# 网易云音乐大数据系统构建

**作者**：王彬，陈香，林丽，谭学，杨磊，冯永

**单位**：重庆大学，计算机学院

**案例版权**：该案例归重庆大学计算机学院所有

**涉及的知识点**：网络爬虫、Hadoop 、Spark、图数据库、贝叶斯分类器、LSTM

**案例来源及案例真实性情况**：该案例来源于重庆大学计算机学院专业硕士（电子信息）课程《大数据架构与技术》中的学生精选汇编课程设计。

**摘要** 网易云音乐作为国内主流的在线音乐提供平台之一，拥有大量的入驻歌手、音乐资源与活跃用户，既为用户提供丰富多样的音乐资源，又发展出了极具特色的评论区，产生了大量的社交数据，深受青年用户的喜爱，并获得了“云村故事会”、“网抑云”等特色标签。基于网易云音乐的海量音乐数据与社交数据，使用大数据分析与挖掘技术，可得到以往的音乐流行趋势、歌手流行趋势、用户喜好等多个维度的信息，并在此基础上预测未来的发展情况。本案例以构建网易云音乐大数据系统为主题，使用爬取网易云音乐官方网站的数据构建数据集，结合数据挖掘、机器学习等相关方法进行评论情感分析以及对用户进行歌曲、歌手个性化推荐。本案例核心内容有：（1）使用Hadoop 和Spark搭建服务平台；（2）使用网络爬虫进行网易云音乐官方网站的数据爬取和存储；（3）使用SVM、贝叶斯模型、LSTM以及BERT等模型进行评论情感分析；（4）基于图结构数据对用户进行歌曲、歌手推荐。

**关键词**：音乐大数据分析、网络爬虫、Hadoop 、Spark、图数据库、机器学习、深度学习

## 1 引言

该教学案例来源于重庆大学计算机学院专业硕士（电子信息）课程《大数据架构与技术》中的学生精选汇编课程设计。该案例以构建网易云音乐大数据系统为具体问题，需引导学生进行的主要内容有：（1）使用Hadoop 和Spark搭建服务平台；（2）使用网络爬虫进行网易云音乐官方网站的数据爬取和存储；（3）使用SVM、贝叶斯模型、LSTM以及BERT等模型进行评论情感分析；（4）基于图结构数据对用户进行歌曲、歌手推荐。

## 2 背景介绍

近年来各种音乐软件层出不穷，网易云音乐作为国内主流的在线音乐提供平台之一，拥有大量的入驻歌手、音乐资源与活跃用户，既为用户提供丰富多样的音乐资源，又发展出了极具特色的评论区，产生了大量的社交数据，深受青年用户的喜爱，并获得了“云村故事会”、“网抑云”等特色标签。基于网易云音乐的海量音乐数据与社交数据，使用大数据分析与挖掘技术，可以得到以往的音乐流行趋势、歌手流行趋势、用户喜好等多个维度的信息，并在此基础上预测未来的发展情况。另外，基于网易云音乐的大量评论文本数据，使用自然语言处理技术，可以分析用户评论所包含的情感信息及“抑郁”属性，深入了解当代青年人的心理状态与需求，对广告个性化投放、情感类产品研发等实际应用场景都有重要价值，亦可以结合音乐大数据为用户进行歌曲推荐、歌手推荐、好友推荐等。因此，对网易云音乐中的大数据进行分析，是非常有研究价值的一个课题。在本案例中，我们选取网易云音乐大数据系统的构建作为主要研究问题，借助大数据分析与处理、数据挖掘、机器学习等相关方法，实现评论情感分析以及对用户进行歌曲、歌手个性化推荐。

## 3 内容

该案例的主要内容主要分为四个小节，分别为搭建Hadoop/Spark平台、网易云音乐官方网站的数据爬取和存储、评论情感分析以及基于图数据库中特定关系的匹配对用户进行歌曲、歌手推荐。

3.1 Hadoop/Spark介绍与平台搭建

3.1.1 Hadoop介绍

Hadoop是Apache软件基金会旗下的一个开源分布式计算平台，为用户提供了系统底层细节透明的分布式基础架构，在分布式环境下提供了海量数据的处理能力。Hadoop是基于Java开发的，具有优异跨平台特性，并可部署在廉价的计算机集群中，其核心是分布式文系统HDFS（Hadoop Distributed File System）和MapReduce。

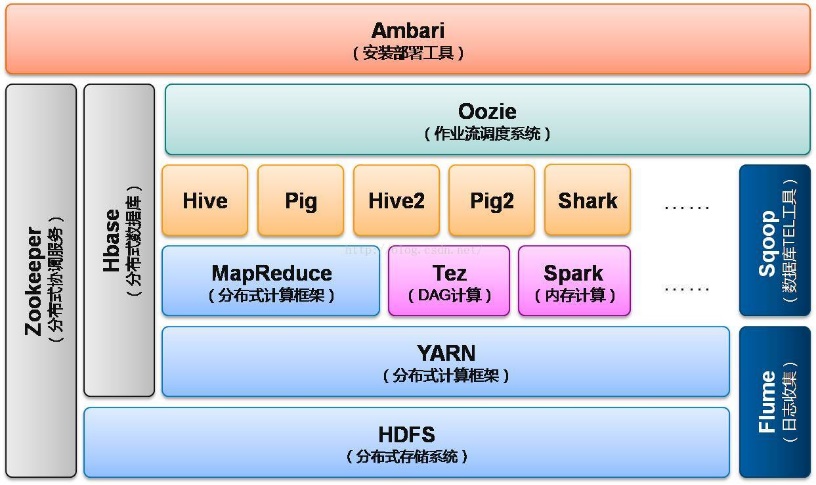
以下是Hadoop2.0时期架构，引入了YARN分布式框架等。

图1 Hadoop2.0时期架构

Hadoop是一个能够对大量数据进行分布式处理的软件框架，并且是以一种可靠、高效、可伸缩的方式进行处理的，它具有高效性，成本低，开源等特性。Hadoop凭借其突出的优势，已经在各个领域得到了广泛的应用，而互联网领域是其应用的主阵地。目前，几乎所有主流厂商围绕Hadoop提供开发工具、开源软件、商业化工具和服务。

3.1.2 Hadoop环境搭建

首先按照以下步骤创建两个服务器。

一、配置阿里源

1. 切换到yum仓库

*cd /etc/yum.repos.d/*

2．备份下原repo文件

*mv CentOS-Base.repo CentOS-Base.repo.backup*

3. 下载阿里云repo文件

*wget-O/etc/yum.repos.d/CentOS-7.repo http://mirrors.aliyun.com/repo/Centos-7.repo*

4. 设置默认的repo文件

*mv CentOS-7.repo CentOS-Base.repo*

5. 生成缓存

*yum clean all*

*yum makecache*

二、python安装

1. yum查找python3

*yum list python3*

2. yum 安装python3

*yum install python3.x86\_64*

三、配置hadoop用户

1. 创建用户

*useradd -m hadoop -s /bin/bash*

同时设置用户密码：123456

*passwd hadoop*

2. 配置权限

为了方便，给用户 hadoop 等同 root 权限：

*visudo*            # 执行 visudo命令进入vim编辑

*hadoop ALL=(ALL)* #添加配置权限

1. 配置SSH

1.生成密钥

*cd ~/.ssh/*        # 切换目录到ssh下

*ssh-keygen -t rsa*   # 生成密钥

2.授权

*cat id\_rsa.pub >> authorized\_keys*  # 加入授权

3.修改权限

*chmod 600 ./authorized\_keys*    # 修改文件权限

4.测试

*ssh localhost* # ssh登陆，不用输入密码直接登录成功则说明配置成功

五、配置Java环境

1. 安装 JDK

*sudo yum install java-1.8.0-openjdk java-1.8.0-openjdk-devel*

2. 配置环境变量

*vim ~/.bashrc*  # vim编辑配置文件

# 在文件后面添加如下单独一行（指向 JDK 的安装位置），并保存

*export JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk*

*source ~/.bashrc*  #是环境变量生效，执行

3. 测试

*echo $JAVA\_HOME*   # 检验变量值

*java –version*  #正确配置会输出java版本号

六、安装hadoop

1. 下载

*Wet –O hadoop-2.8.5.tar.gz*

*https://mirrors.cnnic.cn/apache/hadoop/common/hadoop2.8.5/hadoop-2.8.5.tar.gz*

2. 解压

*tar -zxf hadoop-2.8.5.tar.gz -C /usr/local*

3. 修改文件

*cd /usr/local/*   # 切换到解压目录下

*sudo mv ./hadoop-2.8.5/ ./hadoop*    # 将hadoop-2.8.5重命名hadoop

*sudo chown -R hadoop:hadoop ./hadoop*  # 修改文件权限

4. 测试

*cd /usr/local/hadoop*     # 切换到hadoop目录下

*./bin/hadoop version*   # 输出hadoop版本号

其次，进行Hadoop分布式集群搭建。

一、修改主机名

将两台服务器的主机名分别修改为master、 slave01。两台云服务器的内网IP、外网IP如下：

表3.1 两台云服务器内/外网IP分配

|  |  |  |
| --- | --- | --- |
| 主机名 | 内网IP | 外网IP |
| master | 172.17.58.185 | 47.103.193.58 |
| slave01 | 172.25.22.124 | 182.92.238.70 |

