顺序号（硕）：SS020402
姓名：高新院
学号：'21100211003
学院：信息科学与工程学院
专业：地图学与地理信息系

中国海洋大学
OCEAN UNIVERSITY OF CHINA

硕士学位论文

MASTER DISSERTATION

基于空间位置信息的多源 POI 数据融合

论文题目：问题的研究

英文题目：Study on fusion of multi-source POI based on the spatial location information

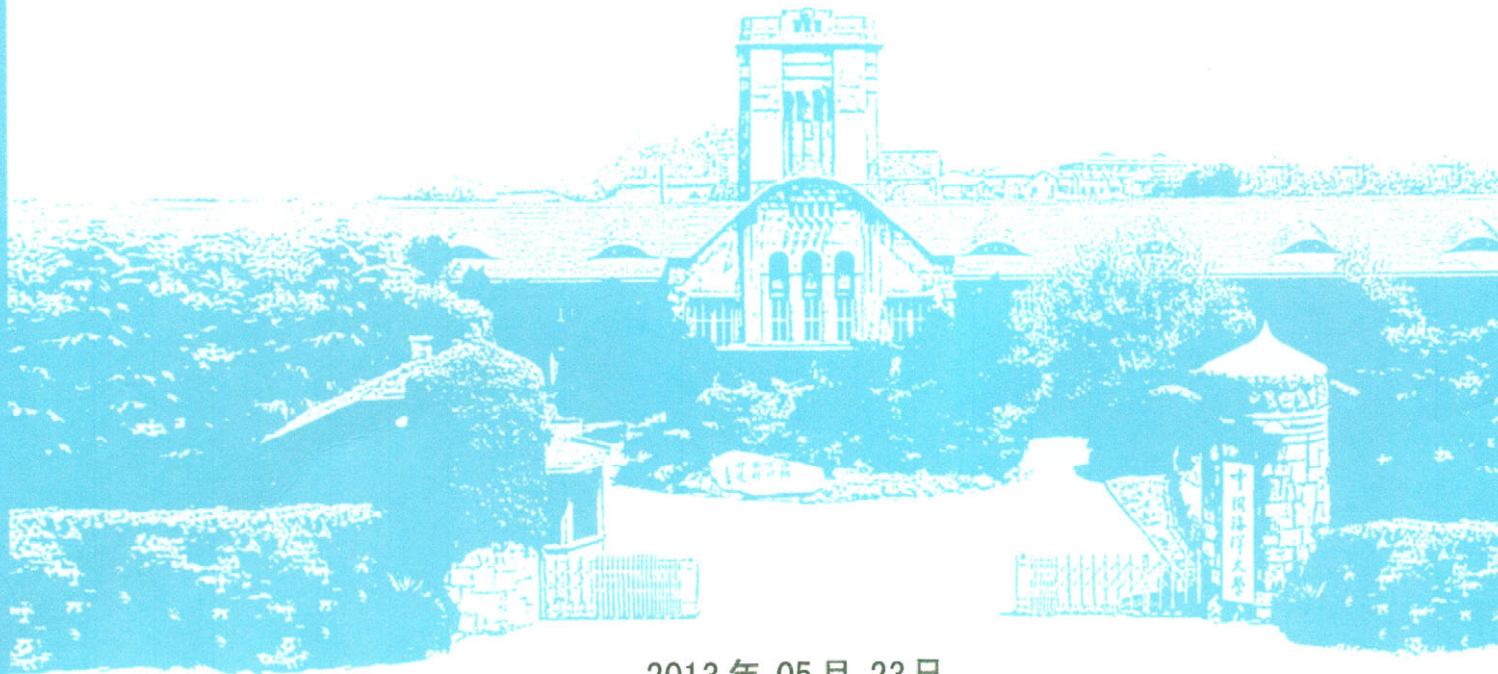
作　　者：高新院

指导教师：张巍　副教授

学位类别：全日制学术学位

专业名称：地图学与地理信息系统

研究方向：海洋信息分布式处理技术



2013 年 05 月 23 日

基于空间位置信息的多源 POI 数据融合问题的研究

学位论文答辩日期：2013.5.23

指导教师签字：

答辩委员会成员签字：

独 创 声 明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含未获得（注：如没有其他需要特别声明的，本栏可空）或其他教育机构的学位或证书使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名：高新院 签字日期：2013年 5月 27日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权学校可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。同时授权中国科学技术信息研究所将本学位论文收录到《中国学位论文全文数据库》，并通过网络向社会公众提供信息服务。（保密的学位论文在解密后适用本授权书）

学位论文作者签名：高新院 导师签字：

签字日期：2013年 5月 27日 签字日期：2013年 5 月 27日

知识产权保护协议

依据《中华人民共和国促进科技成果转化法》第二十八条和《中国海洋大学知识产权管理暂行规定（2004.7.20）》的有关规定，研究生高新院（以下简称研究生）与其导师张巍（以下简称导师）就知识产权保护事宜达成如下协议：

1、研究生在校期间从事科研工作所完成的学位论文以及不论是否写入学位论文的其他成果属职务成果。研究生不得对上述职务成果以自己或他人名义擅自向第三方转让或泄漏。

2、研究生离校后三年内，不得擅自将在校期间从事科研工作的相关数据、研究结果和相关技术发表论文，不得擅自向第三方转让或泄漏。

3、研究生离校后三年内，若进行重复及延续在校研究课题的科技项目，必须经导师及中国海洋大学同意并协商知识产权分享事宜后，方可开展工作。

4、若研究生违反上述规定，导师及中国海洋大学有权追究其法律责任，即：要求其停止侵权行为、公开消除影响并予以经济赔偿。

5、本协议双方签字之日起生效，有效期三年。

研究生（签字）：高新院　　导师（签字）：张巍

2013 年 5 月 10 日　　院系盖章：

2013 年 6 月 30 日

谨以此文献给尊敬的张巍副教授以及我亲爱的朋友和同学们！

----------高新院

本论文得到国家自然科学基金 **60602017**（**No. 60602017**）和山东省自然科学基金（**No. ZR2012FM016**）资助。

基于空间位置信息的多源 POI 数据融合问题的研究[1]

摘　要

伴随着网络电子地图与基于位置服务(LBS)的快速发展，以 POI 为代表的空间地理数据出现了快速增长。POI 是兴趣点(Point of interest)的缩写，是一种代表真实地理实体的点状数据，POI 一般包含名称、类别、经纬度以及地址等基本信息。一方面，POI 信息的搜集、存储以及更新需要花费大量的人力、物力，并且 POI 信息的及时添加和更新服务已经成为各个网络地图的核心竞争力；另一方面，不同来源的 POI 信息完善和丰富程度各有不同。如何把来源不同的 POI 信息进行集成融合从而实现数据复用，已成为急需解决的问题。

POI 数据融合技术是解决地理空间数据复用的关键技术，本文所提到的 POI 数据融合技术最终目标是：将两个 POI 数据集合中表示同一个地理实体的 POI 对象标识出来，并将它们放在“融合集”中。国外研究者提出的解决方案有大致如下：基于 Ontology 的技术；基于空间位置的技术；基于非空间属性的技术。为从两个来源不同的 POI 数据集合中准确找出用于融合的对应对象，本文在国外研究成果的基础上提出一种改进方案，该方案在空间位置属性的基础上利用非空间属性相似度来提高结果融合集的准确性。本文的具体研究工作与研究成果如下：

首先，对两个不同来源的 POI 数据集合实施空间位置技术找出对应对象组成的初步融合集，基于位置方法的优点是它仅仅根据经纬度位置信息就可以找对应对象，而经纬度信息是每个 POI 都必须具备的，不存在数据缺失问题；缺点是来源不同的 POI 的经纬度都普遍存在误差与坐标系不统一的问题。

其次，使用低阈值的名称属性相似度算法排除由空间位置方法找出的错误对应对象。该算法的优点是它只使用非空间特征属性而不用考虑经纬度中存在的差异，方法也更为成熟，缺点是它要求不同来源的 POI 之间必须有比较统一的存储模式，另外，非空间特征属性有可能存在信息缺失与标注错误问题。此外，在此步骤中使用低阈值的名称属性相似度算法的原因是：空间位置相近的 POI 对象有相似的名称。

[1]本文得到国家自然科学基金 60602017（No.60602017）和山东省自然科学基金（No.ZR2012FM016）资助。

第三，使用高阈值的名称属性相似度算法找出空间位置方法未能找出的对应对象。这里之所以使用高阈值的名称相似度算法，是因为二次过滤的 POI 数据没有空间位置的约束。

最后，用多组 POI 数据集合测试改进方案，实验结果表明融合集的准确率、召回率以及 F1 值都有明显提高。

关键词：POI 数据融合；准确率；召回率；F1 值

Study on fusion of multi-source POI based on the spatial location information [2]

Abstract

With the development of Web Electronic Maps and Location Based Services (LBS), the geospatial data such as POI appeared to grow rapidly. POI is the abbreviation of the points of interest (Point of interest), a point data which represents the true geographic entity, generally, POI includes basic information such as name, category, latitude, longitude as well as address. On the one hand, the collection, storage and update of POI information take lots of manpower and material, meanwhile, addition and update services of the POI information have become the core competitiveness of various network maps; on the other hand, the POI information from different sources have different maturity level. How to integrate and fuse the POI information from different sources has become an urgent problem.

The fusion technology of POI information is a key technology to solve this problem. The ultimate goal of the POI data fusion technology mentioned in this article is: identity the POI which represents the same geographical entity from two POI datasets, and put them into fusion set finally. So far, foreign researchers proposed some solutions such as Ontology-based technology; technology based on spatial location; technology based on non-spatial attributes. To find out the corresponding objects from distinct original POI datasets, we propose a novel approach on the bases of the abroad researching results. This approach increases the accuracy of the fusion set by using non-spatial properties similarity based on the spatial location. The detailed research works and results are as follows:

(1) Apply location-based algorithms to two POI datasets to find out initial fusion set consisting of corresponding objects. The advantage of the location-based algorithms is that you can fine the corresponding objects just use the latitude and longitude attributions and each POI almost have the latitude and longitude attributions, however, the shortcoming is that the latitude and longitude of POI from different sources have prevalent error as well as the not unified coordinate system .

(2) Use the methods based on name information with low threshold to exclude wrong

[2] This thesis is supported by National Natural Science Foundation of China (No.60602017) and Natural Science fndation of Shandong Province (No. ZR2012FM016).

corresponding objects obtained using location-based method. The advantage of this algorithm is that it uses only non-spatial characteristics without regard to differences in latitude and longitude, such method is also more mature, the disadvantage is that it requires different sources POI must have unified storage mode, additionally there maybe a lack of information and tagging errors in the non-spatial characteristic attributes. The reason to use the methods based on name information with low threshold is that the POI having similar name will have similar name.

(3) The remain corresponding objects not found will be searched out by using the methods based on name information with high threshold. The reason to use the methods based on name information with high threshold is that the secondary filter of the POI data is not the spatial location of the constraint.

Finally, this modified technique was tested on the different POI datasets. It has been demonstrated by our experiments that the precision, recall and F1-value of the fusion set was improved to a large part.

Key Words: **POI data fusion; precision; recall; F1 value**

目　录

1 引言

1.1 研究背景和意义

由于科技的迅速发展，地理空间数据已经由过去只有地理专业人士才能接触到的专业数据变成了普通人日常生活的一部分。不管是过去还是现在人们都渴望得到自身所在的地理位置和目的地的丰富信息，而网络电子地图的普及和各自API(Application Programming Interface)的免费开放为满足这种需求提供了可能，人们可以利用 API 很容易获得 POI 信息，POI 数据是一种代表真实地理实体的点状地理空间数据，其英文全拼为 point of interest，然而，不同来源的 POI 信息的更新速度以及丰富程度不尽相同，为了获取更新更丰富的信息，POI 数据融合和匹配技术就成为一种迫切需求。这种问题的比较专业的描述为：来源不同的 POI 数据通过 POI 融合技术生成信息量更为丰富与完整的 POI 数据，从而实现了 POI 信息的复用与更新，这样就可以节约大量的人力、物力，进而降低 POI 数据更新成本。

广义上的地理空间数据融合技术有着广泛应用前景，例如对不同来源的矢量地图或卫星影像对应区域进行融合从而得到信息量更加丰富的新地图。矢量地图或卫星影像的融合不仅考虑点状数据，除此之外还要考虑线状和面状地理数据以及它们之间的拓扑关系。具体到以 POI 为代表的地理空间数据融合技术的另一个现实应用是：为查询者提供信息量更丰富的感兴趣点(POI 点)，例如对从不同网络电子地图上抽取的感兴趣点利用 POI 融合技术对其进行融合处理，并最终显示给查询者，从而保证了查询者所查到的 POI 信息更为丰富全面。

综上所述，POI 数据融合技术是解决地理空间数据复用的关键技术，本文就重点研究了如何从两个来源不同的 POI 数据集合中准确找出用于融合的对应对象， 在国外研究成果的基础上体提出了一种可行的改进方案，并取得了不错的实验结果。

1.2 国内外研究现状

地理空间融合技术的最终目的是实现地理数据的复用，而其最常见的应用形态是地图集成，地图集成是将已经存在的数字地图集成融合成一个信息量更加丰富的新数据地图。基于空间数据融合的地图融合，在过去的二十多年中得到了广泛的研究，地图融合开始于锚点的选择，即代表同一位置的对应点对的选择，与锚点相关的数据集进行三角形平面细分（使用三角网格），并且在每个细分区域中使用 rubber-sheeting 转换[1-5]。地图融合的前提条件是锚点的选择，因此基于空间位置匹配算法就可以用在这个过程中，这些算法也可以用于基于中间件的在线地图数据库的集成[6, 7]。

数字地图的集成融合是个十分复杂的过程，其中较难处理的就是待融合的数字地图中的线状和面状地理数据之间拓扑关系的处理，而点状数据的融合集成相对简单些，因此也出现了针对数字地图中点状地理数据融合的空间数据融合技术，而对点状数据的融合可以作为整个数字地图融合的一部分。但是，即使是最简单的空间地理点状数据处理起来也有困难，因为空间数据不像结构化数据（如关系型数据）或者半结构化数据（如 XML 数据）具有全局标识，能用的信息只有空间位置（经纬度坐标）与非空间属性数据[8, 9]。

本文所提出的 POI 数据融合技术改进方案可以从两个来源不同的 POI 数据集合中找出表示同一个点状地理实体的 POI 对象，然后将这两个对应对象放到一个集合中，多组这样的对应对象就形成了一个完整的融合集[10]。针对这个问题，国外的科研工作者提出了一些解决方案，这些方案中比较有影响有：基于本体的技术[11-13]；基于空间位置的技术[14, 15]；基于非空间属性的技术[16-21]；空间位置与非空间属性结合的技术[22-24]。

基于本体(Ontology)的融合技术过于复杂，并且需要领域专家参与，其优点是它为每个 POI 对象生成一个类似结构化数据的全局标识，这样就可以像处理结构化数据一样轻松了；缺点是要为每个 POI 对象都生成这样的标识的前提是必须具备可靠的本体库，而目前为止还没有开发出这样的本体库，所以，这种方案还不实用。而基于空间位置的技术和基于非空间属性的融合技术的研究已经十分成熟，有些应用甚至接近实用并给出了相关的应用实例。

目前来说，基于空间位置信息的 POI 融合技术是一种较为实用且有效的技

术，它仅仅通过 POI 数据的空间位置信息就可以找出需要融合的对象。当前最流行的是片面最近邻方法，该方法也是商业地理信息系统常用方法。Catriel Beeri[14, 15] 等人提出了几种基于空间位置的空间数据融合新方法，其中比较重要的有：相互最近邻方法、概率方法、标准化权重方法。基于位置方法的优点是它仅仅根据经纬度位置信息就可以找对应对象，而经纬度信息是每个 POI 都必须具备的，不存在数据缺失问题；缺点是来源不同的 POI 的经纬度都普遍存在误差与坐标系不统一的问题。此外，在基于空间位置的所有算法中都涉及到一些参数选取最佳阈值问题，例如误差上限（对象位置信息所有误差总和）、融合集阈值参数等。如果来源不同的地理对象之间的距离大于误差上限就认为不可能成为融合集，而对象之间的可信度大于给定阈值就认为可能成为融合集。具体的参数需要根据具体的空间地理对象集合的特征来确定，基于空间位置的融合技术已经扩展到多个空间地理数据库融合的应用中[11-13]。

基于非空间属性的融合技术已经研究的相当深入，并且在具体应用的各个方面都有不错的进展，其优点是它只使用非空间特征属性不用考虑经纬度中存在的差异，方法也更为成熟，缺点是它要求不同来源的 POI 之间必须有比较统一的存储模式，另外，非空间特征属性有可能存在信息缺失与标注错误问题。

1.3 本文研究内容和创新点

本文研究的核心内容为：为从两个来源不同的 POI 数据集合中准确找出用于融合的对应对象，在国外研究成果的基础上提出一种改进方案，该方案在空间位置属性的基础上利用非空间属性相似度来提高结果融合集的准确性。技术路线如下：首先对两个 POI 数据集合实施空间位置方法找出对应对象组成的初步融合集，然后使用低阈值的名称属性相似度方法排除由空间位置方法找出的错误对应对象，最后使用高阈值的名称属性相似度方法找出空间位置方法未能找出的对应对象。用多组 POI 数据集合测试改进方案，结果表明融合集的准确率、召回率以及 F1 值都有明显提高。

本文所研究的对象为从不同网络电子地图上抽取的 POI 数据，该数据是一种地理点状数据，也是地图数据中较为简单的表示形式，并不涉及拓扑和方位关系等复杂运算。本文对 POI 数据库存储空间对象有以下要求：每个对象表示一个真实的地理实体，其中每个实体都至多有一个表示对象(即无重复项)。每个对象都

有相互关联的空间和非空间属性，空间属性描述位置，非空间属性诸如名称、地址以及电话号码等，POI 对象之间的距离是点位置间的欧几里得距离。

本文只考虑从两个 POI 数据集中找出正确融合集的问题，并且首先做出以下假设，首先，在每个 POI 数据集合中，不同的 POI 对象代表不同的真实地理实体，其次，各个 POI 数据集合中除了空间位置信息外还有其他可用的非空间属性信息，本文所处理的 POI 数据至少应包含名称属性字段。

本文提出的改进方案创新之处是在空间位置属性的基础上利用非空间属性相似度来提高结果融合集的准确性。

1.4 本文的组织结构

本文后续内容安排：

第二章 POI 实验数据

本章主要介绍 POI 数据与获取办法，以及相关的预处理过程。实验数据是从网络电子地图上抽取的 POI 数据，相关预处理主要是不同网络电子地图空间位置坐标的统一转换。

