# 1 实习来源与目的

我自2015年1月加入到ACT实验室的车联网项目组，并参与到“车联网大数据处理平台”课题项目中。

实习目的包括：

（1）学习车联网相关背景知识，提高理论水平

（2）进行车联网相关实践，提升技术素养

（3）辅助推动项目顺利进行

# 2 实习内容

本次实习中的主要任务包括：

（1）学习车联网相关知识、技术与工具

大型车联网系统的整体架构涉及端、网、云三大部分。车载端通过安装OBD盒子采集各种原始数据，之后盒子通过实时网络传输将各种数据传到接收端。通常情况下，接收端有多级，每一极中也会有多个服务器。这其中综合运用了多种数据的采集、传输、接收、存储、挖掘、可视化等技术。对其中关键的知识、技术和工具需要有一个全面的了解。

（2）车联网日常行车数据统计与展现

在课题项目中，北航ACT实验室的车联网项目组维护了一个车辆的管理系统，该系统需要对某专车公司在全国范围中的20000多辆车进行存储管理。为了解每天车辆总体和各个车辆的运营情况，需要对车辆的日常行车数据进行统计入库，并在web端进行可视化展现。

（3）车联网订单数据计算与展现

对于专车公司来说，用户订单数据是一类重要的数据，订单直接关系到公司的收益。一种订单分派策略是由调度中心接受用户的订单请求，并就近分派空闲车辆，车辆完成服务后，也由调度中心安排指定备勤点。在这种策略中，需要对每个地理区域中空闲车辆与即将到来的订单数需要有一个良好的匹配。因此，需要通过历史订单数据设计预测算法，并将预测结果进行可视化展现。

此外，一个订单中车辆在订单外（未收到订单时）、空驶中（收到订单后尚未载客时）以及载客中的里程和油耗也是专车服务公司的关注点。尽快地载到乘客能够降低乘客取消订单的概率，给乘客更好的体验，而降低油耗则能够直接降低运营成本。因此，需要对车辆订单外、空驶中和载客中的里程和油耗进行统计和展现。

（4）车联网车载端应用开发

车载端数据是整个车联网系统中数据的入口点，在车联网系统中有十分重要的作用。车载端能够获得第一手的行车数据，因此能够保证良好的实时性并能够分担云端的计算压力。目前，以智能手机为代表的各种智能终端已经拥有很高的普及率，同时各种车联网的车载端应用对于车载端数据有着共性的需求。因此，在智能终端上设计一个数据分析系统有助于大型车联网应用收集车载端信息。

# 3 实习过程与表现

## 3.1 车联网基础学习

2015年1月我正式加入车联网小组同时也加入到课题项目中。刚进入课题项目组时，整个系统的基本规划已经完成，学长和学姐们已经完成了工具调研、系统搭建和性能测试的工作，系统中已有2000多的活跃车辆。在加入车联网小组的首次例会中，张明明师兄为我们介绍了北航车联网的生产环境业务与功能介绍。在介绍中，我了解到了目前北航车联网整体系统的搭建情况，也接触到了zookeeper，kafka，hbase，memcache，ruby on rails等工具。在接下来的3个多月（不算寒假），我的主要工作任务就是进一步了解整体系统的结构，熟悉相关工具的使用并搭建开发环境。

北航车联网的整体架构可以分为数据接收模块、实时计算模块、多级存储模块和前端展现模块。车载端的OBD设备采集到数据后，首先传给OBD设备制造商的服务器，由设备制造商的服务器在分两条链路，分别传给专车公司和北航的服务器。数据接收模块就负责对传过来的原始数据进行解包，并放入一个数据总线中。实时计算模块和多级存储模块是数据总线的主要消费者，实时计算模块不断取出数据总线并进行一些处理分析，将原始数据和分析结果交给多级存储模块进行缓存或持久化存储。前端展现模块对各种数据进行可视化展现。

