基于自然语言处理的歌词信息挖掘

郑亚斌　　刘知远　　孙茂松
（清华大学计算机科学与技术系，智能技术与系统国家重点实验室　北京　100084）
{zhengyb06, zy-liu02}@mails.tsinghua.edu.cn　　sms@mail.tsinghua.edu.cn

摘要：我们在歌词上做了一些传统的自然语言处理相关的实验。大部分的歌曲都具有与其相匹配的歌词，歌词也是歌曲语义上的重要表达，因此，对歌词的分析可以作为歌曲音频处理的互补。我们利用齐夫定律对歌词语料库的字和词进行考察，实验表明，其分布符合齐夫定律。利用向量空间模型的表示，我们可以找到比较相似的歌词集合。另一方面，我们探讨了如何利用歌词中的时间标注信息进行进一步的分析，利用时间标注信息可以找到歌曲中重复的片段，并且可以对歌曲的节奏进行划分，初步的实验表明，我们的方法具有一定的效果。

关键字：歌词；齐夫定律；向量空间模型；k-近邻；节奏

Information Mining of Song Lyrics Based on Natural Language Processing Techniques

Zheng Yabin　　Liu Zhiyuan　　Sun Maosong
（State Key Lab of Intelligent Technology and Systems,
Dept. of Computer Science and Technology,
Tsinghua University, BeiJing, 100084 ）

Abstract: We report experiments on song lyrics based on standard natural language processing techniques. A significant amount of music songs have suited lyrics, which play a important role of the semantics of a song, therefore, analysis of lyrics may be a complement of acoustic methods. We investigate the lyrics corpus based on Zip'f Law using both character and word as a unit, results roughly obey Zip'f Law. Also, by the means of vector space model, we can find a set of lyrics that are similar to each other. Moreover, we discuss how to use the time annotation for further study; we can detect the repeated part of songs as well as categorization of rhythms. Preliminary experiment shows the effectiveness of our proposed method.

Keyword: Song Lyrics; Zipf's Law; VSM; k-NN; rhythm

1 前言

随着全球信息化进程的加快发展，Web 信息呈现出爆炸式增长的趋势，其中音频数据占一定的比重，如何挖掘这些音频数据并使之服务于用户成为一个热门的研究方向[1, 2]。通过现有的搜索引擎，例如 Google、Baidu 等，人们可以利用歌手或者歌曲的名称挖掘出自

己想要的音频信息，但是我们更着重于考虑类似于协作性过滤的歌曲推荐系统，例如：用户喜欢听 X，也很有可能喜欢听 Y 这首歌，X 和 Y 在音频上具有某种程度的相似性，已经有一些音频处理的技术用来解决这方密的问题[3]。

而对于每一首歌来说，人们可能还会关心表达其语义的部分——歌词[4, 5]，和音频处理的技术相比，在歌词层面上的处理有一定的优势：首先，在 Web 上，大部分的歌曲都有对应的歌词信息，且容易收集；其次，和歌曲相比，歌词的数据量要小很多，通常情况下来说，一首歌的大小大约为 4M，而其对应的歌词一般只有 2KB 左右，在面对海量数据的情况下，对于存储空间的要求低；再次，歌词具有很丰富的语义信息[5]，可以解决一些在音频上比较难处理的问题，例如：歌曲语种识别[4]等。而且这也使得一些传统的自然语言处理技术可以应用在歌词上；最后，一些特殊格式的歌词文件具有时间标注信息，我们可以从文本处理的角度解决音频切割划分的问题。

本文利用自然语言处理的技术对歌词进行了一些实验，分别统计歌词语料库和通用语料库在字、词单位上的频率分布，发现两个语料库都近似符合齐夫定律[6]（Zipf's Law），和通用语料库相比，歌词语料库的高频词分布有略微的差别。在文本分类领域，经常用向量空间模型[7]来表示文本，我们在歌词语料库上也做了类似的处理，结合 K-近邻[8]算法找到和当前歌词在语义上比较相似的样本，并给出了结果的示例。

另外，如何利用歌词中的时间标注信息也值得我们探讨，在本文中的实验中，建立了一个基于文本的音频检索定位系统，用户的输入是一个查询词，系统给出所有包含查询词的歌词片段，并利用时间标注信息定位到相应的音频片段，实验证明，有比较高的准确率。这就使得通过文本处理的方法解决音频切割划分的问题成为可能。其次，利用时间标注信息我们可以得到歌词中每一句延续的时间，进而对歌曲的节奏进行分类，例如划分成慢、中、块三个类别，可以对歌曲的信息进行更丰富的描述和处理，和音频处理的相关技术起到互补的作用。