第三章 POI 数据融合技术

主要是空间地理数据融合技术的发展状况、制约因素以及已有 POI 融合技术的介绍，该章节将详细介绍了基于位置的技术与基于非空属性的 POI 融合技术。

第四章 基于空间位置的 POI 融合技术改进方案

该章节是本文的核心内容，本文的创新之处就体现在对基于空间位置技术的改进，通过把改进方案与已有 POI 数据融合算法做测试比较，可以看到其优异的表现性能。其中将详细说明最佳算法组合及阈值参数的选取，然后对本文提出的 POI 融合技术改进方案用多组 POI 数据集合做验证实验。

第五章 总结与展望

该章节对本文说研究的空间地理数据融合技术做出了总结，然后讨论了不足之处并提出后续的研究方向。

2 获取实验数据

2.1 POI 数据介绍

POI 是感兴趣点英文的缩写，其英文全拼为 point of interest，一般情况下，POI 应该具备的信息有名称、经纬度以及地址等。POI 数据是一种代表现实地理实体的点状数据，它可以代表建筑物、商店甚至是占有一定面积的地理存在。本文中所研究的 POI 数据除了以上给出的属性特征外，还可以有门牌号、邮编、地址、电话号码等更多丰富的属性信息。

Esri 中国（北京）有限公司给出的地理信息公共服务平台 **POI** 的行业分类有餐饮、购物、住宿，出行、文体娱乐、金融服务、生活服务、汽车服务、教育、医疗、房产、旅游、企事业单位、行政机构以及公共服务设施。下面给出部分行业分类的分层及其属性结构：

表 2-1 POI 数据分类与代码

一级类		**二级类**	
一级代码	**名称**	**二级代码**	**名称**
01	餐饮	0101	快餐
		0102	西餐
		0103	清真
		0104	海鲜类饭店
		0105	烧烤类饭店
		0106	火锅类饭店
		0107	综合类饭店
		0108	特色饮食
		0109	咖啡茶馆
		0110	食品店
		0111	其他
02	购物	0201	商场
		0202	超市
		0203	电子电器（苏宁、国美等）
		0204	建材家居（包括五金）
		0205	农贸市场

		0206	专营店(指品牌专营，如体育、文化、服装等)
		0207	其他
03	住宿	0301	宾馆旅店
		0302	星级宾馆
		0303	招待所
		0304	公寓
		0305	连锁旅店
04	出行	0401	长途汽车站
		0402	火车站
		0403	机场
		0404	码头
		0405	地铁
		0406	公交站点
		0407	公交 IC 卡营业厅
		0408	加油站（包括加气站）
		0409	停车场
		0410	售票点
		0411	立交桥
		0412	高速出入口
		0413	服务区
05	文体娱乐	0501	博物馆
		0502	展览馆
		0503	会展中心
		0504	图书馆
		0505	书店
		0506	美术馆
		0507	音乐厅
		0508	影剧戏院
		0509	青少年宫
		0510	科技文化宫
		0511	纪念馆
		0512	动物园
		0513	植物园
		0514	水族馆

		0515	公园
		0516	健身场所
		0517	体育比赛场馆
		0518	滑雪（冰）场
		0519	KTV
		0520	酒吧
		0521	网吧
		0522	游乐场（包括游戏厅）
		0523	度假疗养
		0524	洗浴中心
		0525	其他
06	金融服务	0601	银行
		0602	证券
		0603	保险
		0604	ATM
		0605	其他
07	生活服务	0701	水缴费网点
		0702	电缴费网点
		0703	煤气缴费网点
		0704	供暖缴费网点
		0705	通讯
		0706	家政
		0707	洗衣店
		0708	美容美发
		0709	摄影冲印
		0710	花卉礼仪婚庆
		0711	宠物
		0712	邮政（包括邮局、甚至邮筒）
		0713	物流快递
		0714	法律事务所
		0715	电台报社
		0716	人才市场
		0717	典（寄）行

表 2-2　POI 数据图层属性结构：

字段名称	中文说明	字段类型（长度）	说明
*NAME	名称	TEXT(60)	
*TYPE	兴趣点一级分类	TEXT(20)	
*TYPE2	兴趣点二级分类	TEXT(20)	
ADDNAME	地址名称	TEXT(200)	详细地址
ADDCODE	地址编码	TEXT(30)	与地名地址库挂接
TELEPHONE	电话号码	TEXT(20)	
*PAC	所在行政区划代码	TEXT(20)	填至县区级，参考行政区划代码
DES	描述信息	TEXT(200)	该兴趣点文字描述信息
RelateID	关联 Table 表中 RelateID	LONG	作为外键，与关联表的数据进行关联。默认值为 0，当该 POI 存在对应的图片时，应赋唯一值。

2.2 从网络电子地图中获取 POI 数据

现在国内比较流行的网络电子地图有谷歌地图、百度地图、mapabc、mapbar、搜狗地图、51 地图、365 地图网、搜搜地图、微软 Bing 地图、雅虎地图、有道地图等。我们选取谷歌地图和百度地图作为本文实验用的 POI 数据来源，可以通过谷歌地图和百度地图提供的 API 与 Local Search 功能来分行业抽取 POI 数据，本文抽取的 POI 数据所属行业的一级分类为餐饮，其分类代码为 01。

2.2.1 谷歌地图 API 与 Local Search 功能

Google Maps API 是 Google 公司针对网络地图开发需求推出的免费编程开放接口，用户只需要使用网页编程语言调用谷歌地图提供的 API 函数，就可以很容易地实现基于谷歌地图上的各种 API 功能[25]。这种在线的地图服务有两大优势：一是可以同过 Google Maps API 各种功能免费利用谷歌地图强大的后台服务；

二是注册用户可以轻松地将自己的标签添加到网页上进行信息整合[26]。

Google 地图 API 所提供的地图加载、标注、自定义、地图控件与地图属性控制等功能也都展示在 Google 地图这个基础元素之上。由于本文只用到了地图展示和标注功能，所以现在只介绍这两个操作，其他功能的实现可以参照谷歌 Gmap API 的官方网站[27]。通过 Google 地图 API 实现的地图标注结果如下：

图 2-1　利用 Google maps API 在可视域地图中心点做标记

上图表示的是在可视域地图中心点做标记，目的是配合谷歌地图提供的Local Search 功能将从可视域地图中抽取的 POI 数据标注在地图上进行展示，然后可以分区域移动搜索抽取 POI 数据。

下面就来介绍谷歌地图提供的 Local Search 功能（即本地地图搜索功能）：Google Local Search API 与 Google maps API 一样也是谷歌公司为谷歌地图开发者提供了一个 API 编程接口，常见的 Google Local Search API 应用包括：显示经过搜索查询的本地化结果，构建显示查询结果的可搜索地图，构建本地搜索结果的静态地图图像等。本文主要应用是对可视域地图进行分行业搜索查询，并且将搜索查询到的 POI 数据本地化，即保存下来然后利用 MySQ 数据库对所抽取的 POI 数据进行数据融合操作。下面是利用 Google Local Search 功能结合 Google maps API 从谷歌地图上搜索查询青岛市南区某片区的餐饮行业的所有 POI 点数据，然后分别对结果进行本地存储和在地图上标记展示：

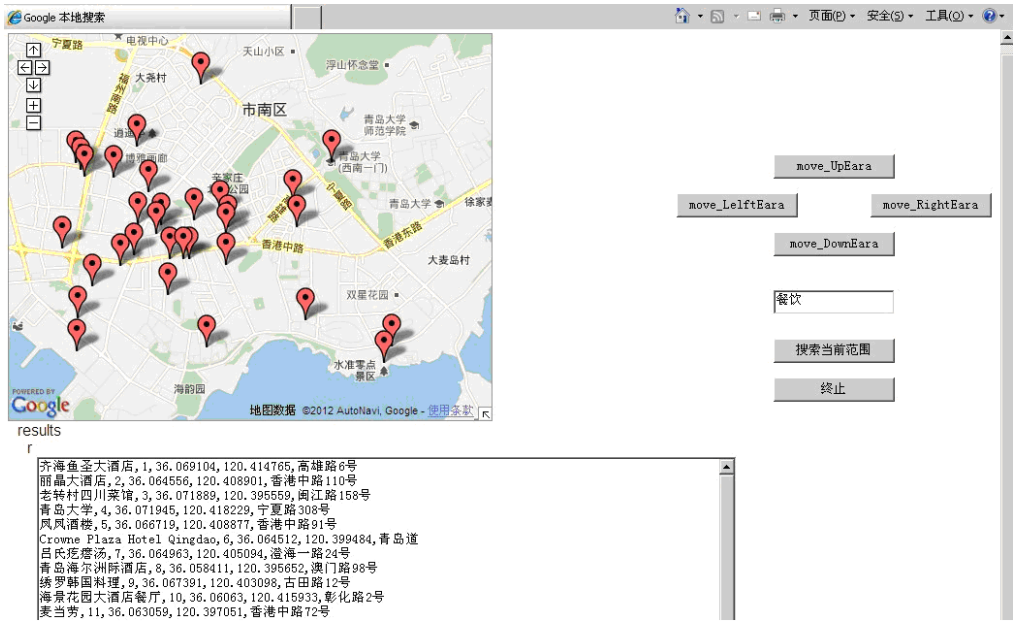


图 2-2　谷歌地图本地化搜索结果

2.2.2 百度地图 API 与 Local Search 功能

百度地图 API 是百度推出的一套以百度地图为基础元素的免费应用编程接口，百度地图 API 提供主要功能有：基本地图展示、本地搜索、路线规划、地理定位、逆地理编码与 LBS 云存储与检索等功能。本文选取的接口版本为基于 JavaScript 语言版本的，百度地图 JavaScript API 是一套由 JavaScript 语言编写的应用程序接口，它可以让用户轻松地在百度地图元素上很方便地实现各种功能操作，此外，百度地图 API 同时将大量复杂的底层逻辑进行了隐藏和封装，从而使得用户更容易使用[28]。类似地，下面给出百度地图标注的示例结果：

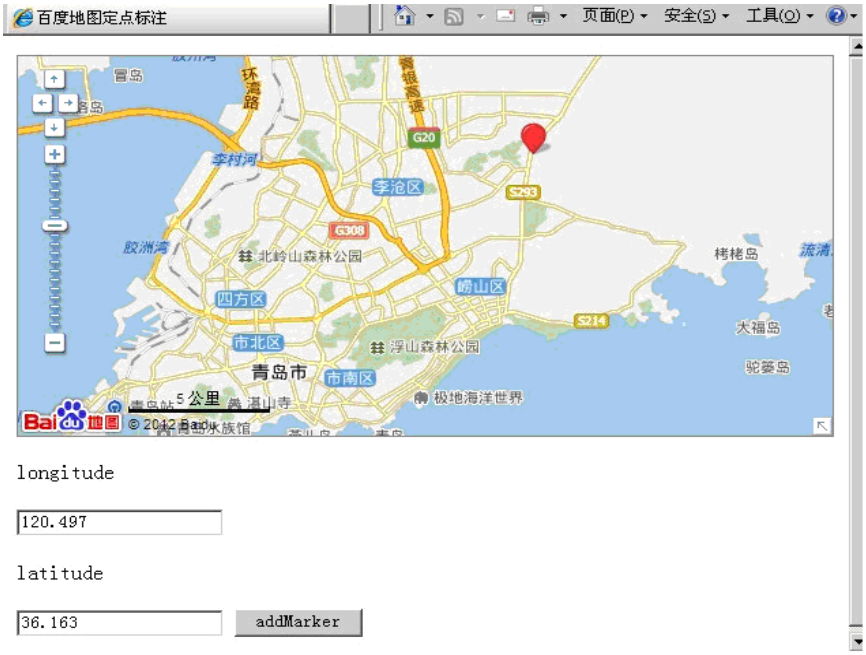


图 2-3　利用百度地图 API 在可视域地图中心点做定点标记

上图实现的功能是利用百度地图 API 把给定的坐标值标注在百度地图上，介绍这个功能的最终目的也是配合百度提供的 Local Search 功能采集 POI 数据，然后以标记的形式显示在百度地图上。下面给出百度地图的本地搜索结果（搜索行业也同为餐饮业）图如下：

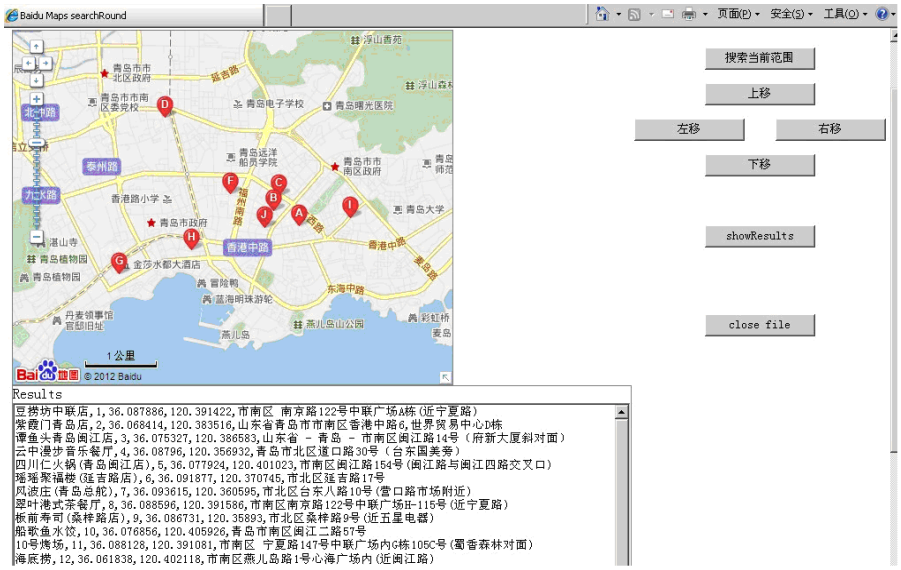


图 2-4　百度地图本地化搜索结果

2.2.3 从谷歌地图和百度地图中抽取 POI 数据

百度地图和谷歌地图都能提供全国各个城市诸如道路、店铺、学校、酒店等信息，这些信息也是我们组织 POI 实验数据的来源。但是从谷歌地图和百度地图

利用本地所搜索所得到的 POI 数据结果中可以看出，从百度地图中抽取的 POI 数据的地址属性要比从谷歌地图中抽取的 POI 数据地址属性详细，就本文中对百度地图和谷歌地图的 POI 数据融合而言，直观的好处就是用百度地图提供的具有详细地址属性的 POI 数据结合代表同一个真实地理实体的谷歌地图 POI 数据，最终为用户产生信息量更为丰富的 POI 数据。其实不同地图 POI 数据之间除了字段属性信息丰富程度不同外，而且还有字段更新速度和字段缺失等问题的差异。

POI 数据抽取过程如下：用百度地图和谷歌地图提供的本地搜索实施可视地图范围内的范围搜索，对所需 POI 数据的范围逐一移动可视区域进行搜索并保存搜索结果，这里搜索的范围选定为山东省青岛市市南区，市北区，崂山区，四方区，李沧区。所涉及的行业有书店、学校、餐饮等行业。抽取结果图示如下：

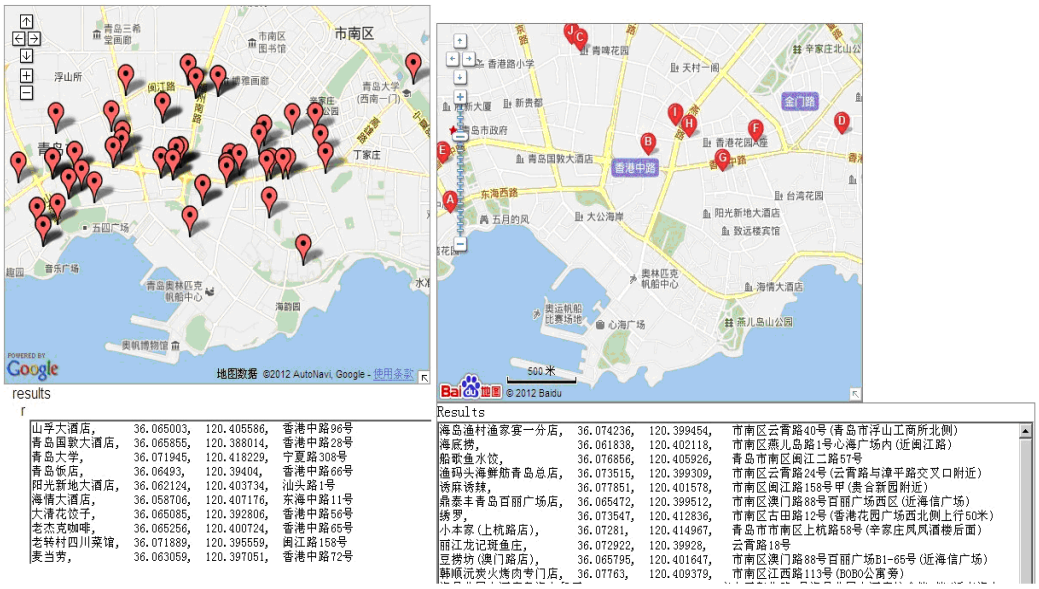


图 2-5　从谷歌地图和百度地图获取对应区域 POI 数据集合

上图表示对同一区域同一行业的 POI 数据抽取结果，并且用 MySQL 数据库存储最终抽取的 POI 数据。从上图中可以看出，从谷歌地图和百度地图上抽取的 POI 数据信息丰富程度不同，特别是百度的名称字段和地址字段要比谷歌地图的详细不少。事实上，除了以上列出的字段值，利用谷歌地图和百度地图的本地搜索功能还可以得到 POI 数据的电话号码、邮政编码以及门牌号码等属性信息，所以不同来源的 POI 数据的融合可以丰富已有的属性字段。

2.3 POI 实验数据预处理

国内来自不同地图的 POI 数据存在较大差异，其中最严重的是空间地理坐标的不统一问题，接下来就着手解决这个问题以及详细说明本文所用的 POI 实验数据的生成过程。

2.3.1 地理坐标统一问题

理论上讲，不同参考椭球体的地理坐标系之间是可以自由转换的，例如，用布尔莎公式完成 WGS-84 坐标系到北京 54 坐标系的转换[29]。要完成坐标之间的转换，前提是必须知道这些坐标是在什么坐标系下生成的，不同的坐标都要对外公布其转换参数的[30]。目前，除了知道英文版的谷歌地图采用的是 WGS-84 坐标系和 WEB 墨卡托投影外，其他的网络电子地图都没有公布其采用的坐标系统。