北航车联网系统所用的数据处理工具包括zookeeper，kafka，jstorm，hbase，postgreSQL，memcache等。ZooKeeper（http://zookeeper.apache.org/）是一个分布式系统中的服务工具。在分布式应用系统中，协调维护各个节点的配置信息、节点命名、心跳监控和同步处理是一些复杂的、共性的需求。ZooKeeper就是为解决这些问题而诞生的，它提供了一些简单的操作，使得分布式应用可以基于这些接口实现诸如同步、配置维护和分集群或者命名的服务。很多分布式工具都用ZooKeeper作为分布式协调工具。Kafka（http://kafka.apache.org）是一个分布式的消息系统，在北航车联网系统中担任系统总线的职责，数据接收模块在对原始数据完成解包后会将数据直接放入kafka。JStorm是使用java重写的Storm流式计算框架，强调实时地处理流式数据，在北航车联网系统中，jstorm负责行程切分、车辆统计等功能。Hbase，PostgreSQL和memcache构成了多级存储系统，Hbase在HDFS上，提供高可靠性、高性能、列存储、可伸缩、实时读写的数据库系统，负责存储大量的历史数据。PostgreSQL 是一个自由的对象-关系数据库服务器（数据库管理系统），负责存储短期内（两周左右）的历史数据和一些需要经常查询的统计信息。Memcached 是一个高性能的分布式[内存](http://baike.baidu.com/view/1082.htm" \t "_blank)对象缓存系统，它通过在[内存](http://baike.baidu.com/view/1082.htm)中缓存数据和对象来减少读取[数据库](http://baike.baidu.com/view/1088.htm)的次数，从而提高动态、数据库驱动网站的速度。Memcached负责存储实时性要求高，并需要频繁查询的数据。在web端，使用tomcat提供HTTP接口查询服务，通过一个URL就能返回车辆信息；可视化展现则是通过ruby on rails实现。

在了解了车联网系统的整体结构和相关工具后，我开始尝试在自己的电脑上搭建开发环境并联系相关工具的使用。以上工具都提供了Java的开发包，在蒋文博师兄和王家兴学姐的指点下，我逐渐掌握了各个工具的使用方法并成功运行wordcount测试程序，期间协助蒋文博师兄测试并部署了hbase双机热备份机制，提高了系统的可用性。之后本来希望另外搭建一整套完整的系统，并将交通学院的数据导入我们的系统中，不过因为机器数量的原因没有实现。

## 3.2车联网日常行车数据统计与展现

2015年5月，沃老师给我和同组的张恿同学安排了车联网日常行车数据统计展现的任务。当时系统中已经有超过10000辆车，专车公司对的活跃车辆数，以及运行的里程、油耗，运行中事件比较关心，因此需要进行前端展现。具体的需求包括：（1）对车辆在一天中的总体数量，运营中的数量，运营的里程、油耗，运营中事件进行展现并显示一天中的变化情况。（2）对每天运营里程和运营油耗排名前10的车辆进行统计和展现。

为了实现这一需求，我和张恿对这个问题进行了分析，并确定了基本的解决思路。数据源方面，在SQL中有一个表负责存储每辆车的信息，每天更新，可以依据这个作为车辆总体数量，车辆停止运营时会停止向服务器发送数据，因此可以将过去几分钟内发送过数据的车辆看作运营中车辆。里程、油耗和事件信息在车辆的OBD信息中可以直接获得。数据处理方面，实时计算的部分由jstorm处理，统计结果暂存在memcached中，每天将当天的暂存结果入库，作为历史数据以供查询。确定了思路后，还有一些细节问题需要明确，如键值名称的设定，每日入库的时间，数据存储格式，数据库设计等问题。之后商定由张恿同学负责jstorm的部分，我负责存储与前端展现的部分。

数据源和数据处理的部分确定后，就需要开始考虑展现的效果。为了让展现更加生动，我采用了d3js（https://d3js.org/）和百度echarts（<http://echarts.baidu.com/）两种基于javascript的可视化工具，利用d3js>绘制进度条，利用echarts展示各类数据全天的变化情况。

## 3.3车联网车载端应用开发

2015年6月，实验室决定在智能终端上研发一款车载端应用。在之前的诸多工作中，车联网小组积累了丰富的车联网云端架构技术，而为了构建一个完整的车联网生态，车载端的地位也不可忽视。同时，在交通出行领域，人们最重视的需求是安全性。在这个背景下，实验室决定将车载端应用功能方向定位为提高交通出行安全。在经过几轮调研后，决定将这个车载端应用命名为“智汇行”，主要包含四个功能模块：（1）保驾护航，通过智能手机对司机行驶过程中的驾驶事件进行检测，对危险事件进行提醒，从而改善司机的驾驶行为，提高驾驶安全性。（2）我要练车，让用户按照系统指定的路线行驶，帮助新手司机熟悉道路环境。（3）PP教练，为新手司机和驾驶教练提供一个P2P平台。（4）信息板块，为用户提供一个交流论坛。在人员分配上，由张明明师兄牵头，我和兰天同学负责“保驾护航”和“我要练车”模块，王海轮同学负责“PP教练”和“信息板块”模块，张恿同学负责动画部分，张绍明同学负责服务端的搭建，吴迪同学负责美术设计相关的部分。