二、修改hosts

分别在两个服务器上运行shell命令sudo vim /etc/hosts，编辑hosts文件如下：

1. 在master上

127.0.0.1 localhost

172.17.58.185 master # master必须用内网IP

182.92.238.70 slave01 # slave01用外网IP

2. 在salve01上

127.0.0.1 localhost

47.103.193.58 master # master必须用外网IP

172.25.22.124 slave01 # slave01用内网IP

三、配置SSH

将master主机的 id\_rsa.pub 传递给slave01主机。

1.在 master 上scp传递公钥

*scp ~/.ssh/id\_rsa.pub hadoop@slave01:/home/hadoop/*

2.在slave01上加入验证

ls /home/hadoop/ # 查看master传送过来的 id\_rsa.pub文件

将master公钥加入免验证：

*cat /home/hadoop/id\_rsa.pub>>~/.ssh/authorized\_keys*

*rm /home/hadoop/id\_rsa.pub*

四、Hadoop集群配置

1. 切换目录

*cd /usr/local/hadoop/etc/hadoop*

2. 修改文件slaves

master主机作为NameNode, slave01作为DataNode

vim slaves

修改如下：*slave01*

1. 修改文件 core-site.xml

*<configuration>*

*<property>*

*<name>****hadoop.tmp.dir****</name>*

*<value>****file:/usr/local/hadoop/tmp****</value>*

*<description>****Abase for other temporary directories.****</description>*

*</property>*

*<property>*

*<name>****fs.defaultFS****</name>*

*<value>****hdfs://master:9000****</value>*

*</property>*

*</configuration>*

4.修改 hdfs-site.xml

*<configuration>*

*<property>*

*<name>****dfs.replication****</name>*

*<value>****2****</value>*

*</property>*

*</configuration>*

5.修改 mapred-site.xml.template

# 首先复制它产生一个新复制文件并命名为： mapred-site.xml

cp mapred-site.xml.template mapred-site.xml

# 然后修改文件 mapred-site.xml ：

*<configuration>*

*<property>*

*<name>****>mapreduce.framework.name****</name>*

*<value>****yarn****</value>*

*</property>*

*</configuration>*

6.修改 yarn-site.xml

*<configuration>*

*<property>*

*<name>****yarn.nodemanager.aux-services****</name>*

*<value>****mapreduce\_shuffle****</value>*

*</property>*

*<property>*

*<name>****yarn.resourcemanager.hostname****</name>*

*<value>****master****</value>*

*</property>*

*</configuration>*

7.复制 /usr/local/Hadoop 给其它节点

1. 在master节点上执行

*cd /usr/local/ rm -rf ./hadoop/tmp* # 删除临时文件

*rm -rf ./hadoop/logs/\**  # 删除日志文件

*tar -zcf ~/hadoop.master.tar.gz ./hadoop*

1. 将压缩好的文件传递给 slave01

*cd~*

*scp ./hadoop.master.tar.gz slave01:/home/Hadoop*

1. 在slave01节点上删除原有hadoop文件夹

# 删除原有hadoop文件夹

*sudo rm -rf /usr/local/hadoop/*

# 解压传过来的文件到指定目录 /usr/local ：

*sudo tar -zxf /home/hadoop/hadoop.master.tar.gz -C /usr/local*

# 设置权限解压出来的hadoop文件夹权限

*sudo chown -R hadoop /usr/local/hadoop*

最后，master上启动集群。

*cd /usr/local/hadoop*

*bin/hdfs namenode -format*

*sbin/start-all.sh*

3.1.3 Spark介绍

Spark是一个开源集群运算框架，最初是由加州大学柏克莱分校AMPLab所开发。相对于Hadoop的MapReduce会在运行完工作后将中介资料存放到磁盘中，Spark使用了存储器内运算技术，能在资料尚未写入硬盘时即在存储器内分析运算。Spark在存储器内运行程序的运算速度能做到比Hadoop MapReduce的运算速度快上100倍，即便是运行程序于硬盘时，Spark也能快上10倍速度。

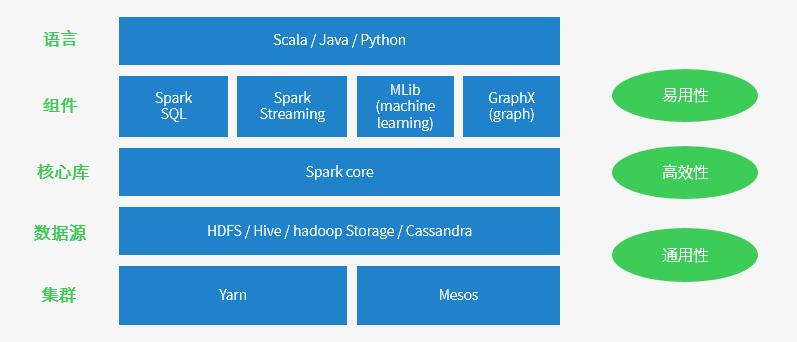
以下是Spark架构：

图2 Spark架构

Spark相对于Hadoop的优点在于：1、Spark编程灵活。计算模式不局限Map和Reduce操作，提供了多种数据集操作类型；2、Spark提供内存计算。可将中间结果放到内存中，对于迭代运算效率更高；3、Spark基于DAG的任务调度。要优于Hadoop MapReduce的迭代执行机制。实际应用中，大数据三种类型：批计算、流计算、交互式分析数据处理。Spark设计遵循“一个软件栈满足不同应用场景”，它既能提供内存计算框架，也支持SQL查询、实时流式计算、机器学习和图计算等，而且Spark可以部署在资源管理器YARN之上，提供一站式的大数据解决方案。

3.1.4 Spark环境搭建

基于Hadoop的环境搭建，本部分介绍Spark环境搭建的详细步骤。

一、准备工作

1.下载

*wget-Ospark-2.4.4-bin-without-hadoop.tgz <http://mirrors.tuna.tsinghua.edu.cn/apache/spark/spark-2.4.4/spark-2.4.4-bin-withouthadoop.tgz>*

2.解压

*sudo tar -zxf spark-2.4.4-bin-without-hadoop.tgz -C /usr/local*

3.设置权限

*cd /usr/local* # 切换到解压目录

*sudo mv ./spark-2.4.4-bin-without-hadoop ./spark*  # 重命名解压文件

*sudo chown -R hadoop:hadoop ./spark*  #设置用户hadoop为spark拥有者

4.配置spark环境

1). 先切换到 /usr/local/spark

*sudo cd /usr/local/spark sudo*

*cp ./conf/spark-env.sh.template ./conf/spark-env.sh*

2). 编辑 spark-env.sh 文件

*sudo vim ./conf/spark-env.sh*

*export SPARK\_DIST\_CLASSPATH=$(/usr/local/hadoop/bin/hadoop* classpath)

5.配置环境变量

1)．在 .bashrc 文件中添加如下内容

*export JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk*

*export HADOOP\_HOME=/usr/local/hadoop # hadoop安装位置*

*export SPARK\_HOME=/usr/local/spark*

*export PYTHONPATH=$SPARK\_HOME/python:$SPARK\_HOME/python/lib*

*/py4j-0.10.7- src.zip:$PYTHONPATH*

*export PYSPARK\_PYTHON=python3*

# 设置pyspark运行的python版本

*export PATH=$HADOOP\_HOME/bin:$SPARK\_HOME/bin:$PATH*

6.测试

*cd /usr/local/spark*

*bin/run-example SparkPi*

二、Spark分布式集群搭建

1.Spark配置

1). 切换配置目录

*cd /usr/local/spark/conf*

2). 配置 slaves 文件

*cp slaves.template slaves* # 先把模板文件复制重命名

# 开始编辑，将默认内容 localhost 替换为以下

*slave01*

3). 配置 spark-env.sh 文件

*cp spark-env.sh.template spark-env.sh*

# 编辑，添加下面内容

*vim spark-env.sh*

*export SPARK\_DIST\_CLASSPATH=$(/usr/local/hadoop/bin/hadoop classpath) export HADOOP\_CONF\_DIR=/usr/local/hadoop/etc/hadoop*

*export SPARK\_MASTER\_IP=172.17.58.185*  # 注意使用master内网IP

4).复制Spark文件到slave节点

*cd /usr/local/*

*tar -zcf ~/spark.master.tar.gz ./spark*

*cd ~*

*scp ./spark.master.tar.gz slave01:/home/hadoop*

5).节点替换文件

以下操作是在 slave01节点上：

*sudo rm -rf /usr/local/spark/*  # 删除节点原有Spark文件

*sudo tar -zxf /home/hadoop/spark.master.tar.gz -C /usr/local* # 解压到local *sudo chown -R hadoop /usr/local/spark*  # 设置spark文件权限拥有者