在国内，谷歌地图分为两个版本，即中文版谷歌地图(http//ditu.google.cn/)与英文版谷歌地图(http://maps.google.com.)，两者的经纬度定位存在差异，也就是说所采用的地理坐标系不同。英文版谷歌地图采用的是 WGS-84 坐标系[31]，但是中文版的谷歌地图没有公布其采用的具体地理坐标系。英文版谷歌地图的卫星视图采用的是真实的 GPS 坐标，而地图矢量模式采用的是经过加偏处理后得到的坐标，两者在同一点存在较大偏差，这是因为地图服务商至少使用了国测局制定的 GCJ-02 加密算法。此插件会将真实坐标进行偏移，且为非线性加偏。下面就对在 WGS-84 坐标系下的地理坐标点（36.052777,120.324637）给出展示，如下：

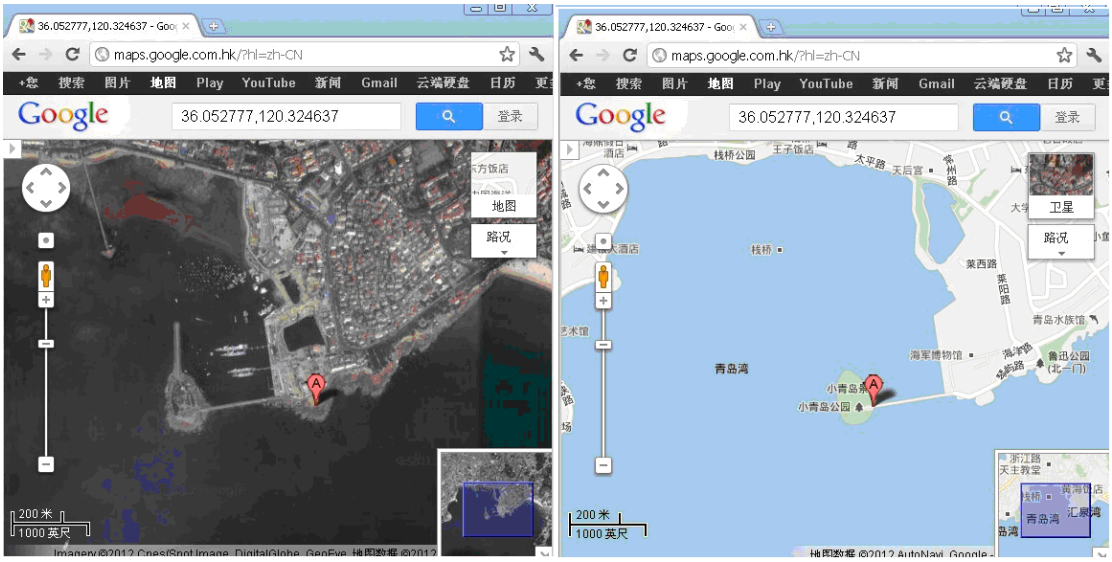


图 2-6 同一地点在国际版谷歌地图中的偏移

卫星视图的地理实体的空间坐标是在 WGS-84 坐标系下定义的，而矢量地图模式是经过加偏后的坐标，可以看出两者存在较大偏移，如果不解决偏移问题就无法用 GPS 提供的坐标进行空间地理数据融合计算。解决办法为：

建立并查找偏移数据库得出相应坐标的偏移数据，纠偏精度取决于纠偏数据库的精度，比如 0.1 精度的谷歌地图纠偏，0.1 是指偏移数据的间隔，即如果 POI 数据的地理坐标值的小数部分中第一位值大小相等，那么就对这样的 POI 数据坐标用相同的纠偏值进行修正（理论依据：加密算法整体非线性，局部可以看作是线性的且变化缓慢），这样不同的点会出现不同的偏差，偏差范围在 10 米到 20 米之间。精度更高的 0.01 度纠偏数据库可以使偏差范围缩小到 5 米到 10 米。经过编程后实现结果如下：

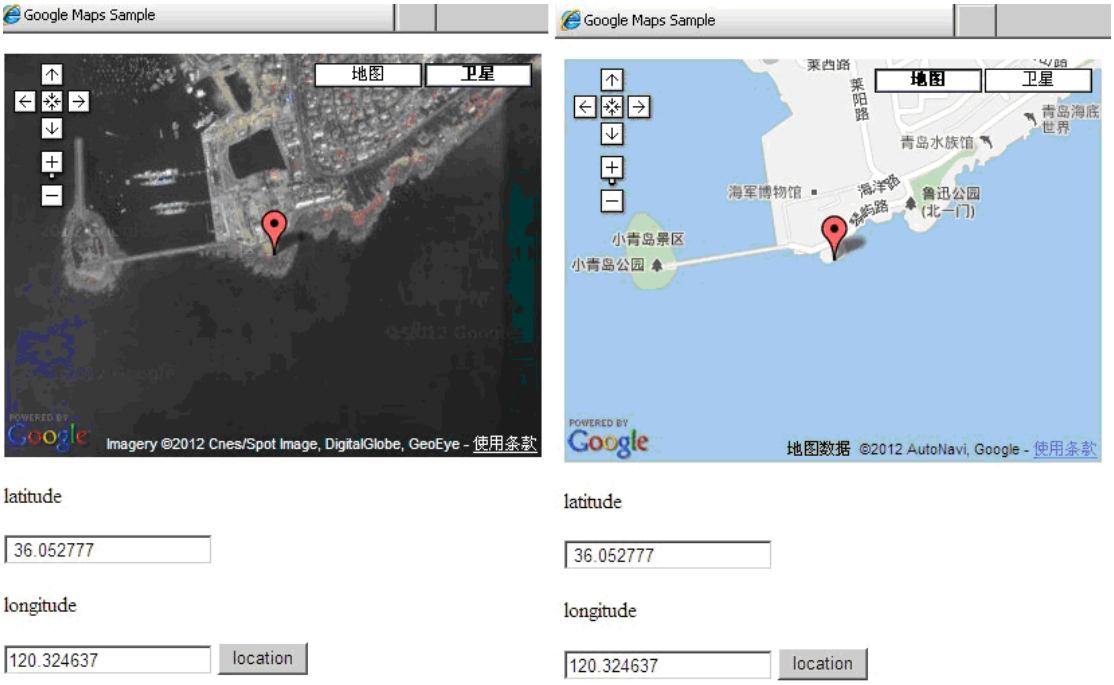


图 2-7　0.1 精度的谷歌地图纠偏结果

图中同为地理坐标点(36.052777,120.324637)经过 0.1 精度数据库纠偏后，真实 GPS 坐标与加偏后的坐标相差最大不差过 20 米。

然而，除了谷歌地图外，国内所有的网络电子地图地理坐标都经过了各个地图运营商的加偏处理。还有些地图运营商比如百度、mapbar 等在此基础上实施了二次加偏，例如百度地图对真实 GPS 坐标使用国测局制定的 GCJ-20 算法首次加密后，又进行了 BD-09 算法的二次加密措施。各个地图的加偏算法不尽相同，这就给不同来源的 POI 信息融合带来困难，所以必须解决不同地图间对应坐标点

的偏移问题才能达到最终目的。

由于不知道各个网络地图的具体采用的坐标系，再加上各个地图都进行了加密处理，因此就不能用常规的在已知坐标系情况下按照公式转换了，但是大多数地图服务供应商都有开放的 API，其中有些提供不同地图间的坐标转换接口，这种方法的缺点是：过度依赖网络且有数量限制。

大多数网络电子地图服务商都有开放的 API，并提供地图间坐标转换接口，这就为实施 POI 数据地理坐标转换提供了便利，本文已实现的坐标转换如下：

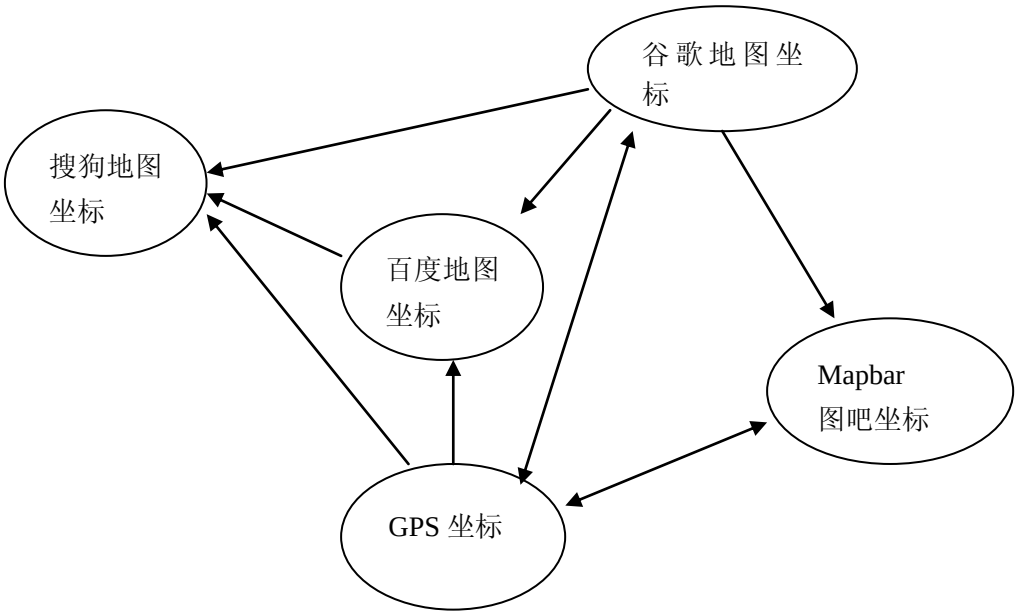


图 2-8　双向箭头表示实现双向转换，单向箭头表示单向转换。

不同地图之间坐标统一的实现示例：

谷歌地图坐标到百度地图坐标转换，其转换接口以及返回结果如下：

http://api.map.baidu.com/ag/coord/convert?from=2&to=4&x=longitude&y=latitude

longitude 与 latitude 分别为谷歌地图模式下的经度和纬度，接口返回的结果为 Base64 码格式。例如对谷歌地图坐标(120.327904,36.05323)做转换：

http://api.map.baidu.com/ag/coord/convert?from=2&to=4&x=120.327904&y=36.05323

转换接口返回格式为：

{"error":0,"x":"MTIwLjMzNDQyMjQ5NjIz","y":"MzYuMDU5MTkyNDU1MjUy"}

解码后的对应的百度坐标(取小数后六位)为：120.334422, 36.059192

所以谷歌地图坐标(120.327904,36.05323)在百度地图中的对应点应该是(120.334422, 36.059192)。

GPS 坐标到百度地图坐标转换，函数接口为：

http://api.map.baidu.com/ag/coord/convert?from=0&to=4&x=longitude&y=latitude

其他地图之间坐标转换过程与此类似，本文使用百度地图 API 提供的接口函数直接进行谷歌地图坐标到百度地图坐标的转换。GPS 坐标到百度地图坐标转换函数为 BMap.Convertor.translate(ggPoint,2,translateOptions)。

谷歌地图坐标到百度地图坐标的转换函数为：

BMap.Convertor.translate(ggPoint,0,translateOptions)；其中 translateOptions()为回调函数，详细解说请参照百度地图 API 开发文档，转换结果如图所示：

图 2-9 同一地理地点谷歌坐标与 GPS 坐标转换到百度坐标结果图

从图中可以明显看出同一地理地点在谷歌地图模式下的地理坐标与 GPS 坐标存在明显差别，而通过坐标转换函数把这两个 POI 点的坐标统一到百度地图坐标下就基本可以消除这种差异了。因此要实现这两个坐标系下的对应 POI 数据融合，就必须解决坐标统一的问题。

2.3.2 生成实验数据集

本文首先把来自谷歌地图 POI 数据坐标统一到百度地图坐标，然后，对从两家地图上抽取的 POI 实验数据集合进行人工判定，找出对应项。判定的标准为先观察名称属性，再观察地址信息等，确认为对应项后人工分配 ID 号，以便下面做实验时计算融合集质量的评价指标，经过预处理的 POI 数据作为最终的算法实验数据源。下面就详细说明生成 POI 实验数据源的过程：

1、用百度地图或谷歌地图提供的本地搜索实施可视地图范围内的范围搜索，对所需 POI 数据的地图范围逐一移动可视区域进行搜索并保存搜索结果，这里搜索的范围选定为山东省青岛市市南区，市北区，崂山区，四方区，李沧区。所涉及的行业有书店和餐饮等服务行业。所选用的地图服务为谷歌地图，最终找到的未去重的 POI 数据总量为 2496 条并保存为 gg_origine_poi。

2、对谷歌地图中得到的 POI 数据集 gg_origine_poi 进行去重处理，这里去重处理有两个方面：

a、title 字段值相同（但不一定是重复项，比如同名异地店） 。

b、longitude、latitude 字段值相同（可以认为是完全的重复项）。

针对以上两种情况分别对源数据进行 title 字段值去重和 longitude、latitude 字段值去重，并对处理后的数据集保存为 gg_no_duplicateTitle_poi (去重后有 2109 条 poi 数据) 和 gg_no_duplicateLngLat_poi (去重后有 2136 条 poi 数据) 。

3、取 gg_no_duplicateTitle_poi 数据集中的 title 字段值，用百度提供的本地搜索功能进行模糊搜索，并保存为 baidu_origine_poi ,对同一条 gg_no_duplicateTitle_poi 中的 POI 数据保证其 ID 字段值对应，这可以提供代替最后人工校正的可能）。

4、最后对 gg_no_duplicateLngLat_poi 进行坐标转换，使其统一到百度地图坐标系下并保存为 ggtobaidu_no_duplicateLngLat_poi (2136 条)，对转换后的数据集去重保存为 ggtobaidu_no_duplicateLngLat_poicopy（有 2119 条数据），此过程由百度地图 API 提供的转换函数实现，然后对 gg_no_duplicateLngLat_poi 和 bd_noDuplicateLngLat_poicopy 实施基于空间位置的片面最近邻算法，从而得到包含对应项的结果集 corresponding_pois。再对结果进行人工校对（可以用 ID 或者 title 字段值匹配代替人工）。

上述一系列生成 POI 实验数据源的过程可以概括为：利用谷歌地图与百度地图提供的本地搜索功能，搜索山东省青岛市市南区，市北区，崂山区，四方区，李沧区范围内的书店和餐饮等服务行业；然后对搜索到的 POI 数据进行本地化存储，选择用 MySQL 数据库存储，建立相应的数据库与数据存储表格；再对所获的百度地图 POI 数据与谷歌地图 POI 数据进行数据去重处理与坐标转换处理，最后根据两个不同来源的各个 POI 的名称属性信息与地址属性信息进行人工判

断，如果确定为表示同一个真实地理实体的对象，即对应对象，就为两个对应的对象分配相同的 ID 编号，详细的 POI 数据源生成流程图如下：

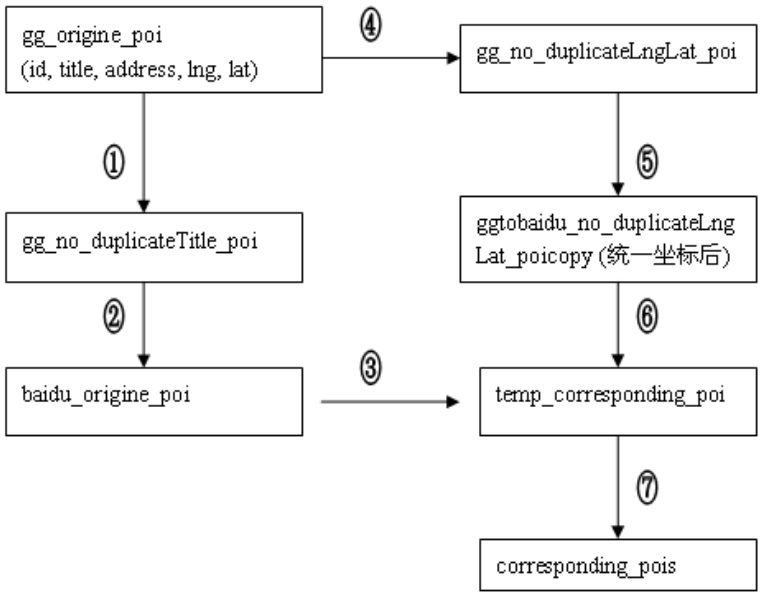


图 2-10　POI 数据源生成流程图

①、名称（title 字段值）去重。去重后的数据集将不再有重名 POI 数据，包括同名异地点。

②、根据 gg_no_duplicateTitle_poi 的 title 字段，进行百度本地模糊搜索，得到百度地图中对应的 POI 数据点集。

④、经纬度（lng, lat 字段值）去重，去掉完全重复的 POI 数据。

⑤、对 gg_no_duplicateLngLat_poi 坐标转换，使其统一到百度地图坐标系下。

③，⑥、对 baidu_origine_poi 和 ggtobaidu_no_duplicateLngLat_poi POI 数据集实施基于空间位置的片面最近邻算法，得到 POI 数据对应集合 temp_corresponding_poi。

⑦、最后对结果进行人工校对（也可以用 ID 号匹配代替人工）。

下面对步骤 ② 所遇到的可能情况做深度介绍：

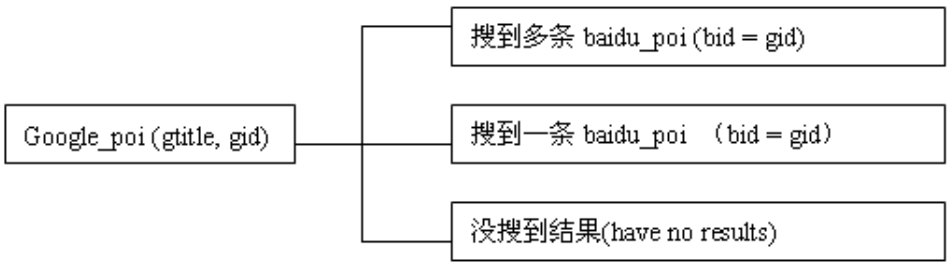


图 2-11　POI 结果集中对应对象图示

在前两种结果中，最好的情形是搜到且仅搜到了 google_poi 的对应百度 poi 点，其次是搜到多条但其中有一条是对应点，最坏的情形是没搜到反而带来干扰项(这会影响下一步与 ggtobaidu_no_duplicateLngLat_poicopy 找对应项的操作)。

经过③，⑥步骤后得到 corresponding_pois 对应对象结果集，该结果集共有 2119 条数据，其字段包括 gtitle, gid, glat, glng, gaddress, btitle, bid, blat, blng, baddress。

