中文图书分类号：TP391
密　　级：公开
UDC：004
学　校　代　码：10005

硕 士 学 位 论 文

MASTERAL DISSERTATION

论　文　题　目：基于语义和空间分布的多样性和比例性检索方法研究

论　文　作　者：兰许
学　　科：计算机科学与技术
指　导　教　师：才智

论文提交日期：2018 年 6 月

UDC：004
中文图书分类号：TP391
学校代码：10005
学　　号：S201507053
密　　级：公开

北京工业大学工学硕士学位论文

题　　目： 基于语义和空间分布的多样性和比例性检索方法研究

英文题目： RESEARCH ON RETRIEVAL METHOD BASED ON DIVERSITY AND PROPORTIONALITY BASED ON SEMANTIC AND SPATIAL DISTRIBUTION

论 文 作 者： 兰许
学 科 专 业： 计算机科学与技术
研 究 方 向： 内容检索
申 请 学 位： 工学硕士
指 导 教 师： 才智
所 在 单 位： 信息学部
答 辩 日 期： 2018 年 6 月
授予学位单位： 北京工业大学

独 创 性 声 明

本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，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，也不包含为获得北京工业大学或其它教育机构的学位或证书而使用过的材料。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了谢意。

签　名：　　兰许

日　期：　2018 年 6 月 7 日

关于论文使用授权的说明

本人完全了解北京工业大学有关保留、使用学位论文的规定，即：学校有权保留送交论文的复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存论文。

(保密的论文在解密后应遵守此规定)

签　　名：兰许　　日　　期：　2018 年 6 月 7 日

导师签名：才智　　日　　期：　2018 年 6 月 7 日

摘 要

空间信息检索技术的出现给我们的生活带来很大得方便，足不出户，便知天下位置，该技术主要是建立在具有空间位置和语义属性的空间数据集基础上，在用户给定位置的前提下，如何返回给令用户满意的l（自然数）个词条信息（元组）。由于用户在检索时大多数情况下，意图是不明确的，这给检索技术带来很大的挑战。故本文在空间信息检索技术方面的主要研究如下：

（1）提出一种新的离线排序策略—ValueRank。此方法主要用来计算数据集中各个节点的初始权值，这避免了目前大多数技术只是根据用户评分排序来选取结果的单一性。ValueRank是ObjectRank的扩展，其引入了动态“数值”的概念，即在计算某些属性节点的VR值时动态考虑的不仅是数量关系和给定的静态数值流动率（数值流动率为关系数据模式图中节点之间的相互贡献程度，详见第一章），还考虑其数值，形成动态数值流动率，即对于Northwind数据集来说，对于一个消费者的评价不仅仅在于其订单的数量，而主要是根据所有订单的总数值来计算其权值。用ValueRank计算出的初始权值不仅避免了仅仅将用户评分作为初始权值的单一性，还有一个可靠的数据理论支撑，为之后的检索做准备。

（2）提出基于语义多样性和等比例特性的检索方法。由于现存的检索技术都是在按权值大小排序的词条集合中，取前k个作为结果返回，这样可能会造成结果在某一类语义上聚集，此时，在并不了解用户意图的时候，返回的结果很难满足用户的需求，故提出基于语义多样性和等比例特性的计算方法。此方法是指用户在给定位置（在空间数据集中）或关键词（在纯文本数据集中）的前提下，能够在语义方面尽可能多样化地返回l（自然数）条结果。基于语义多样性是考虑当某类（语义相似即为一类，详见第四章）词条在结果集中出现多次，那么当下次准备从备选集中选择此类词条时，需要动态减小削弱其权值的系数（使其权值更小），以此来达到基于语义多样性的需求；基于语义等比例特性是考虑某类词条出现频率较高，但其权值较低，那么这也能说明此类词条和检索的关系词或是位置有某些联系，故当下次从备选集中选择此类词条时，本文将动态增大权值的系数，以此来达到基于语义等比例特性的需求。