三、启动Spark集群

在master主机上执行以下操作。

1.先启动hadoop集群

*cd /usr/local/hadoop/*

*sbin/start-all.sh*

2.启动master节点

*cd /usr/local/spark/*

*sbin/start-master.sh*

3．启动slave01节点

*sbin/start-slaves.sh*

3.2 数据获取与存储

3.2.1 数据来源

1、网易云音乐官方网站

网易云音乐官方网站的网址为<https://music.163.com>，在web端提供了网易云音乐的所有功能。从网易云官方网站，我们可以得到歌手、歌曲、专辑、视频、电台、用户及其评论等多种信息。网易云音乐也在提供了Windows、Android及IOS等多平台的客户端，鉴于web端的数据爬取相对较为简单，我们选取网易云音乐官方网站作为数据来源之一。

2、网易云音乐API

“网易云音乐API”是Github上的个人开发者Binaryify的一个开源项目，提供了访问网易云平台各种数据的HTTP接口，项目所在地址为[https://github.com/Binaryify/Netease CloudMusicApi](https://github.com/Binaryify/Netease%20CloudMusicApi)。该项目基于Node.js开发，利用跨站请求伪造 (CSRF)，伪造请求头，调用网易云官方API，实现了对网易云

图3 网易云音乐官网网站首页

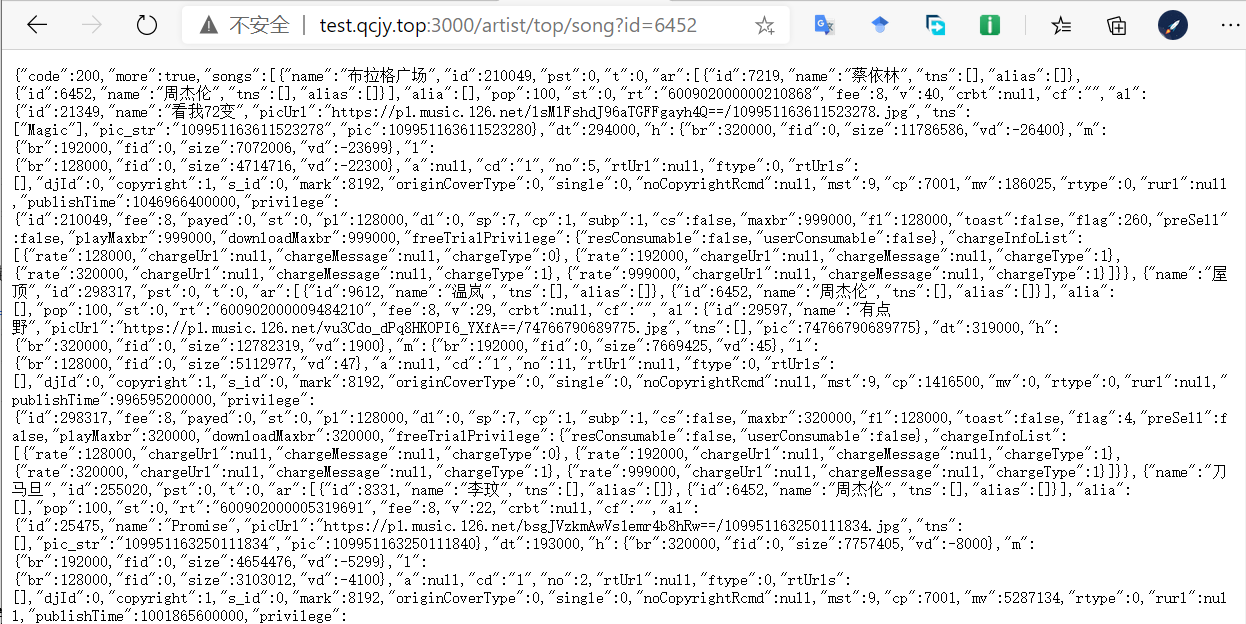
音乐官方服务的访问，提供了多达一百余个API接口，涵盖网易云音乐平台的大多数功能。网易云音乐API所返回的结果为格式化的JSON数据，相比较于网易云音乐网站，可以很大程序上减免数据提取与清洗的麻烦，因此我们选取网易云音乐API作为本课题的数据主要来源。

图4 网易云音乐API调用结果示例

3.2.2 数据爬取

1、爬虫技术概述

爬虫全称为网络爬虫（web clawler），通常指按规则自动从网页上提取信息的网络程序。网络信息不同于传统媒体信息，增长速度太快。如果用传统的简单检索来查找信息，由于其效率低下，会花费过多的时间，因此需要工具来提取用户所需信息，网络爬虫就是在这样的环境诞生的。

常用的爬虫有两类，一种是通用爬虫，另一种是聚焦爬虫。

通用爬虫是基于全网的，通常用于搜索引擎和大型网络服务的数据支持。它从一些种子URL出发，爬取整个互联网的资源。由于其爬取对象是整个互联网，所以需要大量的算法优化，以及庞大的计算资源。

聚焦爬虫则是定向抓取网页资源的。通常来说，聚焦爬虫只会从有限的网站中爬出数据，所以，它需要的计算资源不会很大。另外，聚焦爬虫不同于通用爬虫，它不是基于关键字之类的简单规则爬取，而是用户定制的高级规则爬取，这大大减少了无关页面的资源消耗。

2、Python模拟HTTP请求：urllib，requests

无论是从网页上爬取数据，还是通过调用API爬取数据，其中的一个重要步骤都是使用模拟HTTP请求来获取数据。urllib是python自带的用于处理HTTP请求的标准库，其功能全面但繁琐。requests是第三方开发的一个用于处理HTTP请求的库，其口号为“HTTP For Humans”，相对较urllib，requests库更加简洁和容易理解，同时也提供Keep-Alive、连接池、国际化域名、Cookie、SSL认证等高级特性，在网络爬虫中应用广泛。

3、Python解析HTML页面：Beautiful Soup

当我们需要从网页上爬取数据时，首先使用相关库模拟HTTP请求下载到网页的HTML文档，但并非整个HTML文档中的所有数据都是有用的，因此从HTML文档中解析出所需要的数据是一步重要操作。Beautiful Soup是一个可以从HTML或XML文档中提取数据的Python库。提取数据通常先读取HTML文档转化为BeautifulSoup对象，通过标签名、属性、内容等多种方式来筛选出所需的Tag，进而提取出网页中所需要保留的数据。

3.2.3 数据存储

1、Neo4j图数据库简介

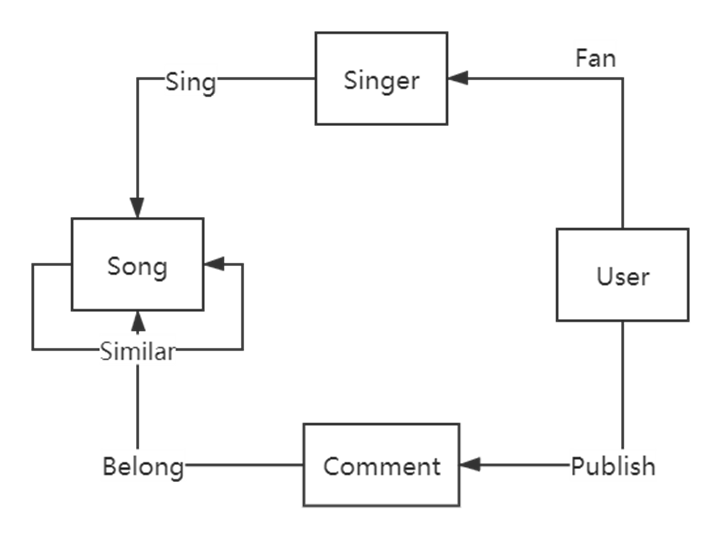
图是一种高级数据结构，包含了节点和边两种基本元素，通常节点用于表示现实中的实体，而边用于表示实体之间的各种关系。节点与边都可以有多种属性，在有向图中，边还具有方向，即代表一种单方向的关系。图数据库目前已被广泛应用于社交网络系统、实时推荐、地理空间管理、授权与访问控制及各类科学研究等多个领域。

Neo4j是一种本地的图数据库，可以灵活地存储数据与数据间的联系，并且保证极快的查询速度。Neo4j存储数据与关系更加的灵活与直观，节点与边的属性都是动态可变的，每个节点都存储指向其连接的所有节点的直接指针，可以更快地、更深入地执行具有复杂连接数量级的查询。而在传统的关系型数据库中，实体的属性通常在设计时就需要确定下来，而且实体之间的联系要被抽象为外键关联或者单独的联系表，在查询时需要进行频繁的多表连接操作，使用起来不够灵活。