本文的组织结构如下：第二节主要介绍相关工作；第三节介绍了实验用到的数据集以及实验的设置和结果等方面；第四节给出了结论及未来的工作展望。

2 相关工作

近年来，如何利用歌词的信息提高音乐检索系统的效果引起了人们的注意。BainBridge[9]等人的实验表明：人们更倾向于用歌词对歌曲进行检索，大约 28.9%的用户直接用歌词片段而不是音频等其他信息。Besson[10]等人指出：歌词的语义信息能很好的表达歌曲的含义，而且是和曲调无关的，是歌曲在音频和文本两个方面独立的互补的表达。

Scott[11]等人利用 400 多首歌的歌词进行了分类方面的实验，实验表明，利用传统的词袋（Bag-of-words）模型并结合 WordNet 信息能提高分类的准确率。Baumann[1]等人把基于本体的文本检索技术用在歌词上，利用这种表示形式的歌词判别歌曲的相似度。Logan[5]等人利用了 16000 条歌词信息来确定歌手之间的相似度，并分析了不同种类的歌曲在用词上的差异，实验表明单纯利用歌词得到相似度匹配结果要比从音频的角度差，但是歌词和音频存在潜在的互补提高效果的可能性。

Mahedero[4]等人在前人的基础上，做了更详尽的实验，综合了自然语言处理方面的技术，进行了歌曲语种识别、歌词结构分析、歌词分类、相似度计算等方面的工作。实验表明在歌曲语种识别方面，基于歌词的方法要比音频的角度效果好，但是如何把自然语言处理的技术更好的用在歌词上还值得进一步的研究。

在文本分类算法中，k-近邻具有比较好的分类效果[12]，其主要思想是，给定一个测试

集样本，系统在训练集中找到和该样本最为相似的近邻，用这些近邻的样本标签决定测试集样本的标签，通常是一个加权投票的过程，其权重一般设置为每一个近邻和当前测试样本的相似度，相似度越高，权重越大。本文中采用了类似 k-近邻的方法，只不过我们没有标签的信息，只是找到和当前歌词样本在语义上比较相近的样本。

音频的切割划分[13]一直也是音频检索的研究方向，用户可能只对一首歌里面包含特定关键字的音频片段感兴趣，利用有时间标注信息的歌词可以方便的做到这一点。

据我们所知，目前还没有人在中文歌词上做过类似的实验，另外，如何利用具有时间标注信息的歌词提高效果也属于空白，本文主要对以上几个方面展开工作。

3 实验

目前还没有比较权威的中文歌词的语料库，实验的数据集全部从网上搜集得到，一共 15737 首歌的歌词，涉及到 450 多位歌手，在数量上和 Logan 等人的实验比较接近。所有歌词都是 LRC 格式的，其和普通文本格式的不同在于多了时间标注的信息，这在后续的实验中将会用到。下表是 LRC 格式歌词的一个示例：

其中 ti 行表示歌曲名称，ar 行表示歌手名称，al 行表示专辑名称，by 表示编辑该歌词的用户信息。LRC 格式歌词具有标准的时间表示形式，一般为 MM:SS，其中 MM 表示分钟，SS 表示秒，用这种格式表示当前歌词片段在音频中开始的时间。如果含有多个这种格式，表示当前歌词片段在音频中重复多次。

[ti:爱情证书]
[ar:孙燕姿]
[al:同名专辑]
[by:炫网资讯 Liuxuan.com]
[00:17.00]寂寞当然有一点
[00:23.00]你不在我身边
[00:25.00]总是特别想念你的脸
[00:33.00]距离是一份考卷
[00:39.00]测量相爱的誓言
[00:41.00]最后会不会实现
[01:51.00][00:48.00]我们为爱还在学
[01:57.00][00:55.00]学沟通的语言

表 1：LRC 格式歌词举例

3.1 歌词语料库和通用语料库的比对

为了更好的了解歌词中的字和用词的分布，我们用通用语料库进行了比较，通用语料库采用的是读者 200 期的文章内容。

在字的层面上，我们选择了频率排在前 15 的字，统计得到的数据如下：

Rank（歌词）	字（歌词）	频率（歌词）
1	我	0.0476
2	的	0.0448
3	你	0.0412
4	不	0.0292
5	一	0.0226
6	爱	0.0156
7	是	0.0155
8	在	0.0128
9	心	0.0113
10	有	0.0104
11	了	0.0095
12	想	0.0080
13	人	0.0077
14	这	0.0075
15	要	0.0066