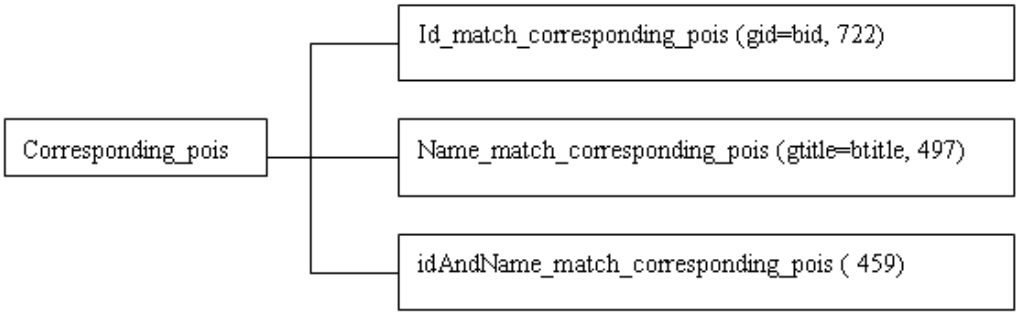


图 2-12　寻找 POI 对应对象结果图示

经过核实，id_match_corresponding_pois 和 name_ match_corresponding_pois 存在不同程度错误项（对应有误），而 idAndName_match_corresponding_pois 结果令人满意(基本无错误对应项)。

影响结果一些因素的补充说明：

1、 ggtobaiddu_no_duplicatelnglat_pois 与 baidu_no_duplicatelnglat_pois 的 overlap（重合度，这依赖于步骤 ②的搜索结果）。 在③，⑥步骤中所用的片面最近邻连接算法结果令人满意的程度取决于两个 POI 数据集的重合度。

2、 在对两个数据集实施最近邻算法中，选取 100 米作为两 POI 数据的误差上限。

3、 Ggtobaidu_no_duplicatelnglat_pois 在向百度坐标转换过程中引入误差，经初步测定这个转换最好情况下精确到坐标小数点后第四位。

4、 百度用谷歌 poi 数据的 title 字段模糊搜索的名称，对应项与谷歌 poi 名称差别比预定的要小（就是同一 POI 点在不同地图中使用不同名称情况很少）。

经过以上操作后，可以找到能用来做融合匹配试验的 POI 数据源，下面就组

织生成不同差异程度的基于空间位置算法的 POI 验证数据集合：

人工标注的基于空间位置的 POI 测试数据的组织方案：

①、新建 MYSQL (poidata)数据库,并将谷歌与百度地图 poi 数据导入新建的表 baidu_origine_poi 和 gg_origine_poi 数据库表中，然后， 依据两数据库表中的 ID 找出可能的对应项集合。

②、然后将数据导出到 excel 表中，通过人工判断所得到对应项集合的非空间数据属性标注是否为真实对应项，如果是就标识为 1 ，否则标识为 0。

③、筛选出人工标注为对应项的集合，再分别导入到两文件中，如两数据库表中。

④、对两文件各随即选取 100、300、500、700、800 不等数量 poi 测试数据，这意味这不同的重合度，假设共有 1000 对对应数据，随即选取的数量越大其重合度就越大。也可以用 k-fold 方法（机器学习算法中的一种测试数据分类方法）。流程图如下：

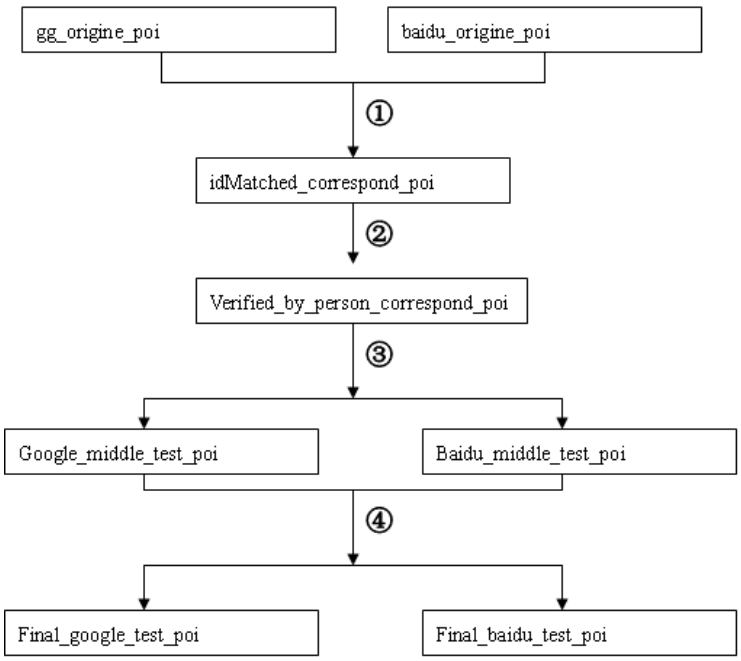


图 2-13　空间位置算法初步测试 POI 实验数据集生成图

算法最终生成的空间位置算法的 POI 验证数据集合为：

表 2-3　不同差异程度的空间位置算法的 POI 验证数据集合

实验数据总数(对)	实体总数	正例数	反例数	正例比例
100	187	13	164	0.07
300	505	95	410	0.19
500	714	286	428	0.41
700	833	567	266	0.68
800	865	735	130	0.85

接下来给出最后生成的 POI 实验数据集合表示，如下图所示：

图 2-14 谷歌地图 POI 与百度地图 POI 实验数据最终形式

从图中可以看出，百度地图 POI 数据的非空间属性中的名称字段与地址地段要比谷歌地图中的名称字段与地址字段详细。两地图 POI 数据的最终融合结果将会包括比单独任何一个地图 POI 数据信息量都丰富，这也是进行不同来源 POI 数据集的初衷。

3 POI 数据融合技术

3.1 基于空间位置的技术

基于空间位置的技术是一种较新的方法，它仅仅用对象的空间位置寻找正确的融合集。Catriel Beeri 等人提出了几种基于空间位置数据融合新方法:相互最近邻方法、概率方法、标准化权重方法。假设有两个来源不同且需要融合的 POI 数据集 A 与 B，其中 A = {a_1, ……, a_m}，B = {b_1……, bn}，基于空间位置的 POI 融合技术都有默认的条件，即对应 POI 对象之间的距离要比非对应 POI 对象之间的距离（欧几里德空间距离）近。

3.1.1 影响因素

POI 数据融合的制约因素除了数据存储模式以及结构的不一致性外，还有来源不同的 POI 数据的空间坐标系的不统一，此外，所要融合的 POI 数据不像结构化或者半结构化数据那样具有全局标识[32-35]，这也增加了具体操作的困难。

在基于空间位置的技术中，影响结果融合集质量的因素还有 POI 数据集的误差间距、数据集密度大小、选择因子以及误差上限等。误差间距指的是 POI 数据集中的地理对象与它所表示的真实地理实体之间的距离误差范围大小，而 POI 数据集密度大小指的是单位面积上地理对象的个数。选择因子的含义则为以误差间距做的圆中地理对象的个数，也就是说选择因子是以误差间距形成圆形区域和空间地理数据集密度的综合产物。误差上限指的是 POI 数据集中所有位置误差的总和。

此外，POI 数据集合之间的重合度大小也是影响结果融合集质量的关键指标。假设有两个 POI 数据集合 A 和集合 B，这两个数据集合中分别有 m 和 n 个 POI 对象，用 c 表示两个数据集合之间对应对象的个数，那么这两个 POI 数据集合 A 和 B 的重合度则为 $c/(m+n)$。重合度最重要的真实意义就是对另一个数据集中有对应项的那部分 POI 对象的估计，本文提出的基于空间位置的 POI 数据融合改进方案要解决的问题是，在各种重合度下所得到的结果融合集既要取得很高的准确率也要取得很高的召回率，本文中所提到的具有不同重合度的 POI 数据集合也称为具有不同差异程度的 POI 数据集合。在所有情况当中有两种特殊的情况，

那就是两个地理数据集合之间的重合和覆盖。

误差上限是 POI 数据集合中所有可能的位置误差总和，假设地图 K 和地图 X 中 POI 点位置与其所代表的真实地理实体之间的误差范围大小分别是 κ 和 χ，那么在地图 K 和地图 X 中，POI 对应对象的误差上限为：

$$\beta = \sqrt{\kappa \wedge 2 + \chi \wedge 2} \qquad \text{式（3-1）}$$

一般情况下误差分布都符合正态分布，δ_K、δ_χ 分别为地图 K 和地图 X 与真实地理实体误差分布的标准差，那么则有 $\kappa = 2.5\delta\kappa$、$\chi = 2.5\delta\chi$。

本文搜集到的针对谷歌地图和百度地图中点状地理数据与真实地理实体误差的估计方法有：在局部用多项式集合；首先假定误差分布符合正态分布规律，然后在局部采集误差样本点，再用数学方法估算出标准差。

3.1.2 片面最近邻连接算法

片面最近邻连接算法已经普遍应用于是商业地理信息系统中。对于一个给定的 POI 对象 $a \in A$，如果 POI 数据集合 B 的所有 POI 对象 $b \in B$ 与对象 a 的空间距离最近，那么就说对象 b 是对象 a 的最近邻。对 POI 数据集合 A 和 B 进行片面最近邻算法运算将产生所有可能的融合集 $\{a,b\}$。每个对象 $a \in A$ 都存在于结果融合集中，但是集合 B 中的对象可能不存在或者出现在多个结果融合集中。片面最近邻连接算法的另外一个重要特点是非对称性，即 POI 数据集合 A 与 B 的连接运算和 B 与 A 的连接运算所得到的结果融合集可能不一样。

为了能提高片面最近邻连接算法运算结果的准确率和召回率，我们做出以下改进：

①、如果在片面最近邻连接算法对 POI 数据集合 A 与 B 的运算结果中，存在 $dis\tan ce(\{a,b\}) > \beta_{AB}$ 的情况，那么就从结果融合集中删除这样的融合对集 $\{a,b\}$，这里 $dis\tan ce(\{a,b\})$ 的意义是 POI 对象 a 与 b 之间的空间欧几里得距离，具体值由地理对象之间的纬度和经度计算；而 β_{AB} 代表两个 POI 数据集合 A 进而 B 的误差上限，即各个数据集合中所有与位置相关的误差总和。

②、最后对于没出现在结果融合集中的 POI 对象 $a \in A$ 和 $b \in B$ 做出以下

处理：形成结果融合单集 $\{a\}$ 与 $\{b\}$ ，结果融合单集表示在另外一个 POI 数据集合中没有要与之融合的对象。

连接运算顺序的选择规则是片面最近邻连接算法中用到的小技巧，其具体内容为：如果要使用片面最近邻连接算法的两个 POI 数据结合的对象规模不一样大或者说差别很大，那么我们要选择的连接运算顺序是用规模小的那个数据集连接较大的那个数据集，这样运算产生的结果就会比较合理。下面列举一个极端的例子，比如有两个 POI 数据集合 A 和 B，数据集 A 包含 100 个地理对象，而数据集合 B 仅仅包含 1 个地理对象，如果我们选择 A 连接 B 的运算顺序，那么将产生多达 100 个结果融合集，但是在这 100 个融合集中仅仅有一个是正确的；如果我们选择的连接顺序是 B 连接 A ，那么就产生一个结果融合集，这个唯一的结果融合集就是我们要找的。

综上所述，片面最近邻连接算法的伪码实现如下：

Input : Datasets **A** and **B**
Output : **a** set **P** of pairs and a set **S** of singletons
Process :
P ← ∅, **S** ← ∅, **C** ← ∅
Let β be the distance upper bound of **A** and **B**
For each **a** ∈ **A** find the nearest **b** ∈ **B** of **a** then
P ← { **a**, **b** }
C ← { **b** }
At end
S ← **B** – **C**
Return (**P, S**)

片面最近邻连接算法实现的流程图如下所示：

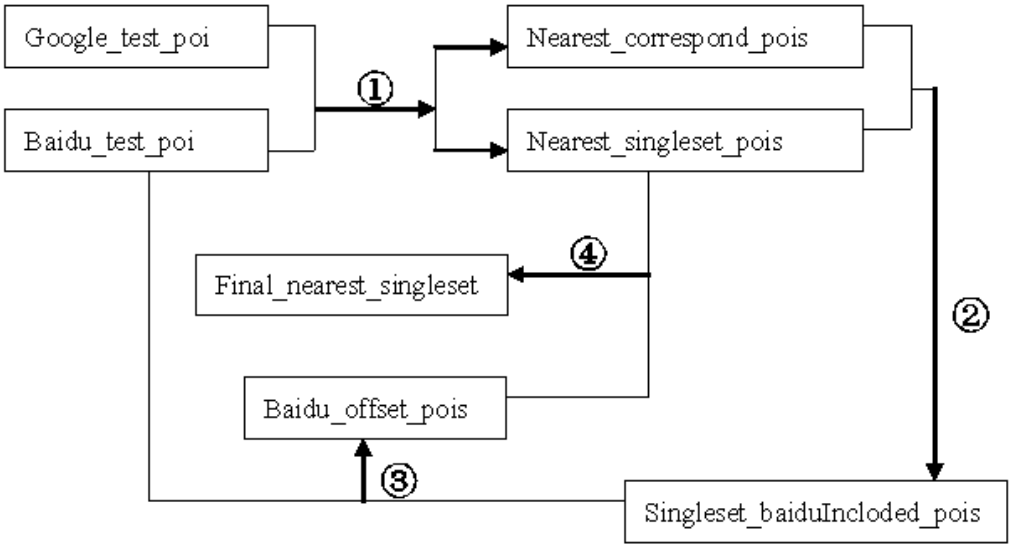


图 3-1 片面最近邻连接算法实现的流程图

步骤详解：

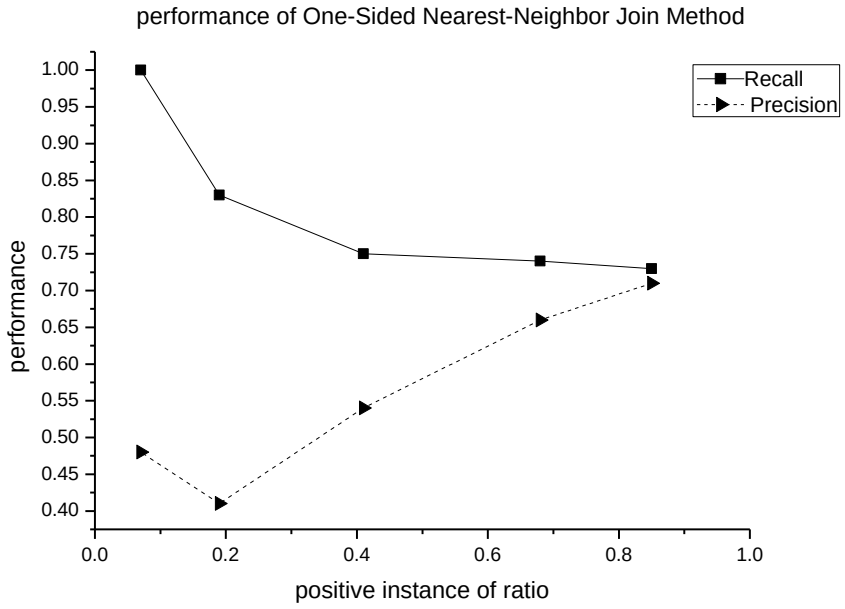
①、对两组实验数据实施片面最近邻算法，Google_test_poi 与 Baidu_test_poi 为前面获得的分别来自谷歌地图与百度地图的 POI 测试数据组。其中算法实现的连接顺序为，谷歌地图 POI 数据连接百度地图 POI 数据。

②、取出结果集合(包括融合对集与单集)中百度地图 POI 点数据。

③、取出未被谷歌地图 POI 点数据选中的百度地图 POI 数据，具体实现所用的 SQL 语句为：Select * from baidu_test_poi where not exists (select * from single_baiduIncloded_pois where id=baidu_test_poi.id); (就是差集运算). 结果集为百度数据集中未被选中的 poi 点数据。

④、未被选中的百度数据集合与算法直接得出的部分单集结合成最终的最近邻单集结果集。

下面给出使用单向片面最近邻连接算法测试自谷歌地图与百度地图的代表不同差异程度的 POI 测试数据所生成的成对融合集的表现性能作图如下：



0

图 3-2　片面最近邻算法的融合集中对集的召回率与准确率

从图中可以看出，当两个不同来源的 POI 数据集重合度越大（即正例比例越高）时，片面最近邻连接算法表现越好。当两个数据存在覆盖或者重合的情况下，片面最近邻算法表现最好。这里我们仅仅对结构融合集中的成对集合作分析，这也是本文衡量算法优劣的核心评价指标，其他的如单集、综合集的表现分析方法类似。

3.1.3 相互最近邻算法

相互最近邻方算其实是对片面最近邻算法的改进，如果来源不同的两个对象之间互为片面最近邻，则称它们是一对相互最近对象。相互最近方法有一个隐含假设，即对应对象彼此相距最近。注意 POI 数据集 A 中的一些对象有可能不在成对的相互最近对象中（同样也可能发生在 B 的对象中）。例如，对象 $a \in A$ 在 POI 数据集合 B 中的最近邻是某个对象 a ,但是对象 $b \in B$ 在数据集 A 中的最近邻不是同一个对象 a ，在相互最近邻算法中，每一组相互最近邻对象形成一个二元融合集。

那么最近邻方法中融合集的置信度定义如下：

$$confidence(\{a,b\}) = 1 - \frac{dis\tan ce(a,b)}{\min\{dis\tan ce(a,b_2), dis\tan ce(a_2,b)\}} \quad \text{式（3-2）}$$

$$confidence(\{a\}) = 1 - \frac{dis\tan ce(a_1, b)}{dis\tan ce(a, b)} \qquad \text{式（3-3）}$$

在公式(3-2）中，$a \in A$，$b \in B$，$confidence(\{a,b\})$ 为融合对集$\{a,b\}$的置信度，其中对象a与对象b互为相互最近邻，a_2为对象 b 在 A 中的次近邻，b_2为对象 a 在 B 中的次近邻。如果置信度大于给定阈值就认为是融合集。

考虑到单集结果的存在，即融合集中只有一个元素，具体到相互最近邻算法中，其具体意义为：在$a \in A$集合中，对象a在数据集合B中不存在互为相互最近邻$b \in B$，如果b为集合B中 $a \in A$的最近邻，a_2为b在集合A中的次近邻，那么公式(3-3)为单集 $\{a\}$ 的置信度定义。这里要指出的是置信度值为非负值，单元素融合集 $\{b\}$ 的置信度值定义与之类似，在此不再赘述。另外，如果对于任一个$b \in B$，都有$dis\tan ce(a,b) > \beta$，其中β为空间位置属性的所有可能的误差之和，即前文中提到的误差上限，那么则有$confidence(\{a\}) = 1$，对所有的$b \in B$，则有定义$confidence(\{a,b\}) = 0$。