由于之前项目组中没有手机端APP的开发经验，一切都要从0开始。我们决定将Android定为开发环境，Android官方网站（https://developer.android.com）上提供了很多有用的学习资料。经过一个月左右的学习，我搭建起了Android开发的环境，掌握了Android开发中的各种基本概念，并了解了传感器模块和GPS模块的使用方法。同时，我还开发了一个简单的数据记录APP，这个APP会不断记录传感器和GPS数值，并在界面上提供了若干表示各类驾驶事件的按钮，用户点击按钮时APP会记录这个事件，最后通过事件和传感器数值的对比，可以发现事件发生时的数据变化情况。

事件检测是“保驾护航”和“我要练车”模块的核心，在事件检测方面，需要用到加速度传感器、陀螺仪传感器、旋转矢量传感器和GPS。加速度传感器能够感知震动、急加减速和启动停止事件；陀螺仪传感器能够感知转弯、偏转、掉头等事件；旋转矢量传感器能够将手机坐标转化为地球坐标，从而简化分析；GPS能够反映车辆行驶的速度和方向。在检测方法方面，我调研了近年来的相关工作，最后决定采用基于阈值和规则的检测方法，即对每一类事件设定一个阈值，当传感器数值超过这个阈值就认为事件发生。

在主要实现思路确定之后，我对手机端系统的整体结构进行了初步设计，并完整了一部分后台功能的开发。在用户界面部分，也针对每一个功能点设置了相应的操作按钮，虽然界面看起来简陋，不过大体的框架已经具备了。在暑假结束之后的一段时间，由于实验室还有其他的一些事务需要处理，“智汇行”的开发进度有所减缓，不过我依然还在对其进行更新和完善。并逐步加入传感器校准、数据缺失处理、阈值自适应等功能，并引入第三方语音和地图工具，使应用的功能更加丰富。

## 3.4车联网订单数据处理与展现

2015年9月，专车公司决定将订单数据也交给北航进行分析和处理，目标是希望能够根据历史数据更准确的预测某一个区域在未来一段时间的订单数量，从而更高效地调度空闲车辆，避免在一个地区行程有单无车或有车无单的情况。订单预测的算法由组里的王媛冬、魏华和张恿同学负责，我负责订单数据的入库以及一部分前端展现的工作。

首先需要解决的是订单入库的问题，专车公司给出的是包含一个月的订单数据文本文件，为了保证之后预测工作的正确性，需要保证订单入库不重，不漏。专车公司给出的订单数据有一些数据质量问题，使入库工作的难度有所增加，这些问题包括：（1）专车公司的订单数据文件基本上是每月给我们一次，因此是按月份划分的，不过有时是按订单创建时间，有时是按订单更新的时间，这就导致了部分订单重复。（2）有部分订单数据文件合并了几个月的订单数据。（3）为保证订单完整入库，需要按月对订单数量的一致性进行检查，上述问题也导致检查难以进行。为了解决上述问题，我拟定了如下方案：首先按照订单id和订单创建时间中的月份将一个大文件中的订单hash到若干小文件中，之后对小文件中的订单进行排序与去重，最后将小文件中的数据导入数据库。完成入库后，以月份为单位比较文件中的订单数量和数据库中的订单数量，保证了数据的完整性。

在展现方面，我设计了三种检索方式，分别是某一区域从开始时间到结束时间中每隔15分钟的详细变化；某一区域的某一时段，从开始日期到结束日期中每天的订单数量变化；某一区域中，从0点到24点，从开始日期到结束日期的平均订单数量变化。此外，用户可以根据查询需要添加一个新的区域，进行对比。通过以上查询方式，可以满足多样化的查询需求。