表 2：歌词语料库高频率字统计

Rank（通用）	字（通用）	频率（通用）
1	的	0.0583
2	一	0.0257
3	我	0.0180
4	了	0.0179
5	不	0.0164
6	是	0.0157
7	他	0.0143
8	在	0.0139
9	人	0.0109
10	这	0.0108
11	有	0.0096
12	上	0.0069
13	你	0.0067
14	大	0.0063
15	地	0.0062

表 3：通用语料库高频率字统计

从上面的两张表可以看出，歌词语料库和通用语料库在高频字上的分布有所不同，在歌词语料库中，出现频率最高的字是“我”，而在通用语料库中出现频率最高的字是“的”，另外，在歌词语料库前 20 名中，“爱”，“心”，“想”等字并没有出现在通用语料库前 20 名，且相同的字在排名上也有所差别。

为了更详细的了解这一点，我们画出了两个语料库以字为单位的齐夫定律曲线，齐夫定律指出：在自然语言的语料库中，把每一个单位（字或者词）按照其出现的频率 F 从高到底排列，并以自然数对 Rank 编号，则有$F \times Rank \approx C$的规律，其中 C 为一个常数。从图中可以看出，除去特别高频和低频的部分，两个语料库在字单位上都很好的符合了齐夫定律。

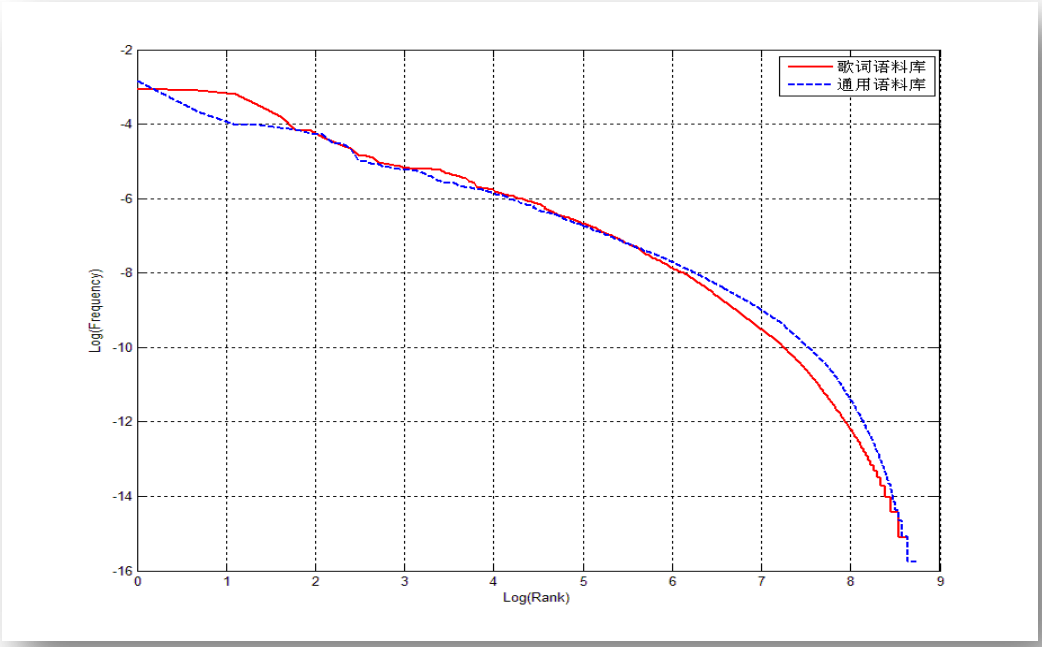

图 1：歌词语料库、通用语料库 Zipf’s Law 曲线（字为单位）

在词的层面上，我们做了类似的实验，我们选择了频率排在前 15 的二字词，统计得到的数据如下：

Rank（歌词）	词（歌词）	频率（歌词）
1	一个	0.0028
2	没有	0.0027
3	自己	0.0023
4	我们	0.0022
5	知道	0.0019
6	世界	0.0016
7	永远	0.0015
8	怎么	0.0013
9	快乐	0.0012
10	如果	0.0012
11	寂寞	0.0012
12	爱情	0.0012
13	一切	0.0012
14	这样	0.0012
15	不要	0.0012