（3）提出基于空间分布多样性和等比例特性的检索方法。此方法主要针对于空间数据集，空间检索大多都是按距离远近进行排序，优先返回l（自然数）个离检索点最近的点组成结果集。由此可见无论是按此方式返回还是按权值大小

返回也都可能造成结果在某一空间分布上聚集，故提出基于空间分布多样性和等比例特性方法，在多样性上根据欧式距离公式的特性来选择备选节点，而在等比例特性方面将空间分布以检索点为中心分为四个方向，沿用基于语义等比例特性的方法生成结果。

最后将基于语义多样性和空间分布多样性结合生成Dsize-*l* OS，将基于语义等比例特性和空间分布等比例特性结合生成Psize-*l* OS。实验结果证明本文提出的检索方法有效。

关键词：信息检索；空间分布；语义关系；静态离线排序策略

Abstract

The emergence of spatial information retrieval technology has brought great convenience to our lives, likes knowing the location information of the world with no leaving the home. The technology is mainly based on spatial data sets with spatial position and semantic attributes. Under the premise of location given by users, how to get the *l* (nature number) information (tuples) that makes the user satisfied. Due to the fact that the user's intention is not clear in most cases when searching, this brings a great challenge to the retrieval technology. Therefore, the main research of this paper in spatial information retrieval technology is as follows:

(1) A new offline sorting strategy named ValueRank is proposed. This method is mainly used to calculate the initial weight of each node in the dataset, which avoids the singleness of most of the current technologies that simply select the results based on user ratings. ValueRank is an extension of ObjectRank, which considers the concept of dynamic “value”, that is, when calculating the VR values of certain attribute nodes, it not only considers the quantitative relationship and the given static value flow rate (Value flow rate is the degree of mutual contribution between nodes in relational data schema diagrams. See Chapter 1 for details.), but also considers its value and forms a dynamic value flow rate. For the Northwind dataset, the evaluation for a consumer is not just the number of orders, but mainly based on the total value of all orders to calculate its weight. Calculating the initial weights using ValueRank not only avoids the use of user scores as a single unit of initial weights, but also provides a reliable data theory support to prepare for subsequent searches.

(2) Proposes the method for calculating the semantic diversity and equal proportions of search results. Since the existing retrieval techniques are based on tuples’ weight, the top *k* is returned as the result. The drawback is that it may cause the result to be aggregated in a certain type of semantics. At this time, when the user's intention is unknown, the result is difficult to meet the needs of users. Therefore, a method for calculating the semantic diversity and proportional of results is proposed. This method means that the user can get *l* (natural number) tuples as semantically as possible in a given location (in spatial dataset) or in a keyword (in plain text dataset). The result of semantic diversity is to consider that

when a tuple of a certain type (tuples have semantic similarity will be identified as the same type, see Chapter 4 for details) appears multiple times in the result set, then the coefficient of its weight (even if its weight is smaller) will be dynamically reduced in the next selection of tuples in the same type from the candidate set, in order to achieve the semantic diversity of the results; the results of the semantic proportional of characteristics when considering the tuples of a certain type with higher frequency but lower weight, then it can also indicate that there are some certain links between such tuples of this type and keywords or positions of the search. Therefore, this paper will dynamically increase the coefficients of the weights of such tuples from the candidate set, in order to achieve the semantic proportional of the results.

(3) Proposes the method for calculating the spatial distribution diversity and equal proportions of search results. This method is mainly aimed at the spatial dataset. Spatial retrieval is mostly sorted according to the distance, and gets the first l points closest to the retrieval point to form the result set. It can be seen that whether getting results in this way or by the weight may cause the results to be aggregated in a certain spatial distribution. Therefore, the method for calculating the spatial distribution diversity and equal proportions of search results is proposed, In terms of diversity, the candidate tuple is selected according to the characteristics of the Euclidean distance formula. In the aspect of proportional, the spatial distribution is divided into four directions with the retrieval point as the center, and results are generated using the method of equal proportions of the result semantics.