Neo4j使用Cypher查询语言（CQL）用于操作数据库。Cypher是一种功能强大的图优化查询语言。在数据库中查询模式或关系时，Cypher查询比需要大量JOIN操作的SQL更为简单，更为容易编写。CQL基础的命令包括CREATE、MATCH、RETURN、DELETE等，这四种命令对应着SQL的CREATE、WHERE、SELECT、DELETE。CQL与SQL有很多相似之处，熟悉SQL的前提下可以很快上手CQL，但CQL是为图数据查询而生的，因此可以更好地表述图数据库中的模式及关系。

2、网易云音乐数据库

本次课题所构建出的网易云音乐数据库包含了4种实体（歌手、歌曲、评论、用户）以及实体间的关系共5中。实体及关系的结构如图2.3所示。数据库包含约500位歌手、2万首歌曲、210万条评论及120万个用户。原始数据已JSON格式存储于txt文件中，经过一定处理后存储至Neo4j图数据库中。图2.4为数据库的部分可视化展示。



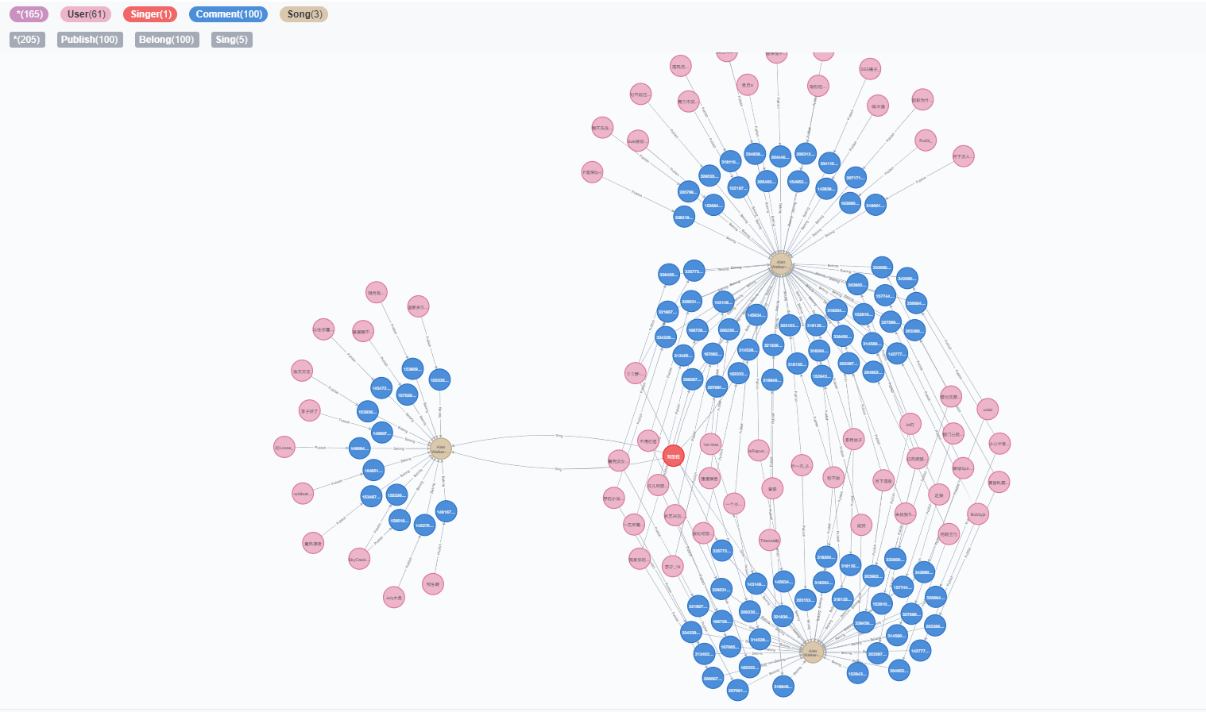
图5 网易云音乐数据库的逻辑结构

图6 网易云音乐图数据库部分可视化展示

3.3 评论情感分析

3.3.1 数据集

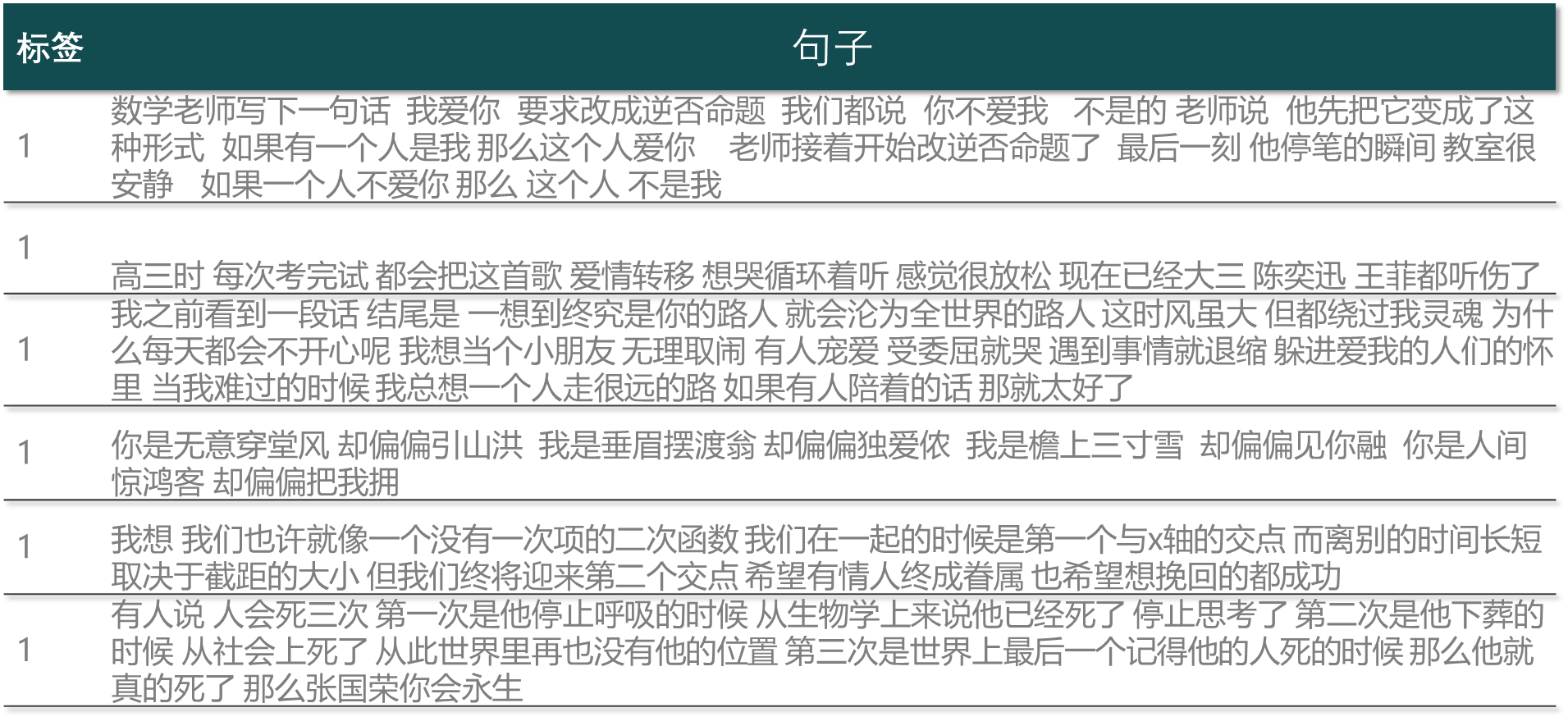
我们训练所使用的数据集是从网易云爬取的数据集，经过数据的清洗后从所有的评论中选择的5000条长度适中、感情色彩鲜明的句子，然后对这5000数据进行人工打标签。我们在数据集中把有关于情、爱、哭、死以表达自身情感的句子标记为1，其它的以阐述事实观点或平平无奇的句子标记为0。很明显，标记为1的句子是具有网易云风格的含有某种特殊情感的句子，而标记为0的句子就是没有明确表达情感的非网易云风格的句子。下图5.1和5.2是我们给数据集打标签的样例：