表 4：歌词语料库高频率二字词统计

Rank（通用）	词（通用）	频率（通用）
1	一个	0.0047
2	我们	0.0032
3	自己	0.0028
4	他们	0.0028
5	没有	0.0022
6	什么	0.0017
7	这个	0.0013
8	孩子	0.0013
9	可以	0.0013
10	这样	0.0012
11	因为	0.0012
12	知道	0.0012
13	生活	0.0011
14	如果	0.0010
15	美国	0.0010

表 5：通用语料库高频率二字词统计

从上面两张表可以看出：在以词为单位的情况下，和以字单位有相似的结果。从二字词的角度来分析，和以字为单位相比，两者之间的差别要大一些。二字词体现了歌词中和通用语料库中不同点：例如：“世界”，“永远”，“快乐”，“爱情”等。

类似的，我们也画出了两个语料库以词为单位的齐夫定律曲线：

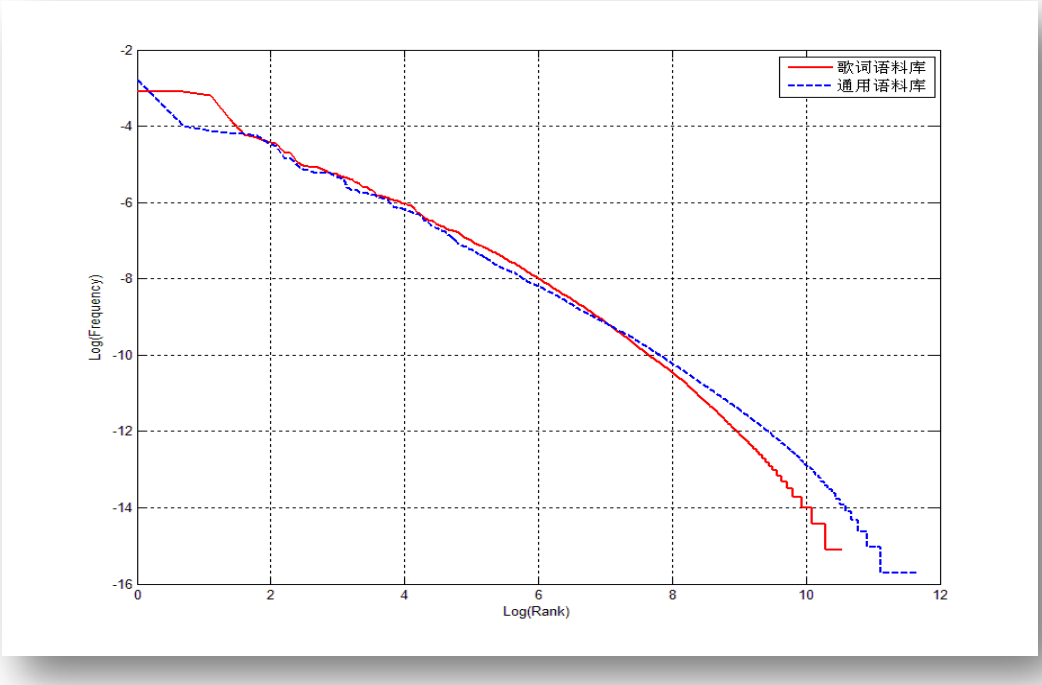

图 2：歌词语料库、通用语料库 Zipf’s Law 曲线（词为单位）

3.2 歌词的相似度匹配

在传统的文本分类算法中，文本的表示主要采用向量空间模型（VSM），其主要思想是用向量来表示文本：$(\omega_1,,\omega_2,\omega_3,\dots,\omega_n)$，其中$\omega_i$为第 i 个特征的在文本中的权重，在本文的实验中，我们采用词作为特征，并利用 TF-IDF 公式计算每个特征在文本中的权重，TF-IDF 的公式为：

$$\omega(t,D)=\frac{\log\left(1+\mathrm{tf}(t,D)\right)\times\log\left(1+{}^{N}/_{n_t}\right)}{\sqrt{\sum_{t\in D}\left[\log\left(1+\mathrm{tf}(t,D)\right)\times\log\left(1+{}^{N}/_{n_t}\right)\right]^2}} \qquad (1)$$

其中$\omega(t,D)$表示词 t 在文档 D 中的权重，$\mathrm{tf}(t,D)$为词 t 在文档 D 中出现的次数，N 为训练文档总数，n_t表示所有训练文档中出现词 t 的文档数，分母为归一化因子，使得向量长度为 1。