相互最近邻算法的示意图如下：

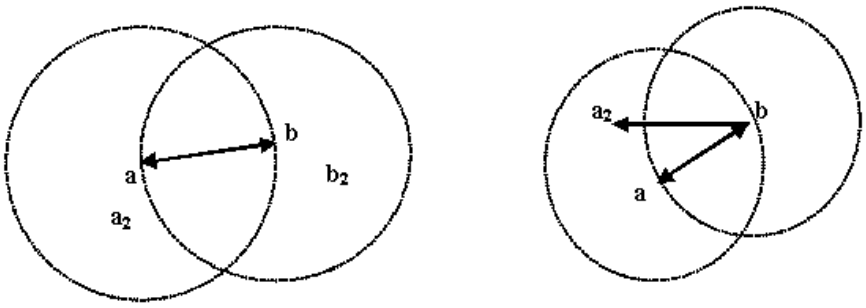


图 3-3　成对融合集的置信度求值示意图

实际上，成对融合集置信度大小取决于两对应对象与各自次近邻中较小的那个与其最近邻之间的距离差。例如对象$a \in A$与对象$b \in B$互为最近邻，a_2为$b \in B$在 POI 数据集合A中的次近邻，然后利用公式（3-2）求出成对融合集 $\{a,b\}$ 的置信度值。

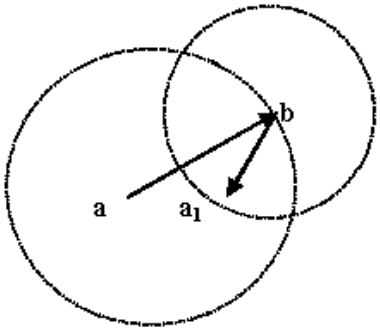


图 3-4　单元素融合集的置信度求值示意图

单元素融合集置信度取决于该元素最近邻对象与其最近邻对象在该元素数据集中最近邻的距离之差，例如 a 的最近邻对象是 $b \in B$，而 $b \in B$ 在 POI 数据集合 A 中的最近邻为 a_1，然后利用公式（3-3）求出单元素融合集 $\{a\}$ 的置信度值。

相互最近邻算法的伪码描述如下：

Input: Datasets **A** and **B**

Output: a set **P** of pairs and a set **S** of singletons

Process:

P ← ∅, **S** ← ∅

Let β be the distance upper bound of **A** and **B**

Let τ be the threshold value

For each **a** ∈ **A** find the nearest neighbor **b** ∈ **B** and second nearest $\mathbf{b_2}$ and for **b** find the nearest neighbor $\mathbf{a_1}$ and the second nearest $\mathbf{a_2}$ do

If **a** equal $\mathbf{a_1}$ then

If distance **{ a , b}** ＜ β then

Produce fusion set **{a, b}**

If confidence (a, b) ＞ τ then

P ← {a, b}

A ← A – {a}

B ← B – {b}

Else if confidence (a, b) ≤ τ then

//a less restrictive approach

Produce **{a} {b}**

S ← {a}, A ← A – {a}

S ← {b}, B ← B – {b}

// a restrictive approach

Discard **{a, b}**

Else if distance **{ a , b}** ＞β then

Produce **{a}**

// here confidence (**{a}**) = 1, and confidence (**{a, b}**) =0 so

S ← {a}

A ← A – {a}

Else if **a** not equal $\mathbf{a_1}$ then

Produce **{a}**

If confidence {a} ＞τ then **S ← {a}, A ← A – {a}**

Else discard **{a}**

At end

S ← S ∪ A ∪ B

Return (**P, S**)

伪码中分为两种方法对小于给定阈值 τ 成对融合集进行处理，分别是严格方法和松散方法，严格方法对置信度在给定阈值 τ 以下的成对融合集直接抛弃，而松散方法对即使置信度在给定阈值 τ 以下的成对融合集也不抛弃而是放入单元素集合中。可以根据不同的目的，人为决定是选择严格方法还是松散方法，算法最终产生一个对集融合集合与一个单元素集合。

使用相互最近邻算法对来自谷歌地图与百度地图的多组 POI 测试数据所生成的成对融合集的表现性能作图如下：

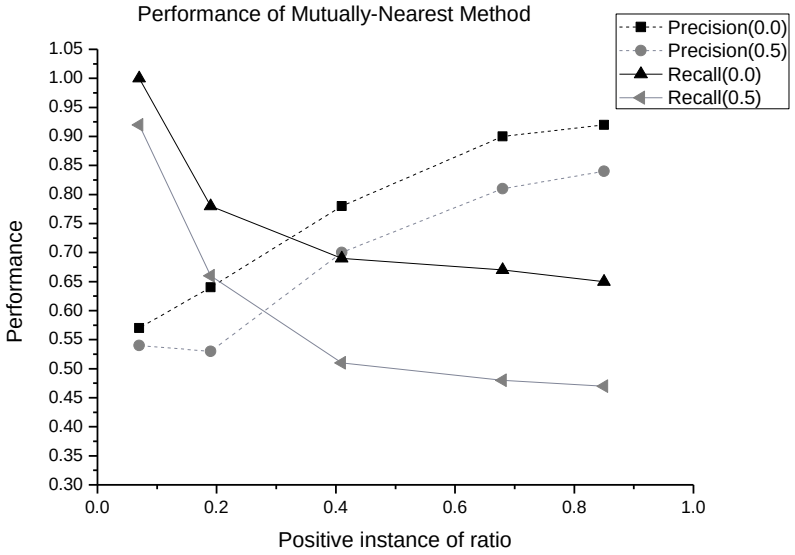


图 3-5 相互近邻算法的融合集中对集的召回率与准确率

经过对相关文献的研究，本文对相互最近邻算法初步处理是：选择具有代表性的阈值参数 0.0 与 0.5 参与初步运算，然后分别用严格方法与松散方法做测试实验，后续章节会给出系统的找到最佳阈值参数的方法与实验结果，从图中可以看出，阈值参数 0.0 要比阈值参数 0.5 的表现好。无论选择 0.0 的阈值参数还是 0.5 的阈值参数，其对集融合集的大致趋势都得到了很好的表现：与传统的片面最近邻连接方法相比，相互最近邻方法的主要优势是对两个 POI 数据集合的重合度不敏感，相互最近邻方法对不同来源的两个数据集的重合度较大时，特别是当一个数据集覆盖另一个数据集时表现良好，即使在两个 POI 数据集重合度不是太大的情况下也能取得不错的结果。然而这种方法只考虑了最近邻和次近邻对象而忽略了其他比较近的对象，所以有它自身的局限性。

3.1.4 基于概率的算法

在基于概率的算法中，POI 对象 $a \in A$ 成为 $b \in B$ 的融合对象的概率大小决定了融合集 $\{a,b\}$ 的置信度值的大小，而概率值的大小取决于两个 POI 对象之间的欧几里得距离。对于每个 $a \in A$，定义概率函数 $P_a : B \to [0,1]$，函数的结果是对象 a 选择对象 $b \in B$ 的概率值；类似地，对数据集 B 中的各个对象 $b \in B$，定义概率函数 $P_b : A \to [0,1]$，函数的结果为对象 b 选择对象 $a \in A$ 的概率值。

对象 a 选择对象 $b \in B$ 与 b 选择 $a \in A$ 的概率值计算公式为：

$$P_{ai}(b_j) = \frac{dis\tan ce(a_i, b_j)^{-\alpha}}{\sum_{k=1}^{n} dis\tan ce(a_i, b_k)^{-\alpha}} \quad \text{式（3-4）}$$

$$P_{bj}(a_i) = \frac{dis\tan ce(b_j, a_i)^{-\alpha}}{\sum_{k=0}^{m} dis\tan ce(bi, a_k)^{-\alpha}} \quad \text{式（3-5）}$$

公式（3-4）与公式（3-5）中参数 α 代表距离衰减因子，其值大于 0，对象 b 选择对象 $a \in A$ 的概率值计算公式定义类似。注意如果两个 POI 对象的距离大于所选定的误差上限 β，认为不可能成为融合集而直接过滤掉。

最终对集融合集 $\{a_i, b_j\}$、单集 $\{a_i\}$ 与 $\{b_j\}$ 的置信度求值函数定义如下：

$$confidence(\{a_i, b_j\}) = \sqrt{P_{ai}(b_j) \cdot P_{bj}(a_i)} \quad \text{式（3-6）}$$

$$confidence(\{a_i\}) = 1 - \sum_{k=1}^{m} \sqrt{P_{ai}(b_k) \cdot P_{bk}(a_i)} \quad \text{式（3-7）}$$

$$confidence(\{b_j\}) = 1 - \sum_{k=1}^{n} \sqrt{P_{bj}(a_k) \cdot P_{ak}(b_j)} \quad \text{式（3-8）}$$

式子中，$P_{ai}(b_j) \cdot P_{bj}(a_i)$ 为 a_i, b_j 互相选择的概率，即 a 与 b 成为 POI 对应对象的概率，对置信度做开平方处理的目的是防止所得到的值过小而不方便进行运算。

概率方法实现过程如下：

步骤一：首先求出空间地理对象 $a \in A$ 选择对象 $b \in B$ 的概率。

$$\begin{bmatrix} P_{a0}(b_0) & \cdots & \cdots & P_{a0}(b_{n-1}) \\ \mathrm{M} & & & \mathrm{M} \\ \mathrm{M} & & P_{ai}(b_j) & \mathrm{M} \\ P_{am-1}(b_0) & \mathrm{K} & \mathrm{K} & P_{am-1}(b_{n-1}) \end{bmatrix} \quad \text{式（3-9）}$$

步骤二：然后再求出空间地理对象 $b \in B$ 选择对象 $a \in A$ 的概率。

$$\begin{bmatrix} P_{b0}(a_0) & \cdots & \cdots & P_{b_{n-1}}(a_0) \\ \mathrm{M} & & & \mathrm{M} \\ \mathrm{M} & & P_{bj}(a_i) & \mathrm{M} \\ P_{b0}(a_{m-1}) & \mathrm{K} & \mathrm{K} & P_{b_{n-1}}(a_{m-1}) \end{bmatrix} \quad \text{式（3-10）}$$

步骤三：对（3-9）和（3-10）的各个对应元素做乘积然后开平方运算，最后添加额外的一行和一列分别存放空间地理对象 $a \in A$ 和 $b \in B$ 的单集概率。

$$\begin{bmatrix} \sqrt{P_{a0}(b_0)\cdot P_{b0}(a_0)} & \cdots & \cdots & \sqrt{P_{a0}(b_{n-1})\cdot P_{b_{n-1}}(a_0)} \\ \mathrm{M} & & & \mathrm{M} \\ \mathrm{M} & & \sqrt{P_{ai}(b_j)\cdot P_{bj}(a_i)} & \mathrm{M} \\ \sqrt{P_{a_{m-1}}(b_0)\cdot P_{b0}(a_{m-1})} & \mathrm{K} & \mathrm{K} & \sqrt{P_{a_{m-1}}(b_{n-1})\cdot P_{b_{n-1}}(a_{m-1})} \end{bmatrix} \quad \text{式（3-11）}$$

额外的最后一行（m-1）和组后一列（n-1）放入值为

$$1-\sum_{k=0}^{m-1}\left(\sqrt{P_{bj}(a_k)\cdot P_{ak}(b_j)}\right) \quad \text{式（3-12）}$$

$$1-\sum_{k=0}^{n-1}\left(\sqrt{P_{ai}(b_k)\cdot P_{bk}(a_i)}\right) \quad \text{式（3-13）}$$

最后把第 m 行与第 n 行交叉的元素值设为 0，也就是最后一个元素值设为 0。

使用基于概率的算法对来自谷歌地图与百度地图的多组 POI 测试数据测试所生成的成对融合集的表现性能作图如下：

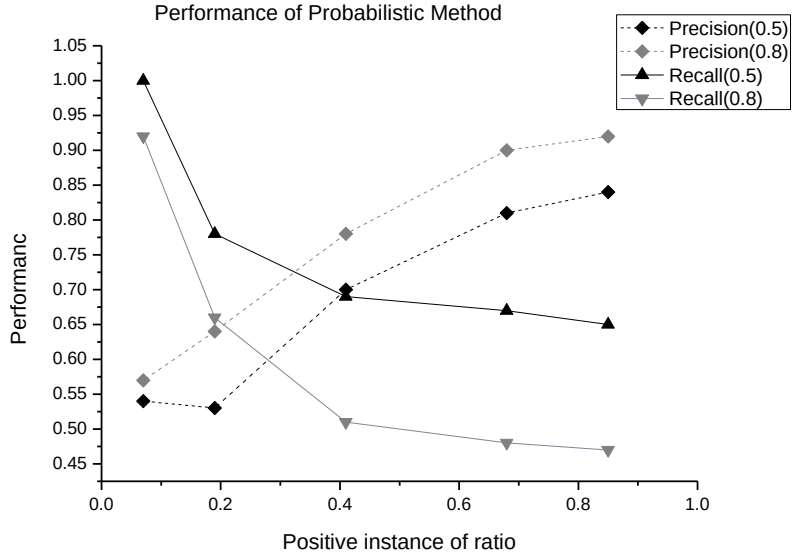


图 3-6　基于概率算法的融合集中对集的召回率与准确率

对于概率方法，本文选择具体有代表性的两阈值参数值 0.5 与 08，从图中可以看出，阈值参数选择 0.5 要比 0.8 产生的结果表现好。这也可以说明 0.5 更接近最佳阈值参数的位置，无论哪种选择，其趋势都可以明显的表现出来，即与前两种方法相比，概率算法对两个 POI 数据集合的重合度要求更低，也就是说在中等重合度下也能取得较好结果。

另外，当数据密度比较大时，概率方法要比相互最近方法和片面最近邻连接方法表现更好，这是因为概率方法为每对对象都设定一个可信度值。概率方的另一个优势是可以通过调低阀值来增大召回率。但是这样做可能导致一个对象出现在多个融合集中。总之，当 POI 数据集合之间重合度较小时，概率方法就不能正确地处理应该在单元素集合里的对象。因此，这种方法只在数据集重合度较大的情况下使用。

3.1.5 标准化权重算法

标准化权重方法其实是概率方法的一种改进后的变型，这种方法利用上面概率函数为每个可能的融合集设置权重值（置信度值），标准化权重方法使用一种迭代算法对这些初始权重值做标准化处理。

利用公式(3-4)、(3-5)、(3-6)定义标准化权重方法如下：

$$\mu_{ij}=\begin{cases} P_{a_i}(b_j)\cdot P_{b_j}(a_i) & :1\le i\le m,1\le j\le n \\ \prod_{k=1}^{n}(1-P_{b_k}(a_i)) & :1\le i\le m,\ j=n+1 \\ \prod_{k=1}^{m}(1-P_{a_k}(b_j)) & :i=m+1,1\le j\le n \\ 0 & :i=m+1,\ j=n+1 \end{cases} \qquad \text{式（3-14）}$$

首先定义一个匹配矩阵 M，M 为(m+1)×(n+1)的矩阵用于存放结果，公式中，$P_{a_i}(b_j)\cdot P_{b_j}(a_i)$ 为 a_i,b_j 互相选择的概率，μ_{ij} 为第 i 行第 j 列的元素值，其实际意义是 a_i,b_j 互相选择的概率，每一行的最后一列为 a 不被 b 选中的概率，每一列的最后一行为 b 不被 a 选中的概率，然后通过迭代对矩阵归一化，矩阵内数值代表该位置对应的 POI 对象成为融合集的置信度。

下面介绍标准化权重方法的改进版本，在该方法中需要计算出所有可能的成对融合集的概率，也就是所说的权重值，这里不仅要计算出所有成对融合集的权

重值还要计算出单集的权重值，最后对各个权重值组成的矩阵做迭代运算，目的是模拟对象之间相互选择的影响。然后对所求的各个初始化权重值做出判断，如果权重值大于给定的阈值参数，那么这个权重值对应的两个空间地理对象就认为是正确的成对融合集，否则直接过滤掉。算法的最重要的改进是在选择概率定义中考虑了误差上限的影响。

设定两个不同来源的空间地理数据集合 A 和 B，则两集合定义为 A = $\{a_1, \cdots\cdots, a_m\}$，B = $\{b_1 \cdots\cdots, bn\}$，对于集合 A 的每个空间地理对象 $a \in A$，我们定义对象 a 选择对象 $b \in B$ 的概率函数为 $P_a : B \to [0,$；类似地，对于集合 B 的每个空间地理对象 $b \in B$，定义对象 b 选择 $a \in A$ 的概率函数为 $P_b : A \to [0,1]$。则改进后的概率函数 P_{ai}，P_{bj} 定义如下：

$$P_{ai}(b_j) = \frac{dis\tan ce(a_i, b_j)^{-\alpha}}{\sum_{k=1}^{n} dis\tan ce(a_i, b_k)^{-\alpha} + \beta_{AB}{}^{-\alpha}} \quad \text{式（3-15）}$$

$$P_{bj}(a_i) = \frac{dis\tan ce(a_i, b_j)^{-\alpha}}{\sum_{k=1}^{m} dis\tan ce(b_j, a_k)^{-\alpha} + \beta_{AB}{}^{-\alpha}} \quad \text{式（3-16）}$$

公式（3-15）、（3-16）中，参数 α 表示距离衰减因子，参数 β_{AB} 表示空间数据集合 A 与 B 的误差上限。

下面说明 POI 对象 a_i 不选择任何对象$b \in B$的概率函数定义，也就是所谓的单集定义。类似的也存在这样的 POI 对象 b_j，分别用 $P_{ai}(\perp_B)$ 与 $P_{bj}(\perp_A)$ 表示，其定义如下：

$$P_{ai}(\perp_B) = \frac{\beta_{AB}{}^{-\alpha}}{\sum_{k=1}^{n} dis\tan ce(a_i, b_k)^{-\alpha} + \beta_{AB}{}^{-\alpha}} \quad \text{式（3-17）}$$

$$P_{bj}(\perp_A) = \frac{\beta_{AB}{}^{-\alpha}}{\sum_{k=1}^{m} dis\tan ce(b_j, a_k)^{-\alpha} + \beta_{AB}{}^{-\alpha}} \quad \text{式（3-18）}$$

在对集融合集的求解中如果 $dis\tan ce(a_i, b_j) > \beta_{AB}$，那么就认为 POI 对象 a_i 与 b_j 成为对应对象的概率就位 0，即定义此时的 $P_{ai}(b_j) = 0$。从公式

（3-15）中，可以看出当对象 a_i 与 b_j 之间的距离减小时，对象 a_i 选择对象 b_j 的概率就增大，这里距离衰减因子 α 大于 0。而距离衰减因子 α 的作用是当对象 a_i 与 b_j 之间的距离增大时来决定概率的缩小比例，经过考察，本文选择的衰减间距因子大小为 2。