图7 数据集中标签为1的句子

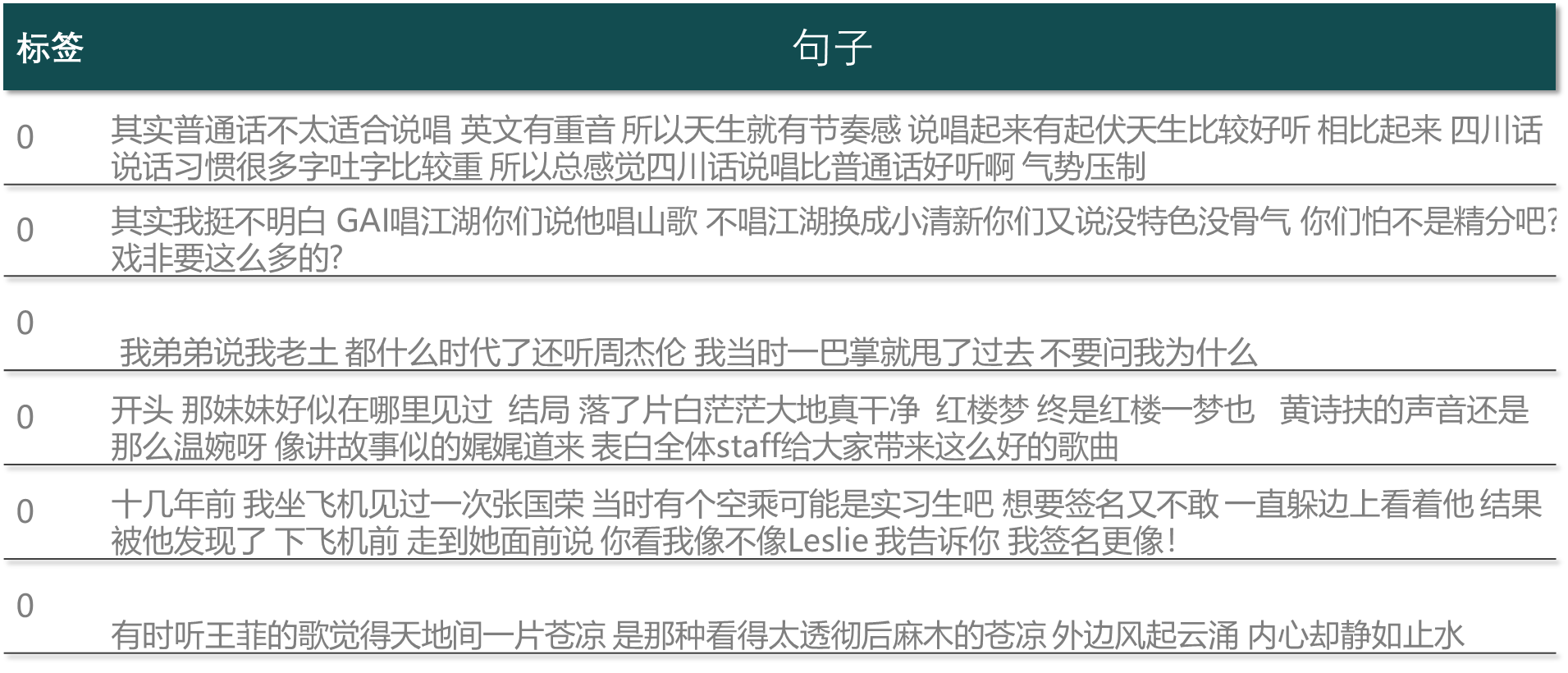
有了数据集之后，我们使用不同的特征提取的方法对文本数据进行特征提取，然后再使用不同的模型进行训练。我们使用不同的模型对数据集进行训练，通过对比分析选择准确率较高的模型来对所有网易云评论进行分析。

图8 数据集中标签为0的句子

3.3.2 情感分析的模型

网易云的情感分析实质上是对文本数据的一个二分类问题，只要能进行文本二分类的模型都能适用于我们的任务。在模型的选择上，我们选择了传统的机器学习模型和深度学习模型。分别是基于传统的模型支持向量机（SVM）、贝叶斯模型（Bayes）和基于深度神经网路的LSTM模型和bert模型，下面的四个小节我将分别介绍这几个模型。

1、支持向量机

支持向量机（support vector machines, SVM）是一种二分类模型，它的基本模型是定义在特征空间上的间隔最大的线性分类器，间隔最大使它有别于感知机；SVM还包括核技巧，这使它成为实质上的非线性分类器。SVM的学习策略就是间隔最大化，可形式化为一个求解凸二次规划的问题，也等价于正则化的合页损失函数的最小化问题。SVM的学习算法就是求解凸二次规划的最优化算法。

SVM学习的基本想法是求解能够正确划分训练数据集并且几何间隔最大的分离超平面。如下图所示， 即为分离超平面，对于线性可分的数据集来说，这样的超平面有无穷多个（即感知机），但是几何间隔最大的分离超平面却是唯一的。支持向量机的超平面划分如图5.3所示。

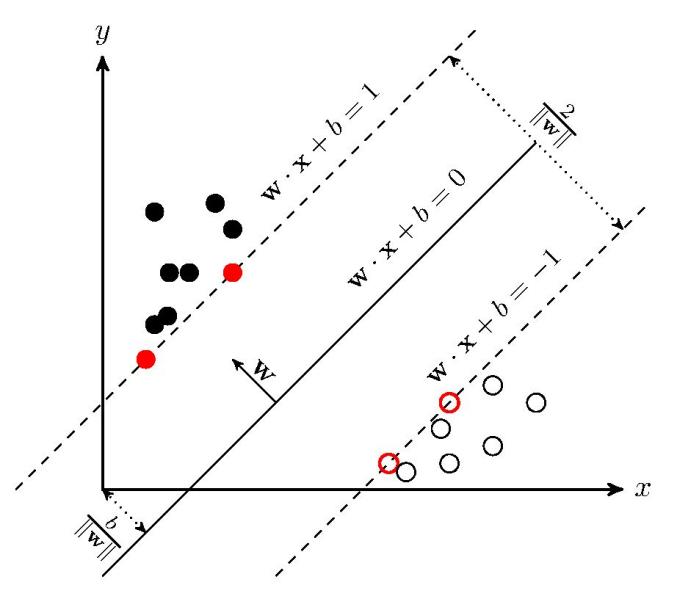
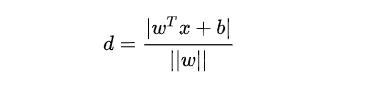
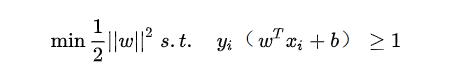


图9 SVM超平面划分示意图

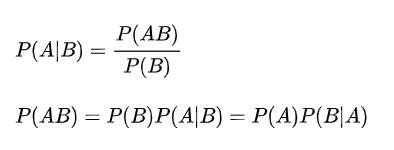
每个支持向量到超平面的距离：

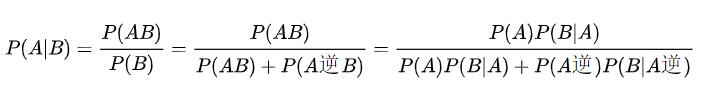
最后得到的最优化问题是：

2、贝叶斯模型

要理解贝叶斯，我们先来看看贝叶斯学派与频率主义学派的区别。简单说来，贝叶斯学派认为，概率是一个人对于一件事的信念强度，概率是主观的。但频率主义学派所持的是不同的观念：他们认为参数是客观存在的， 即使是未知的，但都是固定值，不会改变。

频率学派认为进行一定数量的重复实验后，如果出现某个现象的次数与总次数趋于某个值，那么这个比值就会倾向于固定。最简单的例子就是抛硬币了，在理想情况下，我们知道抛硬币正面朝上的概率会趋向于1/2。非常好理解不是么？但贝叶斯提出了一种截然不同的观念，他认为概率不应该这么简单的计算，而需要加入先验概率的考虑。先验概率也就是说，我们先设定一个假设（或信念，belief）。然后我们通过一定的实验来证明/推翻这个假设，这就是后验。随后，旧的后验会成为一个新的先验，如此重复下去。而归根结底，就得到了这样一个著名的公式：P( A | B ) = P( B | A ) \* P( A ) / P( B ) 。（A | B表示A给定B的概率，也就是说，如果B发生，A发生的可能性有多大。反之亦然。）

贝叶斯公式是在条件概率的公式上发展而来的，条件概率的公式为：

贝叶斯公式是条件概率公式的变型，而贝叶斯公式是整个贝叶斯预测模型行的核心：

对于贝叶斯模型，这里简单讨论一下模型的优缺点：

**朴素贝叶斯的主要优点有：**

（1）朴素贝叶斯模型发源于古典数学理论，有稳定的分类效率。

（2）对小规模的数据表现很好，能个处理多分类任务，适合增量式训练，尤其是数据量超出内存时，我们可以一批批的去增量训练。

（3）对缺失数据不太敏感，算法也比较简单，常用于文本分类。

**朴素贝叶斯的主要缺点有：**

（1） 理论上，朴素贝叶斯模型与其他分类方法相比具有最小的误差率。但是实际上并非总是如此，这是因为朴素贝叶斯模型给定输出类别的情况下,假设属性之间相互独立，这个假设在实际应用中往往是不成立的，在属性个数比较多或者属性之间相关性较大时，分类效果不好。而在属性相关性较小时，朴素贝叶斯性能最为良好。对于这一点，有半朴素贝叶斯之类的算法通过考虑部分关联性适度改进。

（2）需要知道先验概率，且先验概率很多时候取决于假设，假设的模型可以有很多种，因此在某些时候会由于假设的先验模型的原因导致预测效果不佳。

（3）由于我们是通过先验和数据来决定后验的概率从而决定分类，所以分类决策存在一定的错误率。

（4）对输入数据的表达形式很敏感。

通过以上分析，我们发现是可以使用朴素贝叶斯对我们的数据集进行文本分类的。最后我们的实验结果也表明，使用朴素贝叶斯模型进行预测的结果是稍好于使用支持向量机预测的结果的。