除了订单数量和预测数量的信息之外，专车公司还对车辆的订单外的里程油耗、空驶的里程油耗和有效的里程油耗感兴趣。专车公司希望尽量提高有效油耗所占的比例，并减少空驶里程从而减少乘客等待时间，为此专车公司希望能在采用一定的调度策略后，看到前后的对比效果，因此需要进行可视化展现。

为满足这个需求，首先还是要考虑数据的问题。订单数据中包含了收到订单时间，乘客上车时间和实际到达时间，可以据此划分出订单外时段，空驶时段和有效载客时段的，但里程油耗数据无法在订单数据获得，只能通过Hbase中的OBD数据获取，但Hbase中只存了车辆使用的设备号，订单中则是车牌号，还需要通过SQL数据库中的车辆信息表将二者对应起来。最终的解决方案为：对于车辆信息表中的每一辆车，以天为单位，统计该车当天的每一个订单，然后获得该订单的时间区间，通过Hbase查询到车辆的里程和油耗值，并将统计结果存入数据库。

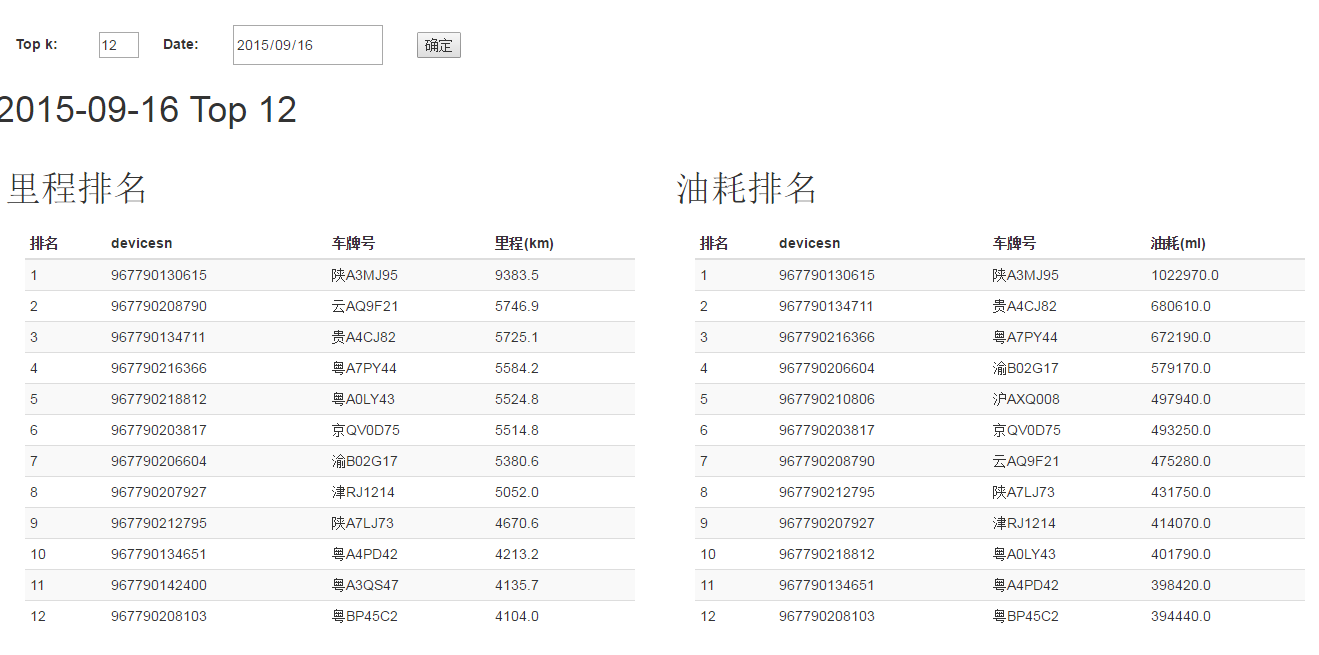
在前端展现方面，我设计将订单外、空驶中和有效里程和油耗用不同颜色显示在同一个横向的条形图中，从而能直观上看到三段所占的比例。用户可以通过按车牌号查询，也可按地区查询，按地区查询就是将同一个地区所有车辆的数据进行累加。按车辆查询时，如果点击某一天的条形图，则可以在地图上显示当天每一条订单的详细情况，便于用户进行分析。