下面对基于以上改进的表转化权重方法重新定义，首先创建一个匹配矩阵 M ，矩阵大小为 $(m+1)\times(n+1)$ ，这样 μ_{ij} 代表第 i 行第 j 列的值大小，其定义如下：

$$\mu_{ij}=\begin{cases} P_{a_i}(b_j)\cdot P_{b_j}(a_i) & :1\le i\le m,1\le j\le n \\ P_{a_i}(\perp_B)\cdot\prod_{k=1}^{n}(1-P_{b_k}(a_i)) & :1\le i\le m,\ j=n+1 \\ P_{b_j}(\perp_A)\cdot\prod_{k=1}^{m}(1-P_{a_k}(b_j)) & :i=m+1,1\le j\le n \\ 0 & :i=m+1,\ j=n+1 \end{cases} \quad \text{式（3-19）}$$

在公式（3-19）中，第一种情况表示对象 a_i 与 b_j 之间互相选择为对应对象的概率；第二种情况表示 a_i 不选择 $b\in B$，同时也不被 $b\in B$ 选择为对应对象的概率；第三种情况类似，表示对象 b_j 不选择任何对象 $a\in A$，同时也不被任何对象 $a\in A$ 选择为对应对象的概率；第四种情况是设置最后一个元素的值为 0，这样不影响计算结果。

下面给出标准化或者成为归一化的过程描述，如果一行或者是一列数据 r 的所有元素的大小之和 s 等于一个值 χ ，且 $\chi>0$，那么就表示 r 标准化或者归一化到 χ，我们可以通过对数据序列 r 的各个元素与 s/χ 做除法运算的途径，就可以使得 r 归一化到 χ。对于矩阵 M ，完整的标准化过程包括一系列有顺序的迭代运算，对于矩阵 M 的前 m 行数据标准化到值 1，而最后一行标准化到的值等于在数据集 B 中没有对应对象的 $a\in A$ 的个数。然后对列做处理，类似地，前 n 列数据序列标准化到值 1，而最后一列所标准化到的值等于在数据集 A 中没有对应对象的 $b\in B$ 的个数。如果对最后一行和最后一列要标准化的值未知的话，那就使相互最近方法的结果值进行估计。

如果用 $M^{(0)}$ 表示初始矩阵 M ，根据上面所说的标准化过程，用 $M^{(k)}$ 表示矩阵 M 经过 k 次迭代后的结果。根据 sinKhorn (1964, 1967) 等人的研究结果可知，当要进行标准化的矩阵的所有元素值都为正值的情况下，标准化的最终结果就跟行和列的每次迭代顺序无关。我们给出迭代运算停止的条件，那就是除了最后一行和最后一列，当每行或者每列之和与 1 差别某个较小值 $\varepsilon > 0$ 时，矩阵 $M^{(0)}$ 的标准化迭代运算结束。

用矩阵 $M^{(t)}$ 表示矩阵 $M^{(0)}$ 的最终标准化运算结果，则可能的融合对集 $\{a_i, b_j\}$ 的置信度大小就是矩阵结果 $M^{(t)}$ 的第 i 行第 j 列的值。而单集 $\{a_i\}$ 的置信度大小就等于矩阵结果 $M^{(t)}$ 的第 i 行第 $n+1$ 列的值；类似地，单集 $\{b_j\}$ 的置信度大小就等于矩阵结果 $M^{(t)}$ 的第 $m+1$ 行第 j 列的值。最终标准化的运算结果集包括所有置信度值大于给定阈值参数 τ 的结果融合集，而其中阈值参数由用户给出，阈值参数选取的要求是：结果融合集有较高的准确率同时也有较好的召回率。

使用标准化权重算法对来自谷歌地图与百度地图的多组 POI 测试数据测试所生成的成对融合集的表现性能作图分析，如下：

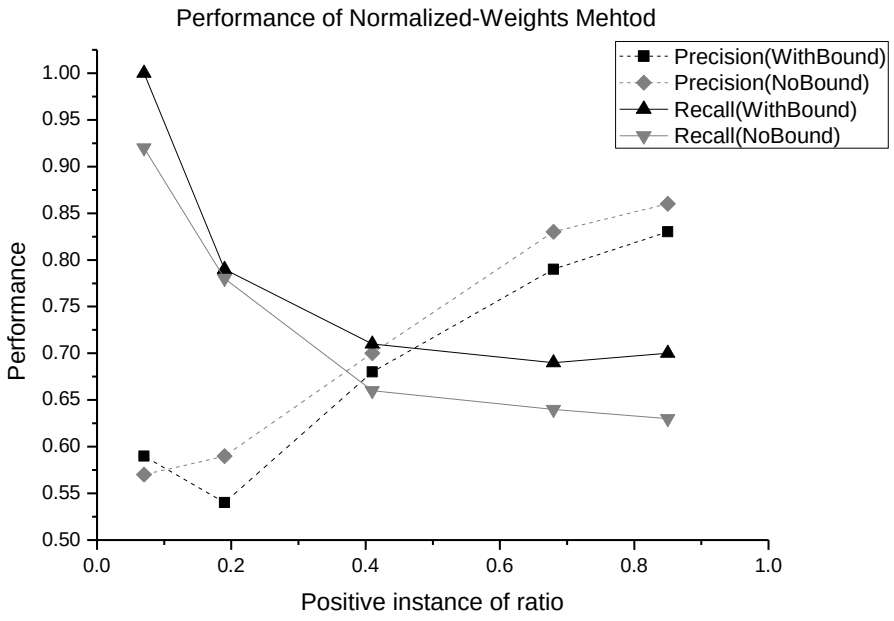


图 3-7　标准化权重方法的融合集中对集的召回率与准确率

标准化权重方法的改进算法在求权重值的时候考虑到误差上限的影响，在初步选定阈值参数为 0.5 后，分别针对标准化权重方法及其变型做测试实验。从结果图中可以看出，标准化权重方法的改进算法结果表现较好些，但是无论哪种方法都可以呈现出标准化权重方法的性能趋势，另外在合适阈值参数的最佳位置处，标准化权重方法的结果融合集的准确率与召回率都较高。

3.2 基于非空间属的技术

通常情况下，能成为对应对象的 POI 数据都有高度相似的非空间数据特征属性，比如名称、地址以及电话号码等等。所以 POI 的非空间特征属性相似度也是除了空间位置信息最重要的衡量标准，而相关研究者已经提出了很多且成熟的字符串匹配算法，本文主要介绍三种常见的字符串相似度算法：*Levenshtein distance* 算法、*Jaro distance* 算法与 *Jaro−Winkler distance* 算法。

3.2.1 文莱斯特距离算法

Levenshtein distance 算法是一种字符串编辑距离算法，主要目的是衡量两个字符串序列之间的差异程度，该算法的核心思想是一个字符串通过最少的操作次数得到另外一个字符串，这些操作包括单个字符的插入、删去或者取代等[36]。*Levenshtein distance* 算法是由俄国科学家 *Levenshtein* 在 1965 年提出的。

Levenshtein distance 算法数学定义如下：设有两字符串 a 和 b，$lev_{a,b}(|a|,|b|)$ 为两字符串的计算结果，其意义为两字符串最终相似度。则有

$$lev_{a,b}(i,j)=\begin{cases}\max(i,j) & ,\min(i,j)=0\\ \min\begin{cases}lev_{a,b}(i-1,j)+1\\ lev_{a,b}(i,j-1)+1\\ lev_{a,b}(i-1,j-1)+[a_i\neq b_j]\end{cases} & ,else\end{cases} \quad \text{式（3-20）}$$

在上式中，最小值计算中第一个因子对应从字符串 a 到 b 的删除操作计算，第二个因子和第三个因子分别对应插入和是否匹配操作计算，本文用 getLevenshtein 表示该算法。

3.2.2 哈罗-温克勒距离算法

Jaro−Winkler distance 算法也是用来计算两个字符串之间的相似度，该算

法是 $Jaro\ \ distance$ 算法的一种变型，如果两个字符串的 $Jaro-Winkler\ \ distance$ 值很高，则表明这两字符串的相似度很高，这种算法比较适合例如名称这样较短字符串之间相似度的计算[37-40]。结果值 0 代表完全不相似，1 代表完全相似也就是匹配。

两字符串 s_1, s_2 之间的 $Jaro\ \ distance$ 结果值 d_j 定义如下：

$$d_j = \begin{cases} 0 & if \quad m = 0 \\ \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) & otherwise \end{cases} \qquad \text{式（3-21）}$$

上式中，s_1, s_2 是要进行对比的两字符串，d_j 最后计算的相似度值，m 是匹配的字符数，t 是换位数目。

$Jaro-Winkler\ \ distance$ 算法具体定义如下：

$$d_w = d_j + (lp(1-d_j)) \qquad \text{式（3-22）}$$

公式中，d_j 是 $Jaro\ \ distance$ 算法针对字符串 s_1, s_2 的结果值，l 是前缀部分匹配的长度值，p 是一个范围因子常量，用来调整前缀匹配的权值，但是 p 的值不能超过 0.25，如果不这样处理最后的结果值有可能超过 1。本文分别用 getJaro 和 getJaroWinkler 代表这两种方法。

4 POI 数据融合技术的改进

4.1 基于空间位置的 POI 融合改进方案

本文提出的 POI 数据融合的改进方案要解决的主要问题是：如何从不同来源的 POI 数据集合中准确找出对应对象，这些对应对象进一步形成融合集，最终实现 POI 信息的融合。改进方案的详细过程为：首先对两个 POI 数据集合实施空间位置方法找出对应对象组成的初步融合集，然后使用低阈值的名称特征属性相似度方法排除由空间位置方法找出的错误对应对象，最后使用高阈值的名称特征属性相似度方法找出空间位置方法未能找出的对应对象。方案的详细描述如下：

Input: Databases G and B

Output: A set P of pairs and a set S of singletons

1: (GS, BS, P) ← Φ(G, B)

2: **for** each {g, b} ∈PL such that η (g.title, b.title) < γ **do**

P← P－ {g , b}

(GS, BS) ← {g, b}

3: **for** each {g} ∈GS and {b} ∈BS **if** μ (g.title, b.title) > τ **do**

P ← {g, b}

GS ← GS － {g}

BS ← BS － {b}

4: S ← GS ∪ BS

5: **return** (P, S)

1、对 POI 集合 G 和 B 实施空间位置方法，得到一个初步融合集 P 和两个无对应项的单集 GS、BS。

2、对于融合集 P 中的每个对应项 {g, b},如果对象 g 和 b 的名称属性相似度小于给定阈值 ɣ ，就从融合集 P 中删去{g, b},然后把对象 g 和 b 分别加入到单集 GS 和 BS 中， η、ɣ 分别为初次过滤函数和给定阈值 。

3、对于余下的每个单集对象{g}∈GS 和{b} ∈BS，如果对象 g 和 b 的名称相似度大于给定的阈值 τ，就把{g, b} 添加到融合对集 P 中，然后再把 g、b 分别从单集 GS、BS 中去除， μ 、 τ 分别为二次过滤函数和给定阈值 。

4、最后，剩余单集对象 GS、BS 的并集作为最终单集元素。算法最终结果为

一个融合对集 P 与一个最终单集集合 S。

在基于空间位置的技术中，本文分别选取相互最近邻方法、概率方法、标准化权重方法，在接下来的内容中给出比较，从而选取最佳方法。在 POI 数据融合改进方案中，基于位置的算法阈值的选取都维持在较低值（大概 0.4 左右），这是为了尽可能的将误差限（本文选取 100 米）内可能的对应项都放进融合对集。因为数据来源不同，其对应项的名称不一定完全相同，但相似的可能性是很大的。来源不同且相邻的对象名称相似度大于给定的阈值就认为是对应对象，首次过滤相似度阈值也维持在较低值，这样就避免漏掉因为对应对象相似度较低而被过滤掉的情况。而在对剩余对象的二次过滤时，名称相似度算法的阈值就选取较高值（0.9 左右），这是因为没有距离约束只有名称高度相似时才被认为是对应对象。

基于空间位置的 POI 数据融合改进方案所用的实验数据抽取自不同网络电子地图的 POI 数据，具体到本文，所选取的网络电子地图为具有代表性的谷歌地图和百度地图，然后用已经存在的具有代表性的基于空间位置算法对实验数据进行测试找出适用于本数据集的最佳方法以及相关阈值参数，最后再用非空间属性相似度算法进行相似处理。

基于空间位置的 POI 数据融合改进方案的处理流程图如下：

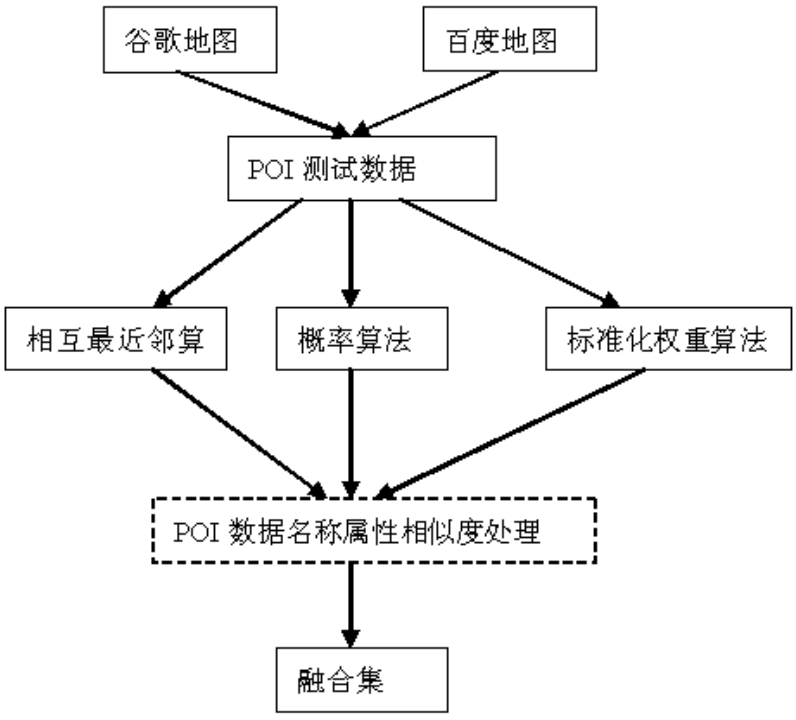


图 4-1　基于空间位置的 POI 数据融合改进方案处理流程图

如图所示，根据前面各个算法的初步测试结果，我们发现单向片面最近邻连接算法要求条件过于苛刻，对空间位置改进后的算法中的空间位置算法部分我们选定相互最近邻算法、概率算法以及标准化权重方法。然后从这三个方法中通过实验测试以及结果分析再最终决定选择哪种算法，对于位置算法处理过后得到的初步融合结果集合再用 POI 名称特征属性相似度处理，跟空间位置算法的选定方法类似，我们也是通过实验测试与结果分析最终选择对于本文所采用的 POI 数据表现最佳的算法，接下来的内容将注重解决最佳算法及最佳阈值参数的选择问题。

本文所提出的 POI 数据融合技术改进方案的具体实现过程如下：

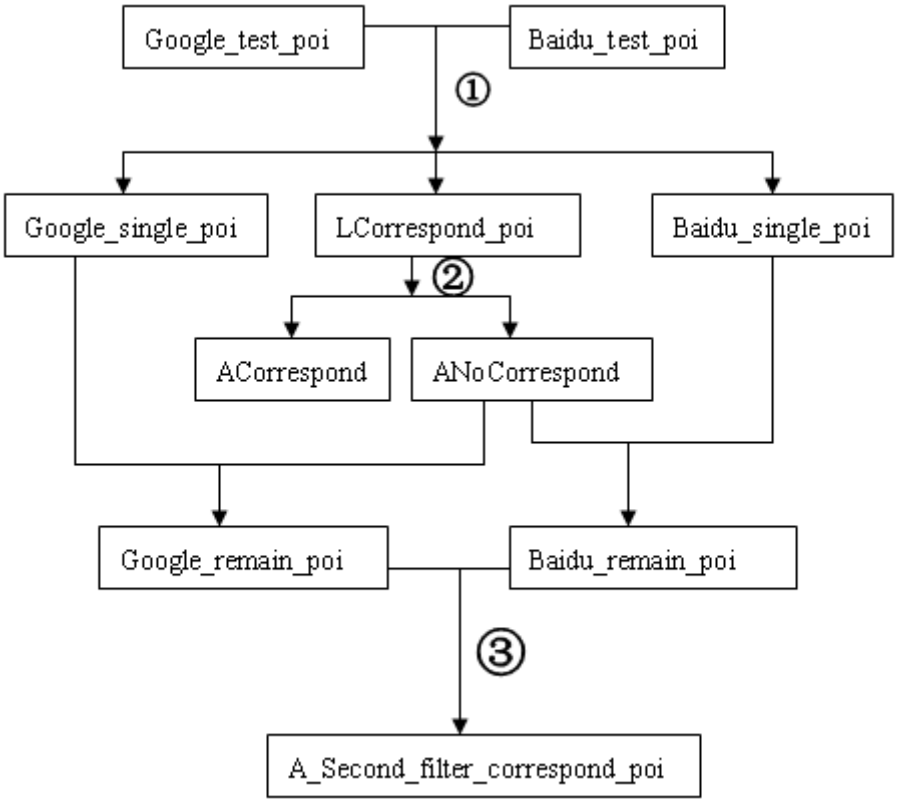


图 4-2　POI 数据融合技术改进方案具体实现过程图示

过程说明：

① 对测试数据集 google_test_poi 和 baidu_test_poi 运用空间位置算法（本文最终选择的是标准化权重方法）。

② 对步骤 ① 中得到的对应融合结果集中，利用 title 属性字段相似度分开对应项和非对应项（也就是正确对应项和错误对应项的初步分离）。

③ 用找出的错误对应对象和步骤 ① 中得到的谷歌和百度单集组合成二次过滤源，再用 title 属性字段过滤，得到剩余对应项。最后 ACorrespond 和 A_Second_filter_correspond_poi 组合成最终结果融合集 correspond_final_pois。

4.2 组织 POI 测试数据集合

组织 POI 测试数据集合并进行测试的目的是通过测试和比较空间位置算法和非空间名称属性相似度算法，选择最佳空间位置算法和名称属性相似度算法及最佳阈值参数，最后把通过实验找出的最佳算法和阈值参数用于本文所提出的基于空间位置的 POI 数据融合改进方案中。