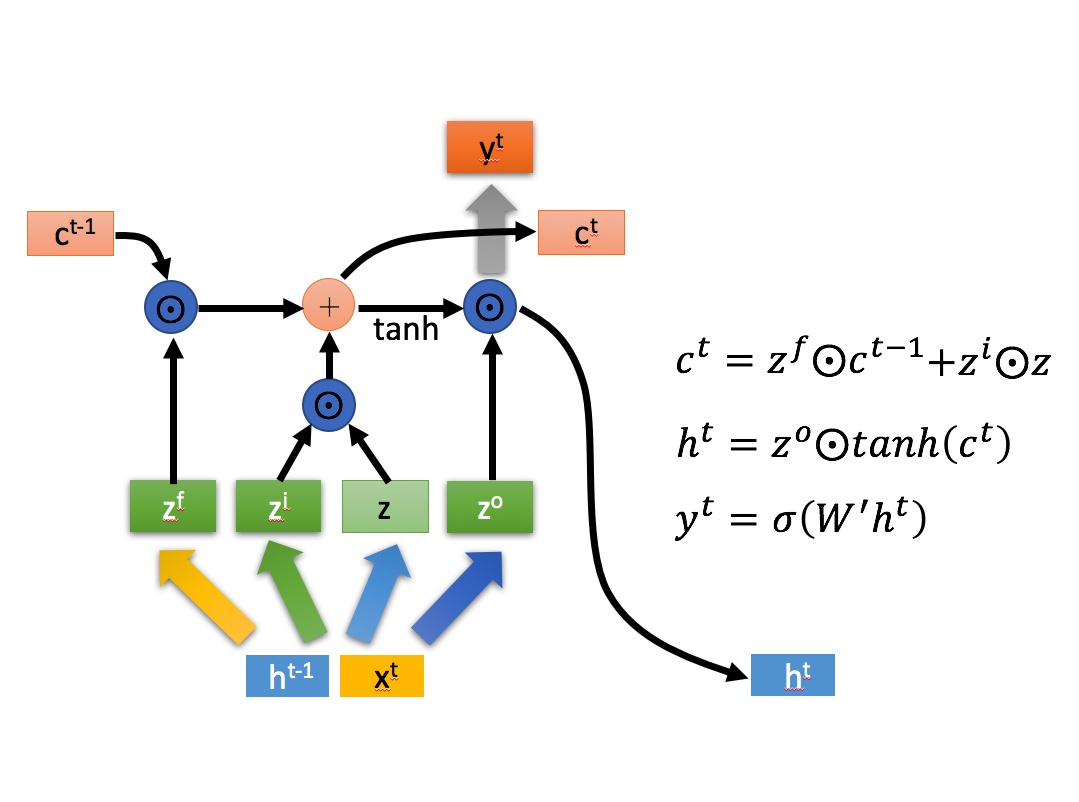
****3、LSTM模型

图9 LSTM的结构图

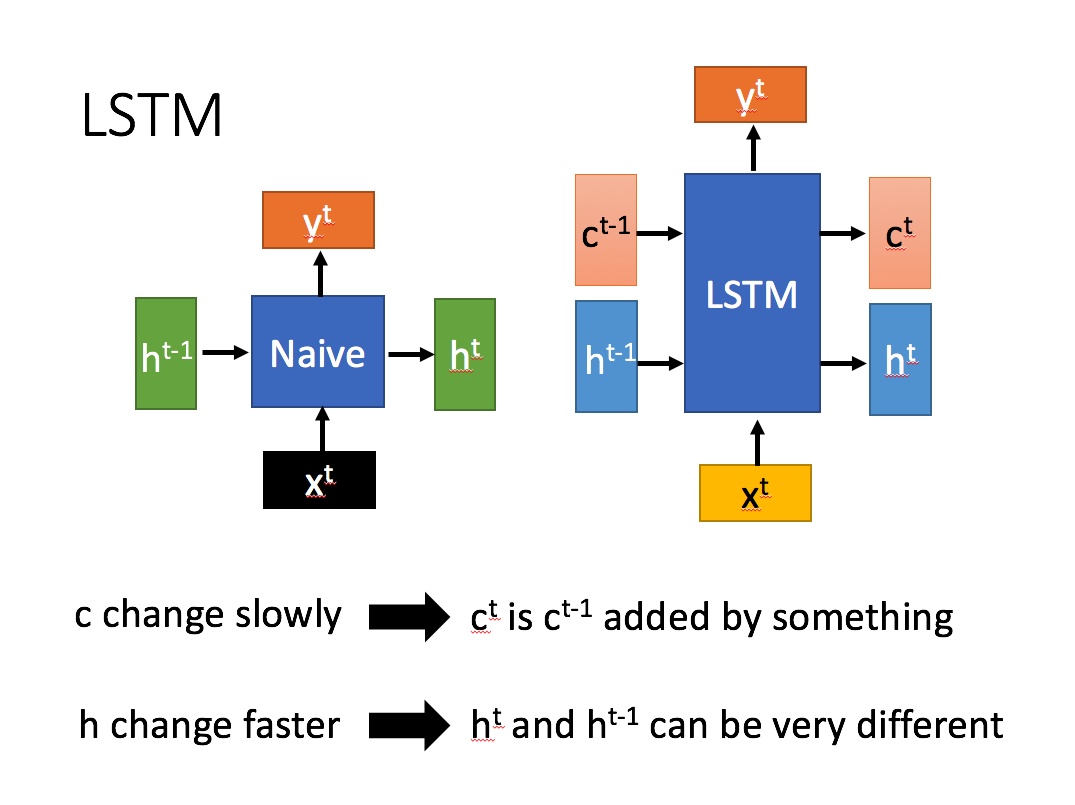
****LSTM的英文全称是Long Short-Term Memory，即长短期记忆。LSTM是一种时间循环网络，论文首次发表于1997年。由于独特的设计结构，LSTM适合于处理和预测时间序列中间隔和延迟非常长的重要事件。LSTM的表现通常比时间循环神经网络及隐马尔科夫模型（HMM）更好，比如用在不分段连续手写识别上。2009年，用LSTM构建的人工神经网络模型赢得过ICDAR手写识别比赛冠军。LSTM还普遍用于自主语音识别，2013年运用TIMIT自然演讲数据库达成17.7%错误率的纪录。作为非线性模型，LSTM可作为复杂的非线性单元用于构造更大型深度神经网络。

图 10 LSTM与传统RNN对比

LSTM结构（图右）和普通RNN（图左）的主要输入输出区别如下所示。

相比RNN只有一个传递状态 **ht** ，LSTM有两个传输状态，一个 **ct** （cell state），和一个 **ht**（hidden state）。（Tips：RNN中的 **ht** 对于LSTM中的 **ct** ）

其中对于传递下去的**ct**改变得很慢，通常输出的 **ct** 是上一个状态传过来的 **ct-1** 加上一些数值，而 **ht** 则在不同节点下往往会有很大的区别。

LSTM对于句子语义的获取是其它模型无法比拟的，特别是对单个句子语义上下文的获取。最后在我们的数据集上也取得了不错的效果，但是由于我们的数据集太少，使用LSTM训练的模型的表现不是很稳定。

4、BERT模型

BERT的英文全称是Bidirectional Encoder Representations from Transformers,翻译过来就是基于变换器的双向编码器表示技术，是雅各布·德夫林及其同事在2018年创建并发布了。Google正在利用BERT来更好地理解用户搜索语句的语义。BERTLARGE模型是一个有24层，每层网络有1024维隐藏层，16个自注意头，340M参数的神经网络结构。两者的训练语料都是BooksCorpus以及英语维基百科语料，单词量分别是8亿以及25亿。自BERT问世以来，已经刷新了多个自然语言处理任务的榜单。

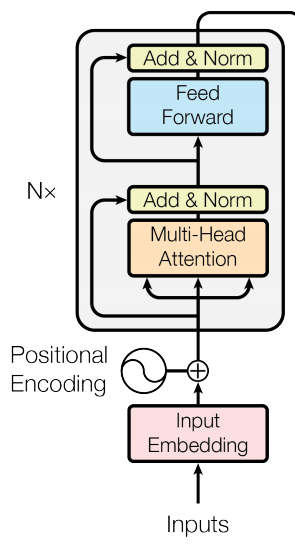
BERT是由多个Transformer结构组成的，在看BERT结构之前，我们先来看看Transformer。