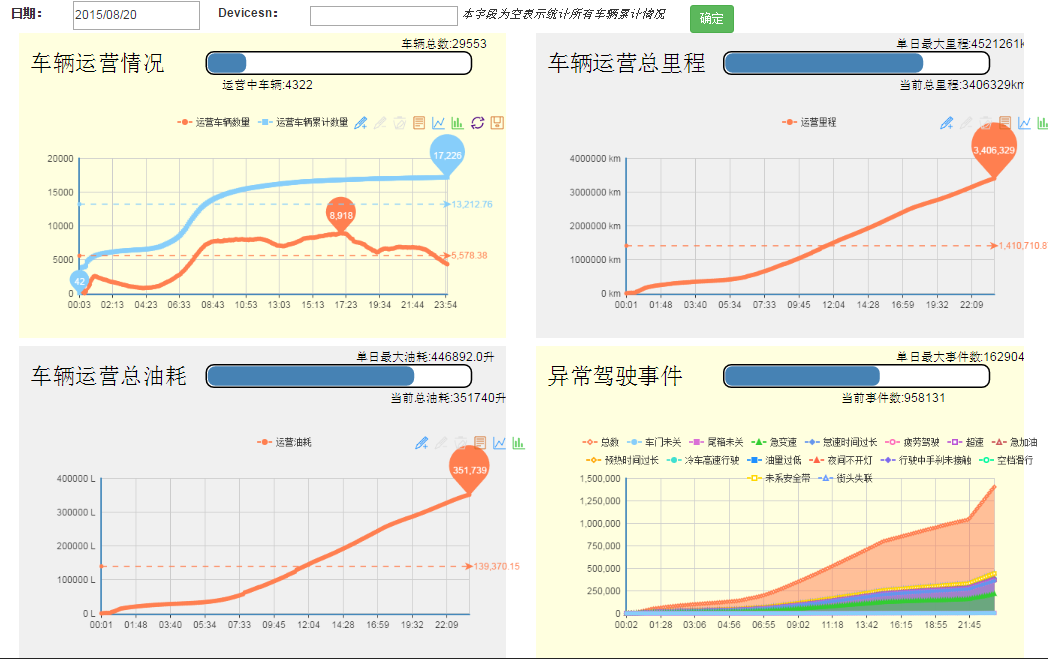
## 3.5实习总体表现

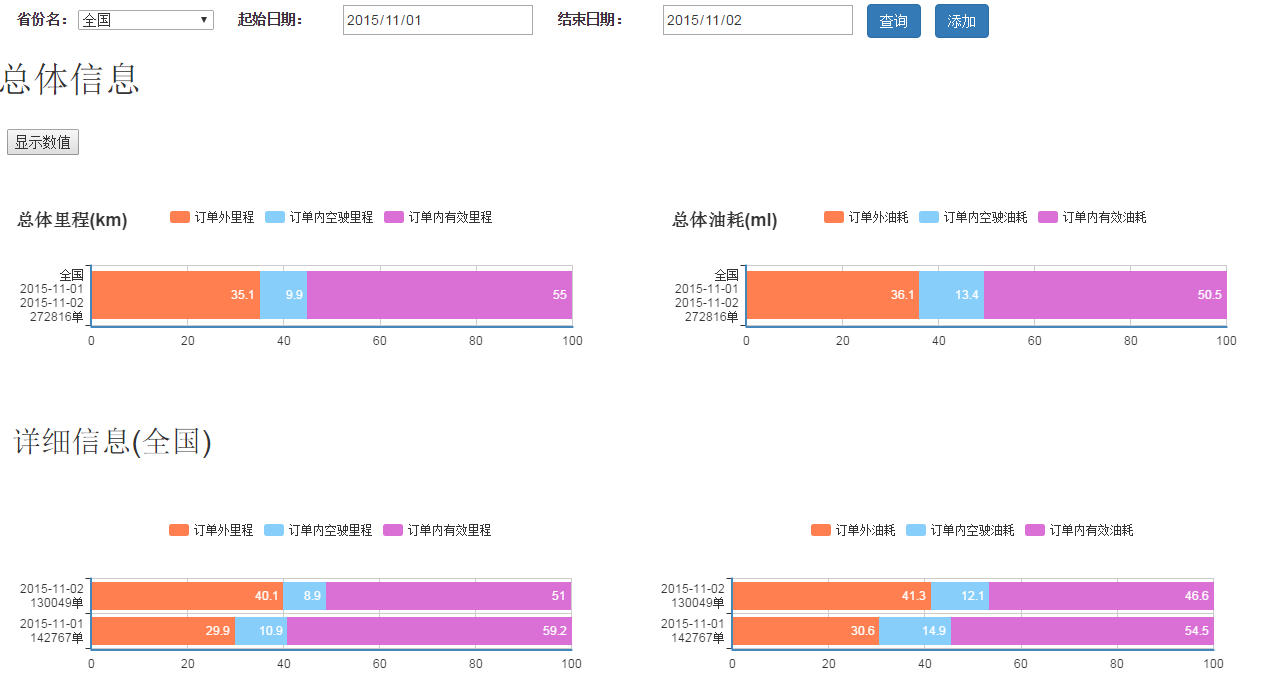
# 4 实习成果

在实习期间，我的主要工作包括车联网服务端运营数据处理与可视化展现、订单数据的处理与可视化展现和车载智能终端应用。实习成果的可展现的形式包括前端页面

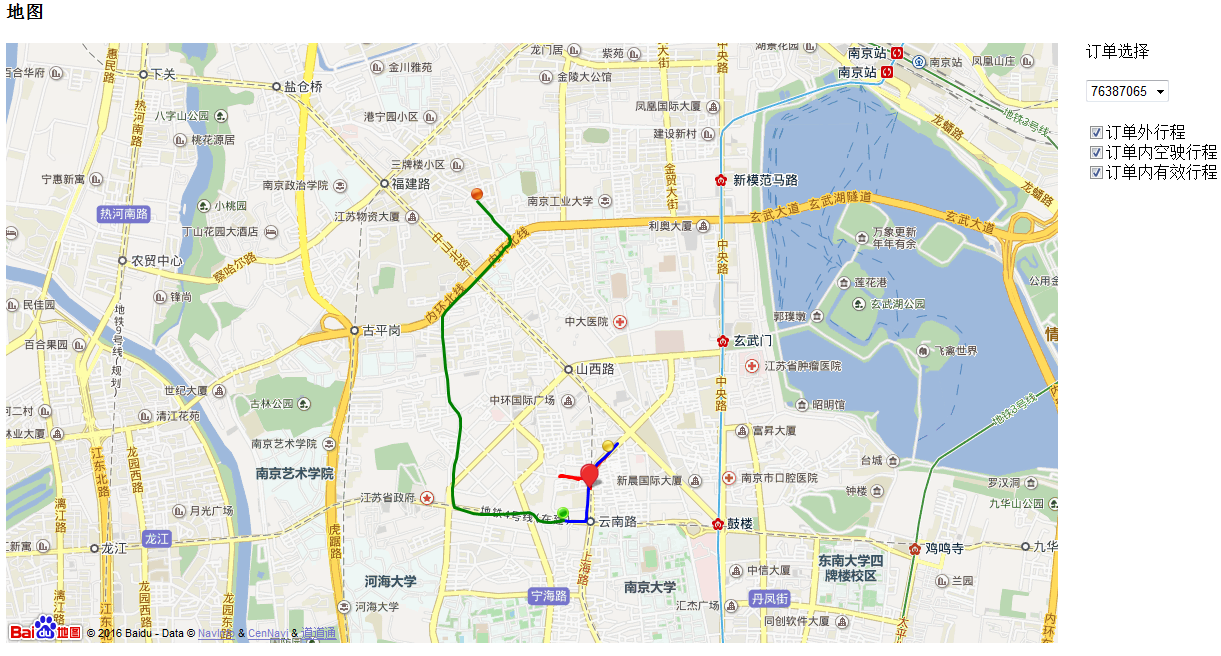
## 4.1 车联网展现页面展示











## 4.2 车载端应用展示

## 4.3 其他成果

（1）创新奖

（2）专利

# 5 收获体会

通过这次实习，我

（5）对车联网前沿技术有了一定的认识

（1）了解了如何构建一个完整的车联网系统

（2）在数据处理（尤其是存储方面）和前端展现方面有了丰富的实践

（3）在Android开发方面有了丰富的实践

（4）了解到了团队合作的乐趣和重要性

（6）

我要感谢