用于测试算法的 POI 实验数据集合分别来自谷歌地图 *POI* 数据集合和百度地图 POI 数据集合，数据集合 *G* 表示来自谷歌地图的 *POI* 数据，数据集合 *B* 表示来自百度地图的 *POI* 数据，其中每个对象都有名称属性、空间地理坐标（经度和纬度）以及地址等特征信息。两 *POI* 数据集合各有 500 个对象，其总共代表的真实地理实体个数为 724 个，反例 448 个，正例 276 个，所谓的正例指的是两实验数据集合中成为对应对象的空间地理对象，反例指的是无对应对象的 POI 对象。两个数据集合中的每个 POI 对象都唯一地代表地图上一个真实的地理实体。有关实验数据集合生成细节请参照第二章节。

表 4-1 来自谷歌地图和百度地图的 POI 实验数据

数据集来源	实体数	正例	反例
谷歌地图	500	275	225
百度地图	500	275	225

4.3 测试结果评价标准

本文用国际上比较权威且通用的准确率、召回率和 F1 值作为衡量算法结果质量的评价标准。由于在数据预处理过程中，两个 POI 数据集合中的对应对象分配了相同的 ID 号，这样就预先知道了来源不同的两个 POI 是否代表同一个地理实体，进而就可以确定融合集中的对应对象是否正确。这些先验知识可以用来定

义评价标准，并求出评价标准的具体值。

准确率是指结果融合集中正例所占的比例，其定义如下：

$$准确率=\frac{融合集中正例数}{融合集中对象总数} \quad 式（4\text{-}1）$$

召回率定义为结果融合集中正例个数占实验数据集合中总正例数的比例：

$$召回率=\frac{融合集中正例数}{实验数据集正例总数} \quad 式（4\text{-}2）$$

$F1$值为召回率与准确率的调和平均数，其定义如下：

$$F1=\frac{2\times召回率\times准确率}{召回率+准确率} \quad 式（4\text{-}3）$$

对于特定算法，阈值降低，召回率增大，准确率减小；反之召回率减小，准确率增大。一般情况下，准确率和召回率之间是种反向关系，在本文中，最佳阈值是指在可以得到较高的召回率，同时也可以得到较高的准去率。

4.4 最佳阈值参数选取测试

4.4.1 基于空间位置算法测试

本章节仅用基于空间位置的方法找出结果融合集，在这种情况下，如果所得到的两个数据集中 POI 对象的置信度高于给定阈值参数，则可以认为这两个 POI 对象为对应对象即可以放入结果融合集，否则不能放入结果融合集。这里算法所涉及的参数有误差上限、距离衰减因子和给定的阈值参数，前两个参数依次为 100 和 2，具体解释见文献[14]与文献[23]。所进行的实验主要解决选取适合本文所用的 POI 实验数据集合的各个空间位置方法（除相互最近方法外）的最佳阈值问题。阈值采用 0.05 为初始值然后依次增大 0.1 直到 0.85 看其结果融合集的准确率、召回率和 F1 值，本文只考虑融合对集。各方法的实验结果图如下：

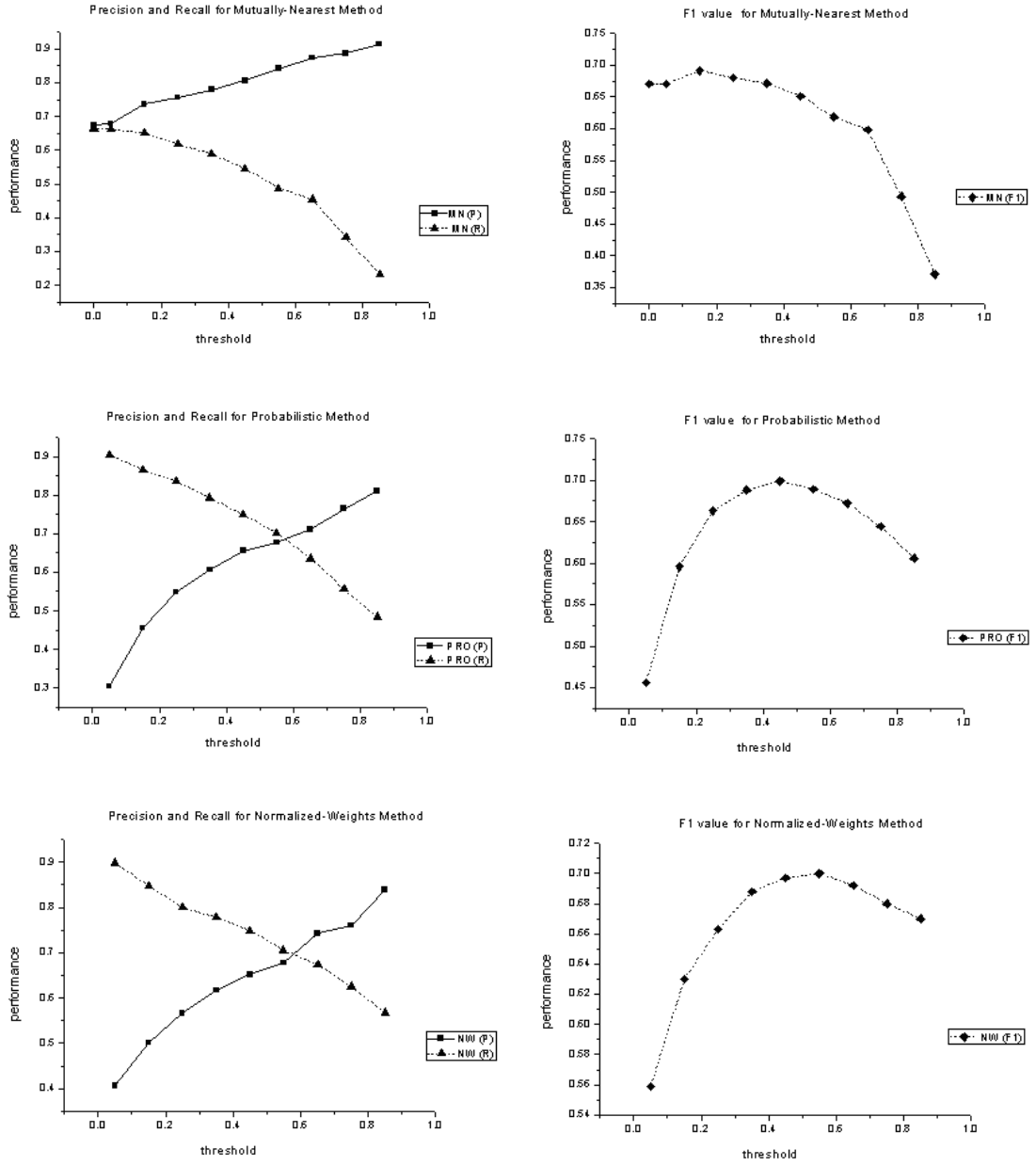


图 4-3　上图依次为相互最近方法、概率方法和标准化权重方法的实验结果，在第一列栏目中直线代表结果融合对集的准确率，虚线代表召回率，第二列为相应的 F1 值。

从上图中可以看出结果融合集的准确率和召回率成反比关系，我们结合 F1 值选择最佳阈值，使结果融合集既有较高的准确率又有较高的召回率，经过比较最后为这三种方法选择的最佳阈值分别为如下：

表 4-2　位置方法的最佳阈值选取

位置方法	选定阈值
相互最近方法	0.0
概率方法	0.6
标准化权重方法	0.6

4.4.2 名称属性相似度算法测试

本小节主要目的是为名称相似度方法选择最佳阈值，使结果融合集具有较高的准确率和召回率。实验过程与空间位置方法类似，阈值采用 0.1 为初始值然后依次增大 0.1 直到 1 看其准确率、召回率和 F1 值，结果只考察融合对集。实验结果图如下：

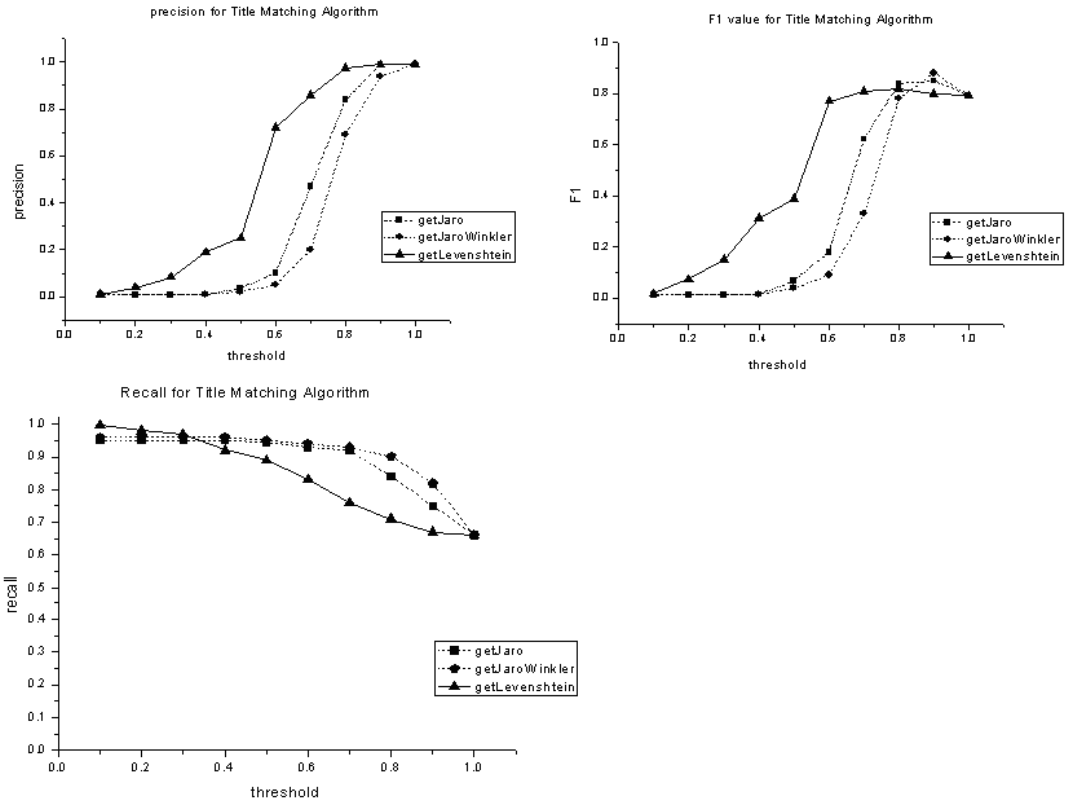


图 4-4　第一列图表示三种名称相似度方法的准确率与召回率，第二列图表示其各自对应的 F1 值。三种名称相似度方法依次为 getJaro 方法、getJaroWinkler 方法和 getlevenshtein 方法。

从上图中可以看出，名称相似度算法只有在阈值较高时其结果的准确率和 F1 值才能表现出不错的结果，由此可以看出如果没有距离的约束，名称高度相似的地理对象才被人为是可以融合的对应对象，而那些由于不同来源导致的同一地理实体采用的名称相似度不是很高的 POI 对象就被认为不是要融合的对应对象，显然，这样看来算法的可信度就大大降低了。结合上图准确率和 F1 值我们给出各个方法的最佳阈值如下：

表 4-3　非空间属性相似度算法最佳阈值选取

名称相似度方法	选定阈值
getJaro	0.8
getJaroWinkler	0.9
getlevenshtein	0.6

4.4.3 基于空间位置改进方案测试

通过对比以上各个方法的实验结果，最终选择出表现最佳的方法和其对应的阈值参数参与到本文所提出的改进方案中，实验数据为本章开始是提供的来自谷歌地图和百度地图的 *POI* 实验数据集合，用前文给出的基于位置方法和名称相似度方法以及其对应的最佳阈值参数做实验，对比结果图如下：

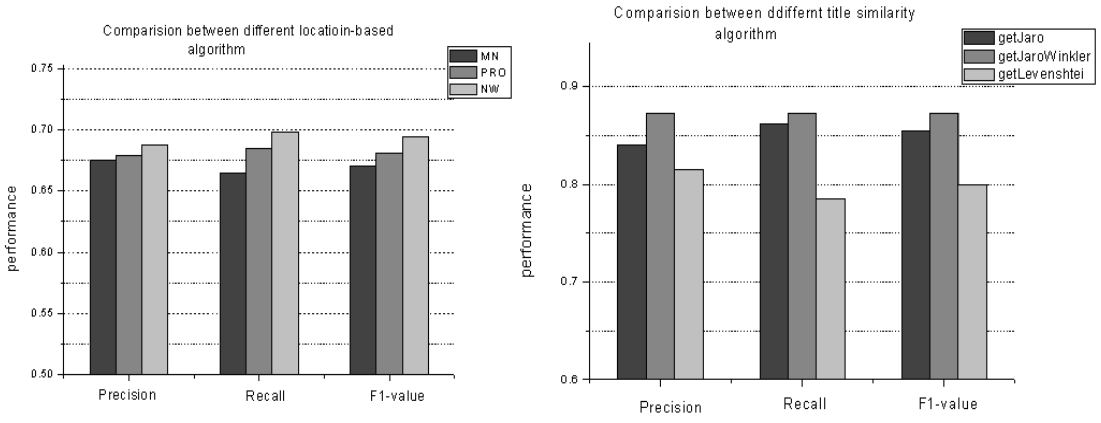


图 4-5　左图为基于位置的相互最近方法、概率方法和表转化权重方法的准确率、召回率和 F1 值，右图为三种名称相似度方法的准确率、召回率和 F1 值。

从上图看以看出，名称相似度方法的表现普遍优于基于位置方法，而名称相似度表现最好的是 getJaroWinkler 方法，基于位置的方法中则是标准化权重方法

表现最好。所以本文基于位置的算法选择标准化权重方法，名称相似度方法则选择 getJaroWinkler 方法。

下面根据前文给出的算法描述来实现基于空间位置的 POI 融合改进方案，其基本思想是先用基于位置的方法最大限度找出两 POI 数据集中有可能是对应对象的对象放入融合对集（也就是最大可能地增大召回率），这就要求位置方法的阈值不要太高，经过多次试验得出阈值设置为 0.4 为最佳。然后对初步的出的融合对进行名称相似度的首次过滤，在这一步中，首先保证尽可能把相似的对象认为是对应对象，因为距离相近且名称相似的对象成为融合对的可能性较大，经过试验比较，首次过滤名称相似度方法选择 getLevenshtein 方法最佳其最佳阈值为 0.3 左右。最后对剩下的 POI 集合做名称相似度的二次过滤，由于没有了距离约束其方法选择要严格且阈值要高，本文选取 getJaroWinkler 方法，阈值设置为 0.9。新方法用 COM－NWT 表示。不同方法之间的比较结果如下：

表 4-4 不同方法之间的比较

方法	准确率	召回率	F1 值
NW	0.68	0.7	0.69
GetJaroWinkler	0.87	0.87	0.87
COM-NWT	0.94	0.94	0.94

上表可以看出改进方法的表现明显优于上面两种方法，这仅是在单个数据集下得出的实验结果，接下来将给出多组 POI 数据集合的验证结果。

4.5 组织不同差异程度的多数据集合

按照前文 *POI* 数据预处理过程，组织不同差异程度的多组 *POI* 实验数据集合，分别选定谷歌地图 *POI* 数据集合与百度地图 *POI* 数据集合作为实验备用数据集合。两个用于实验的测试 *POI* 数据集合各包含 450 个空间地理对象，每个 POI 对象都唯一地代表一个真实地理实体，即所有 POI 对象没有重复项。

具体操作流程如下图所示：

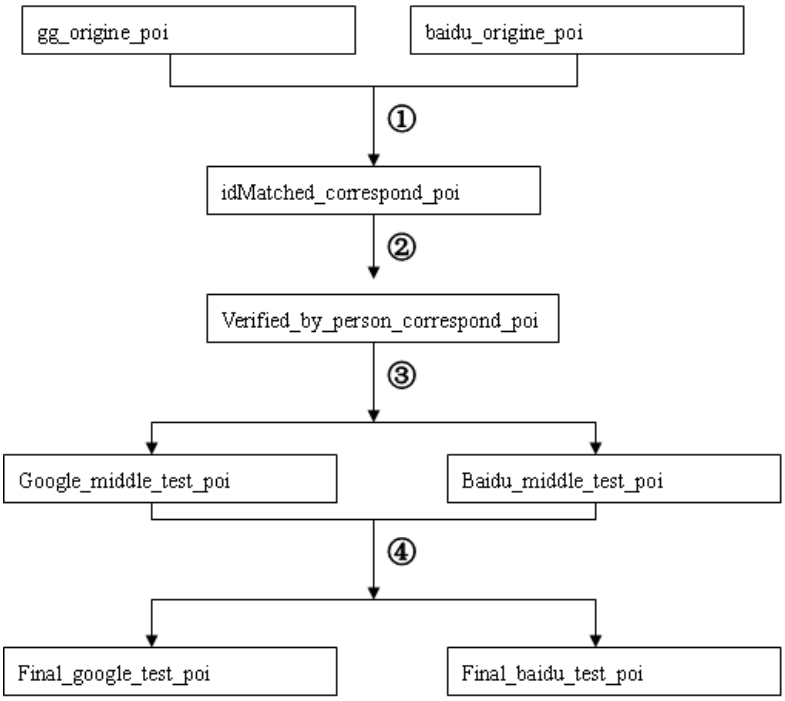


图 4-6　不同差异程度的 POI 实验数据生成流程图

测试数据的组织方案流程图步骤说明：

①、首先在 *MySql* 环境下创建 *poidata* 数据库，数据库中分别包含从谷歌地图与百度地图中抽取的 *POI* 数据，这些数据分别导入表 baidu_origine_poi 和 gg_origine_poi 表依据 *id* 找出可能的对应项集合。

②、依据 *id* 字段把找出的初步有可能成为对应对象结果集合导入到 excel 表中，通过人工判断所得到对应项集合的非空间数据属性标注是否为真实对应项（如果是加标为 1 ，否则为 0）。

③、筛选出人工标注为对应项的集合，再分别导入到两数据库表 Google_middle_test_poi 和 Baidu_middle_test_poi 中。

④、从两数据库表 Google_middle_test_poi 和 Baidu_middle_test_poi 中选取 450 个 *POI* 测试对象，最终选定七组同样大小规模的测试数据，但是每组数据中正例和反例比例不同，这就实现了所谓的不同差异程度（即不同重合度），这一过程可以是认为选定也可以随机选取也可以用 *k – fold* 方法（学习算法中的一种测试数据分类方法）。在本文中，正例

代表在两个 POI 数据集中都出现的对象也就是要找出的融合对集，反例代表只在其中一个数据集合中出现的对象。不同差异程度的多数据集合构造如表所示：

表 4-5　不同重合度测试数据集

实体总数	正例数	反例数	正例比例
800	100	700	0.2
750	150	600	0.3
700	200	500	0.4
650	250	400	0.5
600	300	300	0.6
550	350	200	0.7
500	400	100	0.8