图11 transformer结构

图11是Transformer encoder的结构，后文中我们都简称为Transformer。首先是输入word embedding，这里是直接输入一整句话的所有embedding。如图2.1所示，假设我们的输入是Thinking Machines，每个词对应一个embedding，就有2个embedding。输入embedding需要加上位置编码（Positional Encoding），为什么要加位置编码，后文会做详细介绍。然后经过一个Multi-Head Attention结构，这个结构是算法单元中最重要的部分，我们会在后边详细介绍。之后是做了一个shortcut的处理，就是把输入和输出按照对应位置加起来，如果了解残差网络（ResNet）的同学，会对这个结构比较熟悉，这个操作有利于加速训练。然后经过一个归一化normalization的操作。接着经过一个两层的全连接网络，最后同样是shortcut和normalization的操作。可以看到，除了Multi-Head Attention，都是常规操作，没有什么难理解的。这里需要注意的是，每个小模块的输入和输出向量，维度都是相等的，比如，Multi-Head Attention的输入和输出向量维度是相等的，否则无法进行shortcut的操作；Feed Forward的输入和输出向量维度也是相等的；最终的输出和输入向量维度也是相等的。但是Multi-Head Attention和Feed Forward内部，向量维度会发生变化。

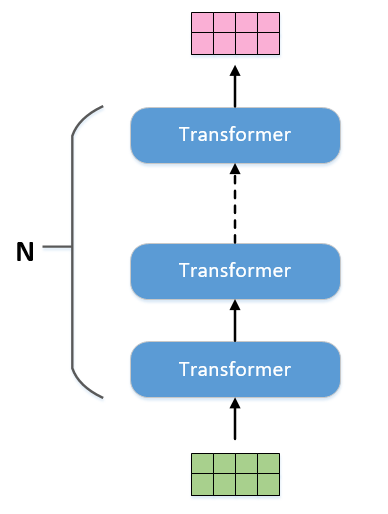
我们把图11作为一个基本单元，把N个这样的基本单元顺序连起来，就是BERT的算法模型，BERT的结构如图12所示。

图12 BERT结构图

BERT是在一个33亿单词的语料库上做预训练的模型，基本上适用于所有的文本语义获取的任务，如文本分类、知识问答、语义匹配等，但是不适合用于文本生成等任务。唯一的缺点就是BERT这个模型需要训练的参数多，训练的时间久。最后我们训练的BERT模型也取得了比支持向量机和贝叶斯模型较好的结果。

3.3.3 模型的训练

在不同模型的训练上方法大概都差不多，不同就是在数据集的处理上以及特征提取的方法。对于支持向量机模型的训练我们是在Hadoop+spark集群环境下完成的，对于贝叶斯模型的训练是使用sklearn包完成的，而对于LSTM和BERT的训练是使用主流的深度学习框架tensorflow2完成的。下面我将分别介绍这几个模型的训练过程。

1、SVM和贝叶斯模型的训练

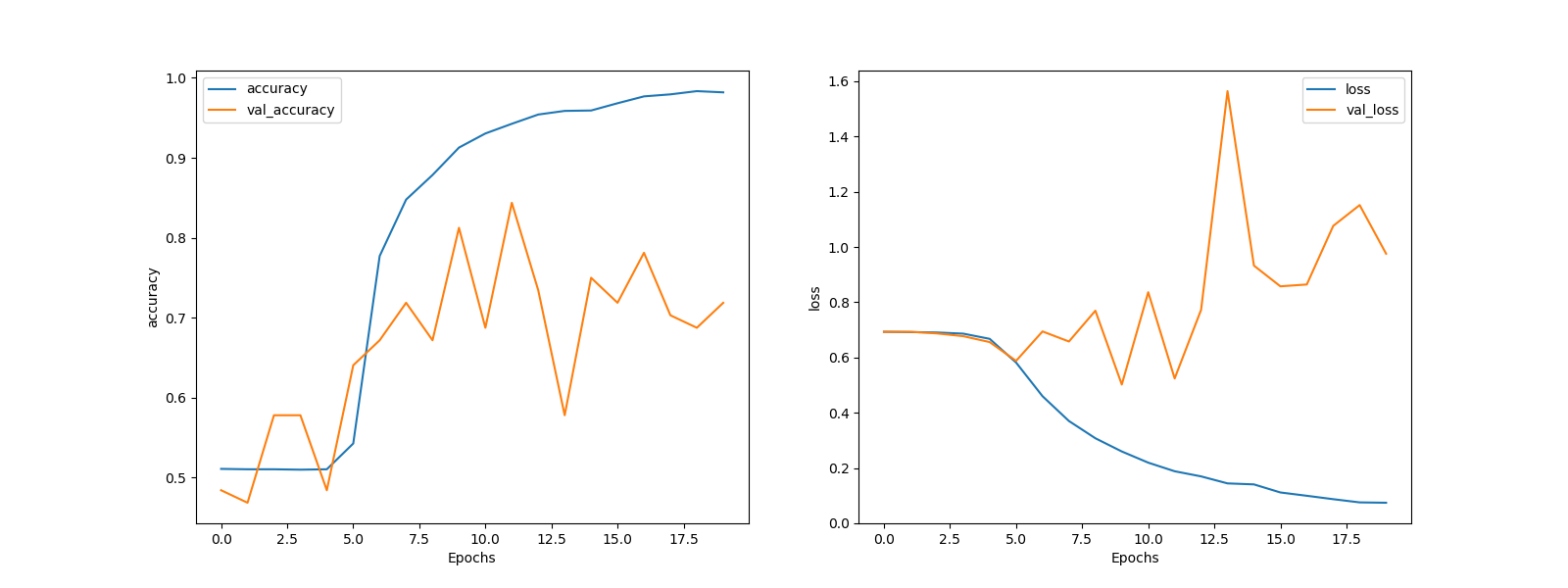
SVM的训练是在Hadoop+spark集群下完成的，首先我们使用sklearn的train\_test\_split模块对我们的数据集进行训练集和测试集的划分，选择的比例为80%用作训练集，20%用作测试集，在特征提取方面，我们使用的方法是doc2vec，也就是把一个句子直接生成一个向量。而在此次实现中是直接调用的gensim包里面的Doc2vec模块，对于模型的选择，我们使用的spark自带的SVMWithSGD模型。最终我们训练的SVM模型在测试集上的准确率是70.2%。

对贝叶斯模型的训练我们是sklearn库完成的，首先我们还是对sklearn的train\_test\_split模块对数据集进行训练集和测试集的划分，然后使用sklearn包里的TfidfVectorizer类对数据集进行特征提取，最后训练的结果如图5.8所示。可以发现，使用贝叶斯模型训练出来的结果比SVM的结果准确率提升了5个百分点左右。我觉得这是由两个方面造成的，第一个方面是在特征提取的方法上，直接使用doc2vec提取特征会损失掉句子的大部分特征，在特征提取方面doc2vec不如TfidfVectorizer；第二个方面就是贝叶斯模型比SVM更好的使用了我们文本分类的任务。

图13 贝叶斯模型训练的结果

2、LSTM和BERT模型的训练

LSTM和BERT模型我们都是tensorflow深度学习框架下实现的，这两种模型对于特征的提取都是使用的word embedding机制，也就是将每一个词映射成一个高维向量。在网络模型上我们使用了双层的双向LSTM架构，训练的过程如图14所示。

图14 LSTM模型中准确度和loss随训练轮数变化的情况

从图中我们可以分析出大概在第11轮的时候损失达到了最小，同时测试集上的准确度达到了最高，超过了80%，在超过第11轮的时候就产生了过拟合。LSTM的结果可以说是非常的令人满意，但是我们在多次训练中发现了使用LSTM模型训练的模型非常不稳定。有时候能达到80%以上的准确率，有时也只有75%左右，总体上表现还是不错。

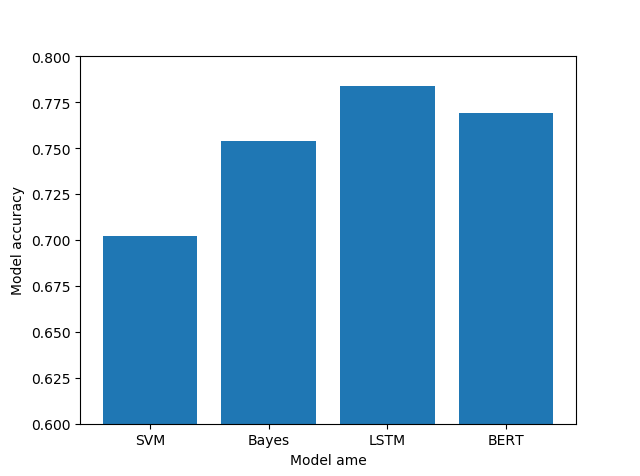
BERT模型我们是使用的tensorflow官方提供的使用中文维基百科语料库预训练的模型，然后再使用我们的数据集在预训练的模型上进行微调，最终在测试集上的准确率为76.9%，比使用的LSTM模型的准确率要稍微低一点，但是传统的机器学习效果要好。

图15 模型结果对比图

3、各个模型结果对比

图15展示了这四种模型在测试集上的准确率。从图中可以直观地看到，基于深度学习的模型在测试集上的准确率在整体上要高于基于传统的机器学习方法。在测试集上准确率的排名是LSTM、BERT、Bayes、SVM，其准确率分别是0.784, 0.769, 0.754, 0.702。