我们把歌词当作文本进行预处理，首先经过中文分词（实验中采用的是中科院计算所汉语词法分析系统 ICTCLAS），经过 TF-IDF 权重计算后，每个歌词样本就表示成了一个向量。由于中文的特殊性，我们往往会面临维数过高的问题，在这次实验中，我们得到每个样本的维数为 29500 维，这样会提高计算复杂度和时间消耗，为此，我们利用全局的 TF-IDF 值进行降维。具体的来说，对于每一个词，我们计算它在全部文本中的 TF-IDF 权重，并按大小排序，选择了前 3000 个词作为最终的特征表示。

在实际的应用中，用户可能想知道还有那些歌词和当前歌词样本比较相似，为此，我们定义两个歌词样本的相似度如下：

$$\mathrm{Sim}(D_i,D_j)=\frac{\sum_{k=1}^{n}\omega_{ik}\times\omega_{jk}}{\sqrt{\left(\sum_{k=1}^{n}{\omega_{ik}}^2\right)\left(\sum_{k=1}^{n}{\omega_{jk}}^2\right)}} \qquad (2)$$

其中$\mathrm{Sim}(D_i,D_j)$表示文档D_i和D_j的相似度，n 表示特征的维数，ω_{ik}表示文档 i 在第 k 维特征上的权重，上式即为两个向量的夹角余弦。

在实验中，我们采用了 k-近邻的算法，给定一个歌词样本，找到与其最为相近的邻居，由于我们实验中并没有标签的信息，所以没有进行分类的实验，在这里给出一些示例：

歌词样本：《毛主席派人来》	毛主席呀派人来 雪山点头笑罗彩云把路开 一条金色的飘带 把北京和拉萨连起来 我们跨上金鞍宝马哟 哈达身上带到北京 献给毛主席 感谢他给我们带了幸福来 毛主呀席派人来 神兵下凡界罗风扫乌云开 千年的大山被推倒 百万农奴站起来 我们高举红旗火把哟 哈达身上带到北京 献给毛主席 感谢他给我们带了幸福来
1.《毛主席的恩情比山高比水长》	巍巍井冈山高 滚滚延河水长 毛主席的恩情比山高比水长 啊毛主席的恩情比山高比水长 天安门红旗飘 中南海明灯亮 赞颂毛主席的歌千代传万代唱 啊赞颂毛主席的歌千代传万代唱 啊赞颂毛主席的歌千代传万代唱
2.《萨拉姆毛主席》	毛主席呀毛主席耶 日夜都在想念你 毛主席呀毛主席耶 日夜都在想念你 我要勤生产多卖力耶 把那盘缠来攒起耶 有一天我去看你 我就说毛主席耶来来来 普天下的人民都爱你耶 萨拉姆毛主席 哎萨拉姆毛主席耶 萨拉姆毛主席 毛主席呀毛主席耶 日夜都在想念你 毛主席呀毛主席耶 日夜都在想念你 今天晚上我就要 骑上毛驴去看你耶 到了北京见到你 我就说毛主席耶来来来 普天下的人民都爱你耶 萨拉姆毛主席 哎萨拉姆毛主席

表 6：歌曲相似度匹配示例一

歌词样本： 《爱情证书》	寂寞当然有一点 你不在我身边 是特别想念你的脸 距离是一份考卷 测量相爱的誓言 最后会不会实现 我们为爱还在学 学沟通的语言 学着谅解学着不流泪 等到我们学会飞 飞越黑夜和考验 日子就要从孤单里毕业 我们用多一点点的辛苦 来交换多一点点的幸福 就算幸福还有一段路 等 我们学会忍耐和付出 这爱情一定会有张证书 证明从此不孤独
1.《爱情字典》	我已经学会爱情的语言 可是却失去你我世界 爱是一条曲折的线 将你我带往两边 分开的两个人 怎么都不能回到起点 在爱情的字典里找不到永远 我们越走越远两个世界 新的感觉也许偶尔会出现 怎么没有了你都不对 陌生的城市生活多考验 最近的天空多半是雨天 因为爱情输给时间 所以要自己体验 不管泪水多咸 有一天我会告别从前 在爱情的字典里找不到永远 等到哭红双眼我才发现 爱情有一条看不见的界限 我们都过不了那条线 到另一边
2.《我在身边》	星星挂在天边 就像梦想来不及实现 把过去想了一遍 想起眷恋你的昨天 等待是久了一些 时间沉默地过了几年 相爱是一种语言 只是你不说我如何听见 我在身边 怎么你看不见 别让我痴心地守候 看不到终点 我在身边 希望最后你能够发现 我能给你的温柔 经得起考验 星星消失在天边 就像诺言来不及实现 把未来想了一遍 仍然是没有把握的明天 挣扎是久了一些 伤痛化成了繁星点点 爱你是一种考验 就让我再赌一个明天 我在身边 怎么你看不见

