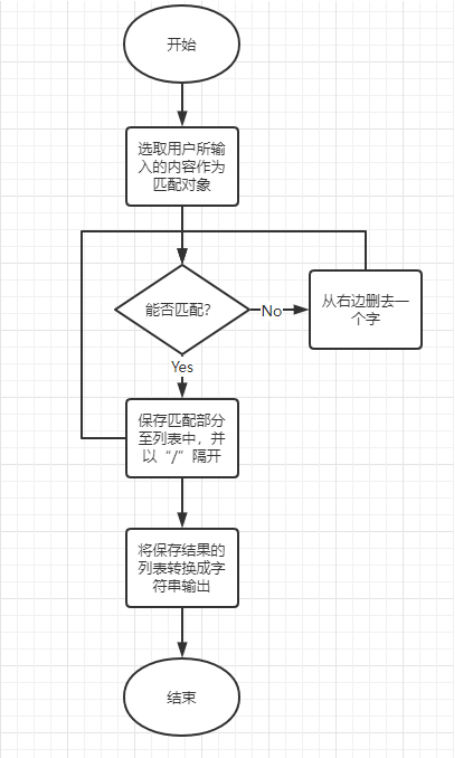

图 11 FMM 算法实现流程

```
// 读取文件构建词典
try {
	String words;
	String first, second;

	BufferedReader doubleBufferRead = new BufferedReader(new FileReader("src/data/doubleWord_new.txt"));

	words = doubleBufferRead.readLine();
	while (words != null) {
		first = words.split(":")[0];
		second = words.split(":")[1];
		map.put(first, second);
		words = doubleBufferRead.readLine();
	}
	doubleBufferRead.close();
} catch (Exception e) {
	System.out.println(e);
}
```

四、系统演示与分析

```
原语句：明天我们将会去哪里
分词结果：明天/我们/将/会/去/哪里/
原语句：深刻洞察了当前经济复苏中的痛点
分词结果：深刻/洞/察/了/当前/经济/复苏/中/的/痛/点/
原语句：这些倡议立足当下又放眼长远
分词结果：这些/倡议/立足/当下/又/放眼/长远/
```

分析：

该系统在基本输入语句上能够进行分词操作，并通过 GUI 窗口与用户进行交互，实现了基本的语句分词需求。用户还可以在分词后查看历史分词记录。

存在的问题：

1. 由于词库规模有限，系统对一些词语，如地名或特定专有名词的划分可能不够准确，存在分词不全面的情况。

2. 面对含有歧义的句子时，系统可能出现划分错误，因为当前算法可能无法明确上下文的语境。

改进措施：

1. 扩充词库：通过引入更全面的词库，特别是包含地名和专有名词的扩充，可以提升系统对特殊词语的划分准确性。

2. 引入上下文信息：考虑使用基于上下文的分词算法，例如基于统计的机器学习方法，以更好地处理歧义情况，提高系统的鲁棒性。

通过以上改进，系统可以更全面准确地进行中文分词，提高在不同语境下的适用性。

五、对这门课的感想、意见和建议

本次实验中，我成功实现了一个基本的中文分词系统，采用了相对简单的 FMM 算法，主要通过字符串与词典中的词进行比较来完成分词。然而，这个简单的实现还存在着很大的提升和改进的空间。在编写代码和进行测试的过程中，我深刻认识到中文分词在自然语言处理中的重要性，并了解到不同算法和模型之间的原理与效率的差异。系统中存在的问题也让我认识到解决中文分词需要不断研究和深入探讨，激发了我深入研究的兴趣和动力。

关于自然语言处理这门课程的感想，它是机器学习课程的一门更深入的延伸，涵盖了语料库、基本数学概率知识以及后续模型和算法等方面。通过课程实验的实践，我们得以将理论知识具体应用和实现，全方位地理解了自然语言处理领域的体系结构。这门课程的学习不仅提高了我的眼界和实践能力，还加强了我的代码能力。自然语言处理为我在计算机学习的道路上迈出了更大的步伐，为我未来的学术和职业发展带来了积极的影响。