3.3.4 情感分析的可视化展示

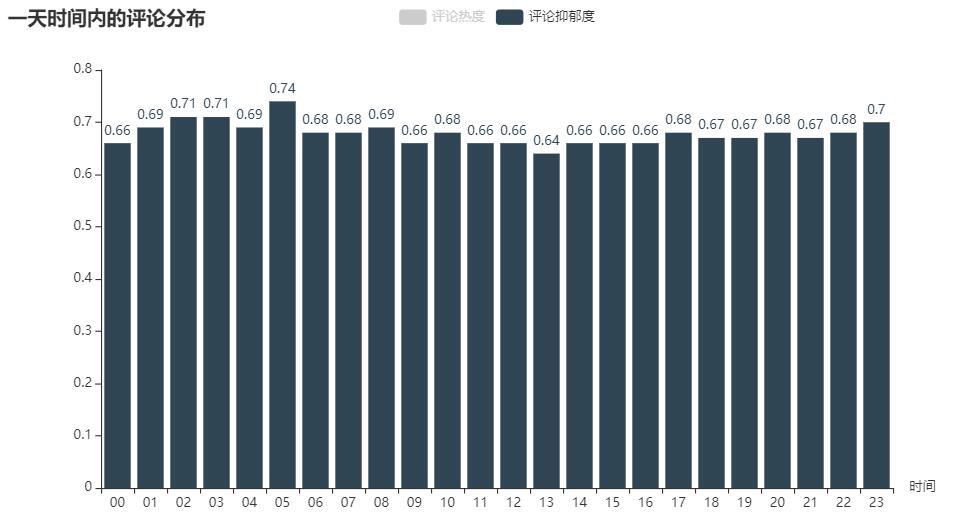
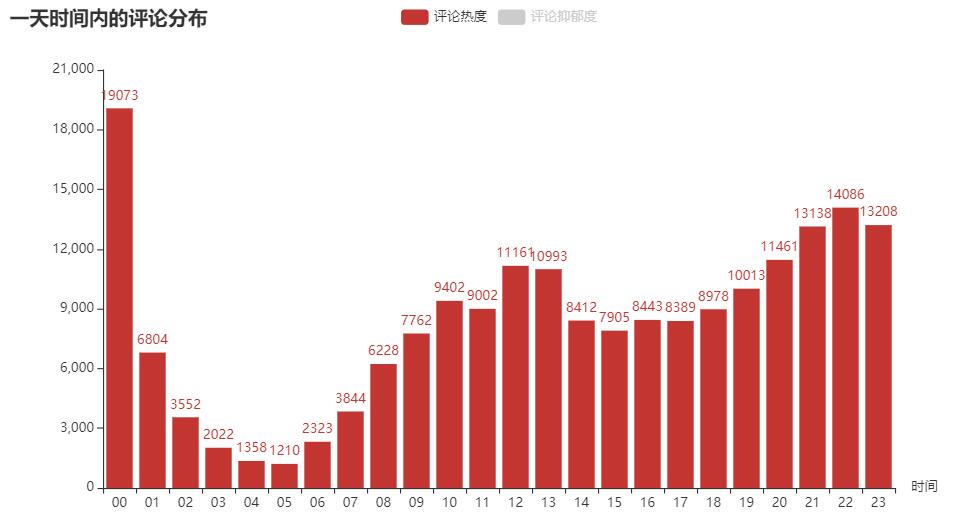


图16 一天中评论数量的变化，下为每个时间段中网易云风格评论占总评论数中的比例

在训练完所有的模型后，我们使用训练好的贝叶斯模型对网易云的评论进行情感分析。我们对网易云的情感分析分为三个方面。第一个方面是分析一天24小时每个时间段的评论数量以及情感变化；第二个方面是对400多个歌手进行情感分析，看哪个歌手的评论具有的网易云风格的比例最高；第三个方面是对10000多首歌曲进行评论分析，分析哪首歌最能激起网易云用户们进行网易云风格的评论（网易云风格评论即为数据集中标签为1的评论）。

3.4 基于图的推荐

1、基于图的推荐概述

图作为一种对关系具有很强的表达性的数据结构，十分适合用于推荐系统的数据载体。基于图的推荐方法有关系匹配、节点相似度计算、图嵌入以及图神经网络等。关系匹配通过匹配图中的特定关系模式来寻找所需要的节点，完成推荐任务。节点相似度计算通过计算各个节点之间的相似度来找到一批相似的节点作为推荐的目标，常见的相似度计算方法有Jaccard相似度、余弦相似度等。图嵌入是指将图数据转化为欧式空间的向量或向量机，然后使用各种基于欧式空间数据的算法来进行推荐任务。图神经网络将神经网络引入到了对图数据的处理，对图数据可以进行有效的特征提取，通常使用图神经网络可以有效地学习到每一个节点的嵌入表示，进而使用其它算法度量节点间的相似性或挖掘更深层的关系，从而完成推荐任务。

本项目中的推荐功能基于图中特定关系的匹配。

2、基于图数据库的歌曲、歌手推荐

基于当前关注的歌曲与歌曲之间的相似性，可以实现歌曲推荐的功能。假定需要推荐与“海阔天空”相关的歌曲，则使用如下的查询语句：

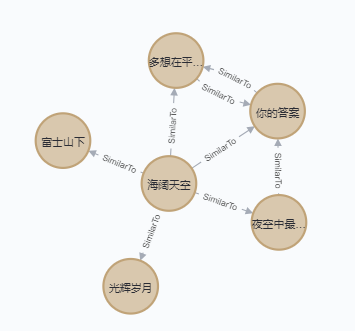
1. MATCH (s:Song)-[:SimilarTo]->(r:Song) **WHERE** s.**name**='海阔天空' **RETURN** r

图17 歌曲推荐可视化示例

图17为推荐结果的可视化。图中基于歌曲间的SimilarTo关系，得到了5首推荐的曲目。

由于在已构建的数据库中一首歌曲所包含的相似关系是有限的，可进一步基于歌手演唱歌曲的关系来进一步查找歌手及更多的推荐歌曲。

进行相似歌手及更多歌曲的推荐，可使用如下的查询语句：

1. MATCH (s:Song)-[:SimilarTo]->(song:Song)<-[:Sing]-(singer:Singer) WHERE s.name='海阔天空' RETURN singer
2. MATCH (s:Song)-[:SimilarTo]->(song:Song)<-[:Sing]-(singer:Singer)-[:Sing]->(moreSongs:Song) WHERE s.name='海阔天空' RETURN moreSongs

图18为推荐结果的部分可视化展示。这样的推荐相对来说还比较局限，可以基于歌曲热度，歌词相似性分析及评论的感情相似性等方面对推荐结果进行仅一部分优化。

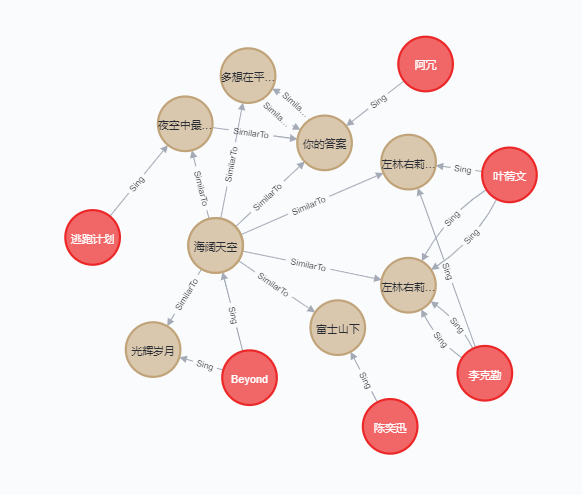


图18 歌手推荐及更多歌曲推荐

## 4 小结

本案例选取构建网易云音乐大数据系统为主题，使用爬取网易云音乐官方网站的数据构建数据集，结合数据挖掘、机器学习等相关方法进行评论情感分析以及对用户进行歌曲、歌手个性化推荐。主要内容包括搭建Hadoop/Spark平台、网易云音乐官方网站的数据爬取和存储、评论情感分析以及基于图数据库中特定关系的匹配对用户进行歌曲、歌手推荐。该案例主题新颖，结合了工业界现实需求与大数据分析与挖掘的多种理论与技术，可以充分增强学生的实践能力与理论基础。另外，本案例的内容仅为指导性的过程，在实际教学中，可保持基本研究内容不变，鼓励学生引入其它的数据预处理、数据挖掘、机器学习方法完成任务。

## 附录

1. 本案例提供配套的PPT、视频、数据集与代码等，发布于Github，链接为：https://github.com/Wanghui-Huang/CQU\_bigdata。

2. 本案例涉及到数据预处理以及多种机器学习算法，建议使用python语言进行编写，推荐的工具包有requests（网络请求库），BeautifulSoup（HTML解析库），scikit-learn（机器学习库）。

3. 本案例参考文献如下：

[1] 周志华. 机器学习[M]. 北京: 清华大学出版社, 2016.

[2] 阮泽楠. 音乐社交平台用户情绪特征研究[D]. 浙江理工大学,2019.