表 7：歌曲相似度匹配示例二

3.3 时间标注信息

如前文所述，如何利用 LRC 格式歌词中的时间标注信息是值得我们考虑的一个问题，为此，我们简单的做如下的处理：求得每句歌词的演唱节奏，用平均每个字延续的时间来衡量：

$$\mathrm{AverageTime}(L_i) = \mathrm{Duration}(L_i) \Big/ \mathrm{Length}(L_i) \qquad (3)$$

其中$\mathrm{AverageTime}(L_i)$表示当前歌词片段每个字延续的时间，用这个量可以刻画其节奏，时间越短，表示节奏越快，$\mathrm{Duration}(L_i)$表示当前歌词片段延续时间，$\mathrm{Length}(L_i)$表示当前歌词片段长度。

通常情况下，每首歌都有某些的片段会重复多次出现，而这些片段往往是整首歌的高潮或者主旋律部分，也是更容易被听者记住的部分。我们可以很容易的从歌词中发现这一点，这些片段在文本的表示上是差不多的，有些甚至完全一样。在本文的实验中，我们从歌词片段每个字延续的时间来考虑这个问题。

下图是一首歌的时间曲线示例：

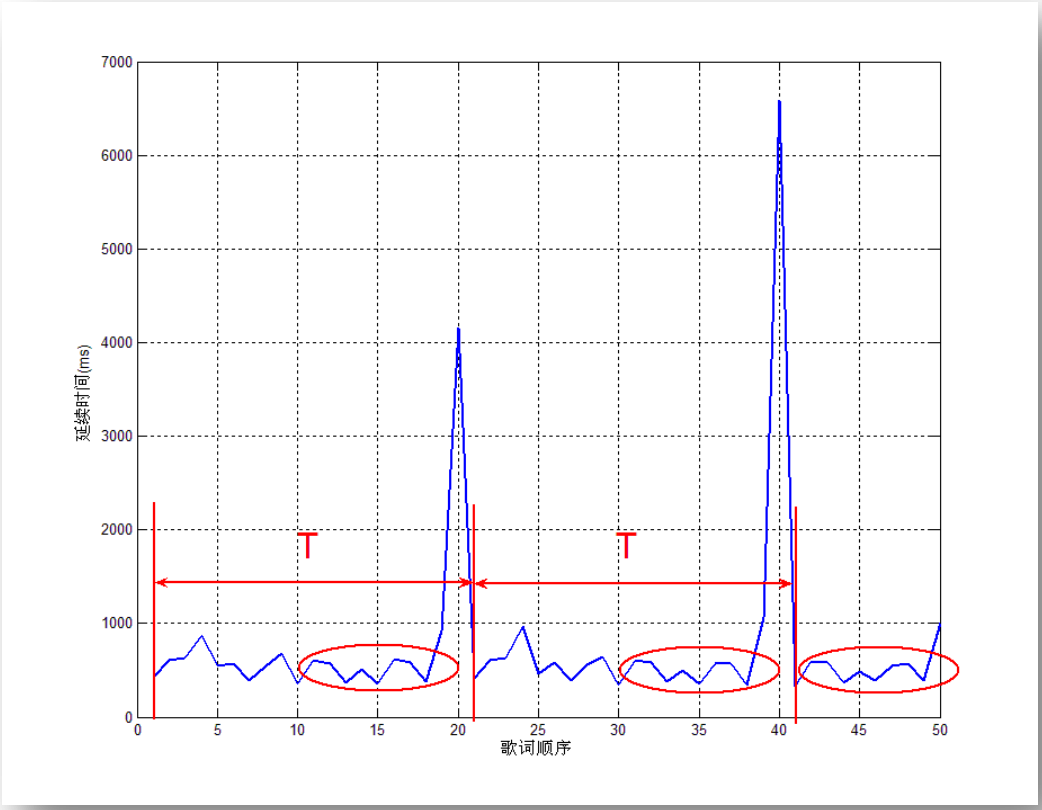

图 3：歌曲时间曲线

其中横坐标表示歌词的顺序，从 1 开始编号，纵坐标表示当前歌词片段每个字延续的时间，单位为毫秒。如图标注所示，我们可以很容易的发现两个“周期”，其曲线的形状基本类似，另外，三个椭圆标注的部分曲线形状也相似，这些都是在歌曲中重复出现的片段。两个尖峰表示停顿，即并没有人声，仅仅是背景音乐的播放。从这里可以看出，我们可以从歌词中挖掘出时间的信息，并具有比较好的效果，这可以作为音频处理的一个补充。