4.6 多组 POI 数据集合验证改进方案

下面就用空间位置算法中表现最佳的标准化权重算法、名称相似度算法中表现最佳的 getJaroWinkler 算法和本文给出的空间位置改进算法对上面多组不同差异程度的实验数据集合进行测试，然后分别用准确率、召回率和 F1 值等标准对测试结果融合集合做出评价，其中求得的部分结果融合集的准确率、召回率和 F1 值如下：

表 4-6　800 实体，反例 700，正例 100 测试结果

方法 评价	NW	COM-NW	Title-Similarity
对集准确率	0.73	0.92	0.77/0.83
对集召回率	0.62	0.90	0.92/0.93
对集 F1 值	0.67	0.91	0.84/0.88
单集准确率	0.97	0.97	
单集召回率	0.97	0.99	
单集 F1 值	0.97	0.98	
综合准确率	0.95	0.97	
综合召回率	0.93	0.97	
综合 F1 值	0.94	0.97	

表 4-7　700 实体，反例 500，正例 200 测试结果

方法 评价	NW	COM-NW	Title-Similarity
对集准确率	0.81	0.95	0.86
对集召回率	0.67	0.92	0.89
对集 F1 值	0.73	0.94	0.87
单集准确率	0.93	0.94	
单集召回率	0.96	0.98	
单集 F1 值	0.94	0.96	
综合准确率	0.9	0.94	
综合召回率	0.88	0.96	
综合 F1 值	0.89	0.95	

表 4-8　600 实体，反例 300，正例 300 测试结果

方法 评价	NW	COM-NW	Title-Similarity
对集准确率	0.86	0.97	0.89
对集召回率	0.68	0.90	0.84
对集 F1 值	0.76	0.93	0.86
单集准确率	0.83	0.82	
单集召回率	0.94	0.99	
单集 F1 值	0.88	0.90	
综合准确率	0.84	0.90	
综合召回率	0.81	0.94	
综合 F1 值	0.83	0.91	

表 4-9　500 实体，反例 100，正例 400 测试结果

方法 评价	NW	COM-NW	Title-Similarity
对集准确率	0.88	0.98	0.93
对集召回率	0.72	0.90	0.85
对集 F1 值	0.79	0.94	0.89
单集准确率	0.57	0.55	
单集召回率	0.83	0.97	
单集 F1 值	0.67	0.71	
综合准确率	0.78	0.85	
综合召回率	0.74	0.92	
综合 F1 值	0.76	0.88	

其中标准化权重方法、基于名称相似度的方法和本文提出的空间位置改进方法的结果融合集中对集的准确率、召回率与 F1 值对比图示如下：

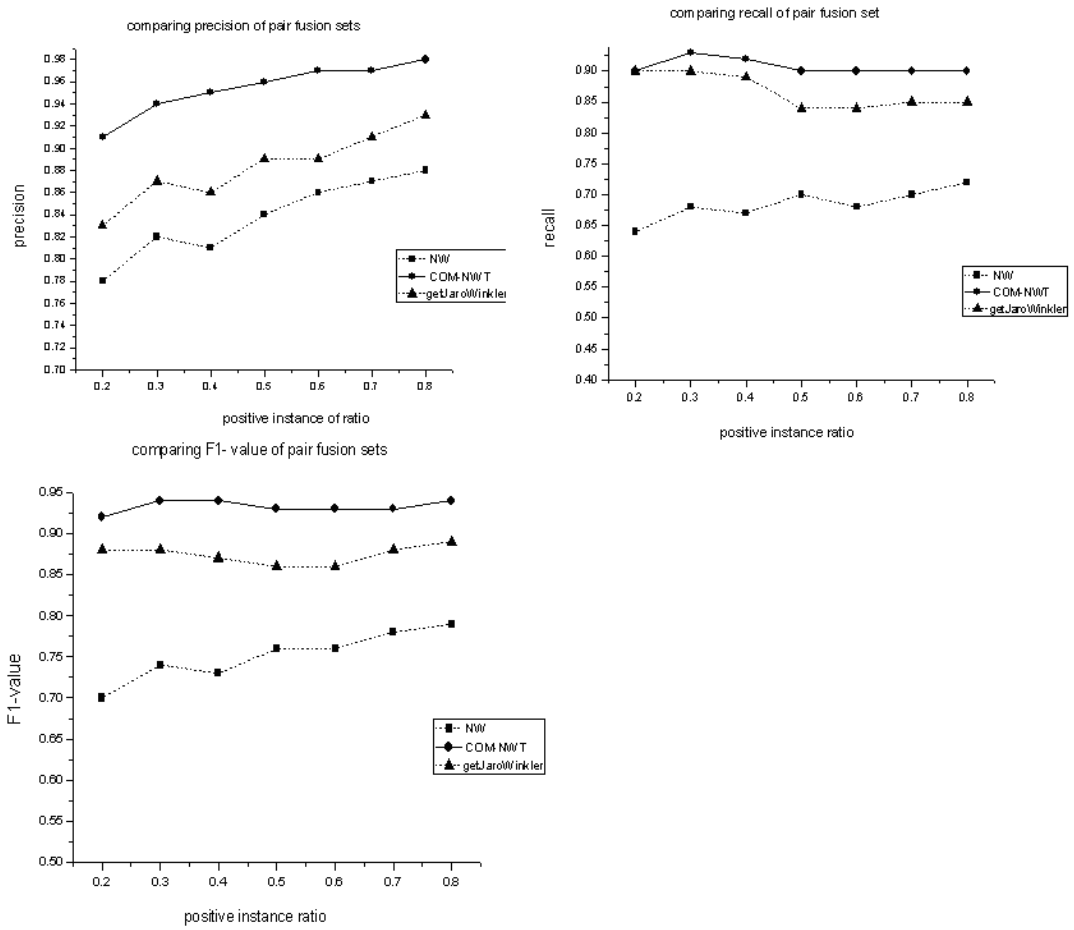


图 4-7　图中 NW 为仅用基于位置的标准化权重方法，TITLE 为仅用名称属性相似度方法，COM-NWT 为本文提出的改进后的方法，图中结果为各个方法对多数据集测试的准确率、召回率和 F1 值。

从上图中可以看出，改进后的方法的性能表现明显优于单独使用空间位置方法和非空间属性相似度方法。本文所使用的 POI 实验数据来自密度大范围小的繁华地图区域，加上地图间坐标误差比较大等因素导致了单独使用空间位置方法性能表现弱于单独使用非空间属性相似度方法。如果用现实中的 POI 数据，情况就会好些，但这不会影响本文提出的新方法的性能表现。另外，由于实验数据和现实中具体的数据特性有所不同，新方法中的某些参数可能要做相应调整以达到最好效果。

5 总结与展望

5.1 总结

本文对 POI 数据融合技术进行了深入的研究，最终实现了基于空间位置算法和非空间特征属性相似度算法，并在此基础上进了改进创新。本文的主要研究内容和创新之处如下:

① 对空间位置算法和非空间特征属性相似度算法原理进行了深入研究，并且编程实现了空间位置算法中的相互最近邻算法、概率算法以及标准化权重方法，非空间特征属性相似度算法中的文莱斯特距离算法以及哈罗-温克勒距离算法。

② 在实现了空间位置算法和非空间特征属性相似度算法的基础上，在空间位置算法中引入非空间属性信息相似度匹配算法，从而对现有的空间位置方法提出改进。改进后的方法在空间位置方法找到的融合集基础上使用名称属性相似度进行二次处理，这就实现了在空间位置的基础上结合其他非空间属性的多源 POI 数据融合设想。算法改进创新的基本思想是：如果来自不同的地理数据集合的 POI 有较近的位置同时也具有相似的非空间特征属性信息，那么它们成为正确融合对象的可能性就增大。

③ 选取测试本文提出的空间位置改进算法的实验数据集合，本文选定的实验数据是从谷歌地图与百度地图中利用两家地图提供的公共本地搜索 API 接口抽取的 POI 数据集合，POI 是地图中感兴趣点数据的简称，其英文全称是 Point of interest，POI 数据也是地图中比较简单的数据表示形式，可以理解为地理点状数据。在 POI 数据处理过程中，最重要的一个环节就是不同地图之间的坐标统一问题，这可以使用各个网络电子地图运行商提供的公共 API 中的接口转换函数来实现。

④ 最后，利用从谷歌地图以及百度地图中国抽取的 POI 实验数据集合组织七组不同差异程度即不同重合度的测试数据集合，然后用这些测试数据集对空间位置算法中的标准化权重算法、非空间特征属性相似度算法中的文莱斯特距离算法以及本文提出的基于空间位置的改进算法作对比测试实验，之所以采用空间位置算法中的标准化权重算法与非空间特征属性相似度算法中的文莱斯特距离算法，是因为进过测试发现这两种方法在各自算法领

域中表现最佳。最终结果表明本文提出的空间位置的改进算法表现明显优于单独使用空间位置算法和非空间属性相似度算法。

6.2 展望

本文只是采用了除空间地理数据中位置数据之外的名称特征属性，而一般的空间地理数据还应该包括其他非空间特征数据信息，就本文采用的 POI 实验数据而言还包含诸如地址、邮编以及电话号码等特征属性信息，所有这些特征属性都可以参与到非空间数据的处理中，如果使用多个非空间特征属性信息参与到算法运算中，那么可以预见其结果融合集质量将会有更大的提高。

此外，本文还存在不足之处，因为空间位置算法和非空间特征属性相似度算法的最佳算法和最佳阈值参数的选取以及空间位置算法中的影响因子的确定都是比繁琐而较复杂的过程，如何简化这些操作也是以后要解决的问题。

参考文献

[1] M. A. Cobb, M. J. Chung, H. Foley, F. E. Petry, K. B. Shaw, H. V. Miller. A rule-based approach for the conflation of attributed vector data. GeoInformatica, 1998, 2(1): 7-35
[2] Y. Doytsher, S. Filin. The detection of corresponding objects in a linear-based map conflation. Surveying and Land Information Systems, 2000, 60(2): 117-128
[3] Y. Doytsher, S. Filin, E. Ezra. Transformation of datasets in a linear-based map conflation framework. Surveying and Land Information Systems, 2001, 61(3): 159-169
[4] B. Rosen, A. Saalfeld. Match criteria for automatic alignment. In Proceedings of 7th International Symposium on Computer-Assisted Cartography (Auto-Carto 7), 1985: 1-20
[5] A. Saalfeld. Conflation-automated map compilation. International Journal of Geographical Information Systems, 1988, 2(3): 217-228
[6] O. Boucelma, M. Essid, Z. Lacroix. A WFS-based mediation system for GIS interoperability. In Proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems, 2002: 23-28
[7] G. Wiederhold. Mediation to deal with heterogeneous data sources. In Interoperating Geographic Information Systems, 1999: 1-16
[8] B. Amann, C. Beeri, I. Fundulaki, M. Scholl. Ontology-based integration of XML Web resources. In Proceedings of the First International Semantic Web Conference, 2002, Sardinia(Italy): 117-131
[9] Y. Papakonstantinou, S. Abiteboul, H. Garcia-Molina. Object fusion in mediator systems. In Proceedings of the 22nd International Conference on Very Large Databases, 1996: 413-424
[10] 彭煜玮, 彭智勇. 空间数据融合技术的研究. 计算机工程, 2007, 33(18): 51-55
[11] F. T. Fonseca, M. J. Egenhofer, P. Agouris. Using Ontologies for Integrated Geographic Information Systems. Transaction on GIS, 2002, 6(3): 231-257
[12] F. T. Fonseca, M. J. Egenhofer. Ontology-driven geographic information systems. In Proceedings of the 7th ACM International Symposium on Advances in Geographic Information Systems, 1999, Kansas(Missouri, US): 14-19
[13] H. Uitermark, P. V. Oosterom, N. Mars, M. Molenaar. Ontology-based geographic data set integration. In Proceedings of workshop on Spatio-Temporal Database Management, 1999, Edinburgh(Scotland): 60-79
[14] C. Beer, Y. Kanza, E. Safra, Y. Sagiv. Object fusion in geographic information systems[C]. Proceedings of the 30th VLDB Conference, 2004: 816-827
[15] C. Beer, Y. Doytsher, Y. Kanaza, E. Safra, Y. Sagiv. Finding corresponding objects when integrating several geo-spatial datasets. In ACM-GIS, 2005: 87-96
[16] I. Bhattacharya, L.Getoor. Iterative record linkage for cleaning and integration. Data Mining And Knowledge Discovery: Proceedings of the 9 th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery, 2004: 11-18
[17] W. E. Winkler. Methods for record linkage and Bayesian networks. Technical report, Statistical Research Division, US Census Bureau, Washington, DC, 2002:
[18] Parag, P. Domingos. Multi-relational record linkage. In AOM SIGKDD Workshop on Multi-Relational Data Mining, 2004.
[19] P. M. X. Li, D. Roth. Semantic integration in text: From ambiguous names to identifiable entities. AI Magazine Special Issue on Semantic Integration, 2005:
[20] A. McCallum, B. Wellner. Conditional models of identity uncertainty with application to noun conference. In neural Information Processing Systems Conference, 2004.
[21] A. Samal, S. Seth, K. Cueto. A feature Based Approach to Conflation of Geospatial Sources . International Journal of Geographical Information Systems

Science, 2004, 18(5): 459-489
[22] V. Sehgal, L. Getoor, P. D. Viechniki. Entity resolution in geospatial data integration. Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems ACM, 2006, (83-90):
[23] E. Safra, Y. Kanaza, Y. Sagiv, Y. Doytsher. Integrating data from maps on the world-wide web. Web and Wireless Geographical Information Systems, 2006: 180-191.
[24] N. Guarino. Formal Ontology and Information Systems. Amsterdam, Netherlands: IOS Press, 1998:
[25] 江宽, 龚晓鹏. Google API 开发：Google Maps 与 Google Earth 双剑合璧. 北京电子工业出版社, 2008:
[26] 孙晓茹, 赵军. Google Map API 在 WEB GIS 中的应用. 微计算机信息, 2006, 22(1):
[27] https://developers.google.com/maps/documentation/javascript/v2/reference.
[28] 杜传明. 百度地图 API 在小型地理信息系统中的应用. 测绘与地理空间信息, 2011, 34(2): 152-153
[29] 赵湘, 陈罡. 地理信息系统中的坐标变换. 第 11 届全国计算机在现代科学技术领域应用学术会议论文集, 2003:
[30] 杨元喜, 徐天河. 不同坐标系综合变换法. 武汉大学学报 (信息科学版), 2001, 26(6): 509-513
[31] 徐绍铨等. GPS 测量原理及应用(3S 丛书). 武汉测绘科技大学出版社, 1998:
[32] S. Nestorov, J. Ullman, J. Wiener. Representative objects: Concise represention of semistructured, hierarchical data. In Proceedings of ICDEBirminggham, U K, 1997:
[33] D. Suciu. An Overview of Semistructured Data. SIGACT News, 1998:
[34] J. Mchugh, S. Abiteboul, R. Goldman. A Database Management System for Semistructured Data. ACM SIGMOD Record, 1997:
[35] 鲁明羽, 陆玉昌. 基于 OEM 模型的半结构化数据的模式抽取. 清华大学学报(自然科学版), 2004:
[36] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. Soviet Physics-Doklady, 1966, 10(6): 707-710
[37] W. E. Winkler. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. Proceedings of the Section on Survey Research Methods, American Statistical Association, 1990: 354-359
[38] M. A. Jaro. Advances in record linkage methodology as applied to the 1985 census of Tampa Florida. Journal of the American Statistical Society, 1989, 84(406): 414-1989
[39] M. A. Jaro. Probabilistic linkage of large public health data file. Statistics in Medicine, 1995, 14(5-7): 491-498
[40] W. E. Winkler. Overview of Record Linkage and Current Research Directions. US Bureau of the Census, Statistical Research Division Report, 2006, 7(2): 196-210

致谢

时光荏苒，岁月如梭，一转眼三年的硕士研究生生活即将结束，回想这三年的点点滴滴，心中有万分不舍。在海大的这三年中，回想我从一个对学术一无所知的本科毕业生到一名对学术研究有所认知的硕士研究生，心中感慨良多。在这充实的三年硕士研究生生涯中，我不仅学到了扎实的专业理论，也感受到了导师治学严谨以及对学生负责的态度，也感受到了实实验室浓浓的学术氛围以及同窗之间的互帮互组、团结友爱的情谊。在这即将来临的毕业之际，我衷心的感谢每一位曾经给予我指导、关心、支持和帮助的老师、同学和朋友，谢谢你们！

首先，我要衷心感谢敬爱的导师张巍副教授，感谢他精心的指导和关怀，让我在学术研究中少走了很多弯路。感谢他对我的教导和帮助，为我在学术的道路上指明方向，教会了我进行科研的方法，同时他那严谨的学术态度也深深感染了我。张巍副教授孜孜以求的治学精神和学术上敏锐的洞察力，对我的科研工作给予了巨大指导和帮助；同时，感谢张巍副教授对我生活上的关心，在我困难的时候，总是给我勇气和鼓励；使得我的科研能够顺利完成。

感谢同一个科研小组的周广超、李瑞姗和王婷婷同学，感谢她们为解决科研中遇到的问题所付出的努力。

感谢我身边的同学和朋友，感谢你们对我各方面的帮助。

感谢我的家人，感谢他们对我的养育之恩，感谢他们为我付出的一切。

最后，对所有帮助我的老师、同学、朋友和家人表示崇高的敬意和由衷的感谢！

个人简历

1987 年 10 月 22 日出生于四川省广元市苍溪县。

2006 年 9 月考入四川师范大学地理与资源科学学院地理信息系统专业，2010 年 7 月本科毕业并获得理学学士学位。

2010 年 9 月考入中国海洋大学信息科学与工程学院地图学与地理信息系统专业攻读理学硕士学位。

发表的论文

[1] 张巍,高新院,李瑞姗. 空间位置信息的多源 POI 数据融合[J]. 中国海洋大学学报自然科学版,2013,43（10）.(已被中国海洋大学学报.自然科学版录用,核心期刊)

[2] 张巍,李瑞姗,高新院. 基于相似度模型的可融合兴趣点分类研究[J]. 中国海洋大学学报自然科学版,2013,43（12）.(已被中国海洋大学学报.自然科学版录用,核心期刊)

[3] 张巍,周广超,高新院. 全球海温变化的方差及其相关性分析. 中国海洋大学学报. 2012,42（1-2）：17~22

研究项目

[1] 国家自然科学基金项目（No.60602017,2007.12-2009.12）

[2] 山东省自然科学基金项目（No.ZR2012FM016,2010.01-2012.12）

中国海洋大学 OCEAN UNIVERSITY OF CHINA 硕士学位论文