如前所述，我们可以利用时间标注信息对每首歌节奏进行分析。为了有一个更细致的分析，我们统计了整个歌词语料库中的时间信息，见下表：

歌曲数（首）	歌词数（句）	每首歌含歌词数（句/首）	平均延续时间（ms）
15737	480325	30.520	570.7203

表 8：歌词语料库数据统计

从上表中可以看出，平均每首歌含有大约 30 句歌词，平均每句歌词的“节奏”大约为 570ms/字，也就是说歌手唱歌时平均每半秒钟一个字。为了进一步的分析数据，我们按照歌词片段每个字延续的时间画出数据的直方图如下：（横坐标表示区间，从 0ms 到 1000ms，区间的间隔大小为 10ms，纵坐标表示落在该区间范围内歌词片段数量）

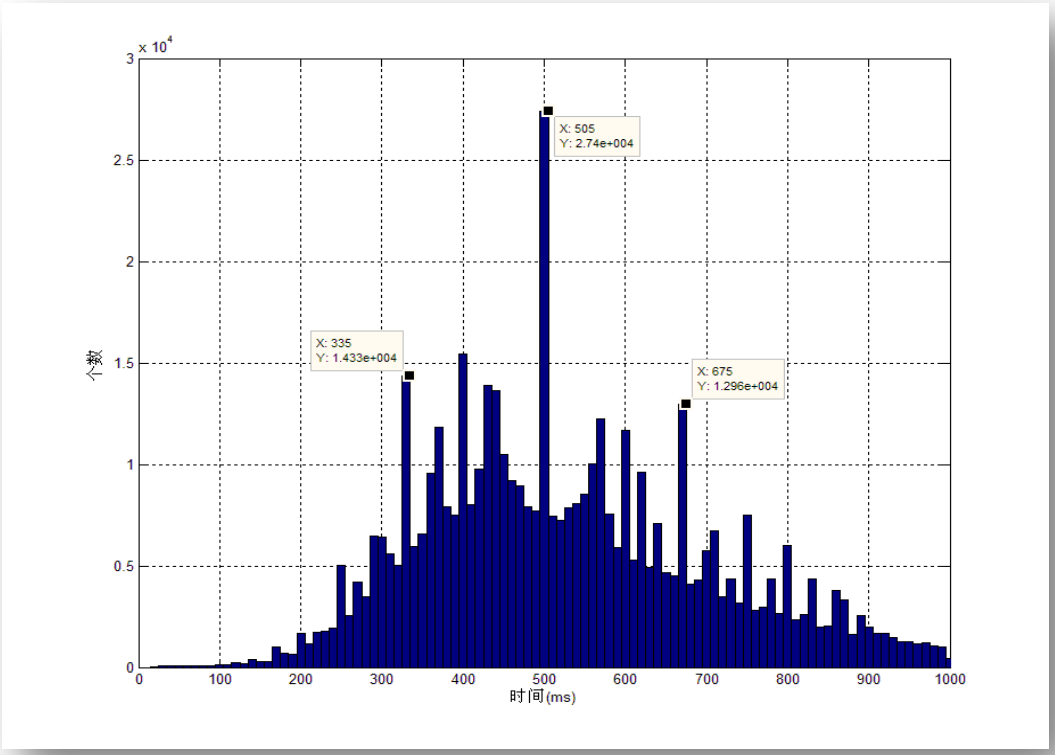

图 4：歌词延续时间直方图分布

从其分布的直方图可以看出，有三个明显的尖峰，分别为 330ms~340ms 区间，含有 14330 个样本；500ms~510ms 区间，含有 27400 个样本，670ms~680ms 区间，含有 12960 个样本。因此，我们按照区间分布对歌曲的节奏进行分类：落在 0ms~335ms 区间内的作为快节奏；落在 336ms~505ms 区间内的作为中快节奏；落在 506ms~675ms 区间内的作为中慢节奏；落在 676ms 之外的作为慢节奏。根据这个划分，对歌曲的分布做如下的统计：

歌词数（句）	快节奏歌词	中快节奏歌词	中慢节奏歌词	慢节奏歌词
480325	54087（11.3%）	169570（35.2%）	157358（32.8%）	99310（20.7%）

表 9：歌词节奏分布统计

从上表中可以看出：大部分的歌词属于中快或者中慢节奏，这和我们定义的范围有关，这里我们只是给出初步的实验结果，如何更好的自动调整范围，达到比较好的划分效果值得我们进一步的思考。

3.4 音频定位

歌词中的时间标注信息还可以帮助我们定位音频。在本文的实验中，用户的输入是一个查询词，系统在歌词语料库中查找包含查询词的片段，并利用其对应的时间标注信息定位到相应的音频片段，完成检索的功能，下图是系统的示例：

图 5：基于文本的音频检索定位系统示例

如上图所示：用户的查询词是“爱”，系统在右侧给出了语料库中所有包含“爱”的歌词片段，选中之后点击 Play 就可以定位到相应的音频。实验表明其具有比较高的准确率，不过需要指出的是准确率和歌词本身的时间标注信息有关，标注信息越准确，则效果越好。

4 结论及未来工作

歌词是歌曲语义上的重要表达，如何利用歌词的信息并结合音频处理相关的技术提高音乐检索系统的效果是人们关注的研究方向。本文利用自然语言处理相关概念和技术，例如：齐夫定律，中文分词，向量空间模型表示，相似度计算等。实验表明：歌词语料库和通用语料库相比，在用字和用词上有所差别，但都基本符合齐夫定律。利用传统的文本分类概念并结合 k-近邻算法能找到具有相似度的歌词集合，这可以为歌曲的相似度计算提供依据。另外，我们还探讨了如何利用歌词中的时间标注信息对歌曲进行进一步的分析和检索，实验表明，平均每句歌词的“节奏”大约为 570ms/字，根据歌词总体的时间分布，我们对歌词的节奏进行分类，发现大约 68%的歌词节奏属于中速，11%的歌词节奏属于快速，而 21%的歌词节奏属于慢速。另外，我们还实现了一个基于文本的音频检索定位系统，用户通过输入查询词，可以方便的定位到相应的音频片段，实验表明，该系统具有较高的准确率。

进一步的工作包括：歌词和普通文本在用词等方面具有差异，如何对歌词进行特征抽取，使得相似度的计算具有更好的效果，能否为歌词定义类别，例如：流行音乐、校园民谣、儿歌等，并对其进行分类也是值得思考的问题。另外，如何自动的确定节奏划分的区间，使得其具有更合理的分类需要进一步的讨论。

参考文献

[1] S. Baumann and A. Kluter. Super-Convenience for Non-Musicians: Querying mp3 and the Semantic Web. In Proc. of the 3rd International Conference on Music Information Retrieval (ISMIR), Paris, France, 2002.

[2] A. Berenzweig, B. Logan, D. Ellis and B. Whitman, A large-scale evaluation of acoustic and subjective music similarity measures, Proc. of the ISMIR, 2003

[3] B. Logan, D. Ellis and A. Berenzweig, Towards Evaluation Techniques for Music Similarity, IEEE ICME, 2003

[4] J. P. G. Mahedro, A. Martinez, P. Cano, M. Koppenberger and F. Gouyon, Natural language processing of lyrics, in ACM Int. Conf. on Multimedia Proc., pp. 475-478, 2005.

[5] B. Logan, A. Kositsky and P. Moreno, Semantic analysis of song lyrics, IEEE ICME, 2004

[6] G.K.Zipf, Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology .Cambridge,Mass Addison-Wesley Press,INC,1949

[7] G. Salton, A.Wong. A vector space model for automatic indexing. Communications of the ACM, 18,613-620, 1975

[8] Tom Mitchell. Machine Learning. McGraw Hill, 1996

[9] D. Bainbridge, S.J. Cunningham and J.S. Downie, Analysis of queries to a Wizard-of-Oz MIR system: Challenging assumptions about what people really want, IEEE ICME, 2003

[10] M. Besson, F. Faita, I. Peretz, A.-M. Bonnel, and J. Requin, Singing in the brain: Independence of Lyrics and Tunes, Psychological Science, 494-498, 6, 9, 1998

[11] S. Scott and S.Matwin. Text Classification usingWordNet Hypernyms. In S. Harabagiu, editor, Use of WordNet in Natural Language Processing Systems: Proceedings of the Conference, pages 38–44. Association for Computational Linguistics, Somerset, New Jersey, 1998.

[12] Y.Yang and X.Liu 1999. A re-examination of text categorization methods. In the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.

[13] M. Bobrek, D.B. Koch, Music signal segmentation using tree-structured filter banks, J. Audio Eng. Soc. 46 (5) (May 1998) 412–427.