Finally, Dsize-l OS is composed by l tuples considering combining the diversity of semantic and spatial distribution, Psize-l OS is composed by l tuples considering combining the proportional of semantic and spatial distribution. Experimental results prove that the retrieval method proposed in this paper is effective.

Keywords: Static offline sorting strategy, semantic relations, spatial distribution, information retrieval

目录

第 1 章　绪论

1.1　研究背景及意义

21 世纪以来，随着互联网中信息内容的丰富性和易访问性，使其快速地融入到人们的日常生活中。互联网的功能从最初的仅仅被动发布数据，然后是交互地获取所需数据，发展到现在的根据用户提出的需求来获得信息，并能进行智能检索。随着信息资源越来越丰富、信息量越来越大并且仍将持续地爆炸式的增长，此时，信息检索成为信息社会中不可或缺的一种工具手段。在数据表达方面，随着信息量的增长，关系数据作为一种新型的数据存储方式使得检索的信息更加准确，基于关键词的关系数据集检索技术[1~3]的兴起给用户带来了巨大的方便。

随着无线网络技术和移动便携终端的发展，基于位置的服务（LBS）[4]兴起，空间检索技术给人们的生活带来了巨大的方便。用户通过授权移动终端等设备将其当前所在位置信息发送给位置服务商（LSP, location based service provider），以获取附近查询服务，如查询周围的餐厅、停车场等。后来，将空间与关键词检索结合而来的空间关键词查询（SKQ, spatial keyword query）[5~9]应运而生，一个词条也称作是兴趣点（POI, point of interest），除了具有文本描述属性之外，还加入了位置坐标属性。SKQ 即在具有位置坐标与文本双重属性的大量 POI（Point of Interest）数据基础上，给定一个位置和若干个关键词作为参数，返回满足空间与文本约束的结果，这些结果词条往往根据基于权重排列的，而词条权值过去又往往是基于词频和反文档词频的组合（简称 TF.IDF）模式加权。但是随着近几年数据量的大幅度增加，在大数据规模下，语义网络的出现，使得加权模型计算权值的代价不是线性的，而且包含符合约束的词条结果可能是成千上万的，对于目前的检索技术所返回给用户的结果是否是精简地、符合用户的需求等这个都给空间语义检索技术带来了巨大的挑战。我们可以得到传统的语义空间信息检索技术存在以下几个问题：

1. 评分标准单一。

在空间文本检索中，词条大多是按用户打分来排序的，而在传统信息检索系统中，其结果都是按照其词条权重排列的，权重是由离线排序算法算得，在词条方面过去由 TF.IDF 模式加权，后来在对于关系型数据集方面运用了基于链接分析的离线算法 ObjectRank [47]（简写为 OR）迭代计算权值，虽然 OR 避免

了结果过于倾向于词频衡量重要性，但是还是过于依赖于数量，因此评价的标准还是较单一，并不能适用于所有的关系数据集。

2. 忽略结果语义多样化的需求。

如图 1-1，对于关键词“饭店”，百度地图搜索结果列出的 5 个文本词条，其中 4 个都是酒店，显然，对于一个想要找餐馆的用户来说可能并不能满足其需求，所以此处在“酒店”语义上过于密集，忽略了语义的多样性。不仅如此，线上应用大多数都是根据用指定的关键词来检索，而很少存在给定一个坐标位置，根据周围 POI 坐标点来解释此检索点，即用户给定一个位置进行模糊检索能够返回给用户的结果在语义方面是多样化的。

3. 忽略用户基于位置的空间分布多样化的需求。

在使用图 1-1 等精准搜索时，给定一个位置和多个关键词，返回给用户的是基于这个位置附近、根据相似度计算等方法计算与用户关键词相似度最高的多个词条结果，如图 1-1，“北工大”是给定的地理信息，“饭店”是给出的语义文本关键词，百度地图列出 10 个标注，都过于聚集在西北方，很明显，对于一个未能明确方位需求的用户，返回的结果可能并不符合用户的意图，也无法向一个对此陌生的用户更好地解释此位置。

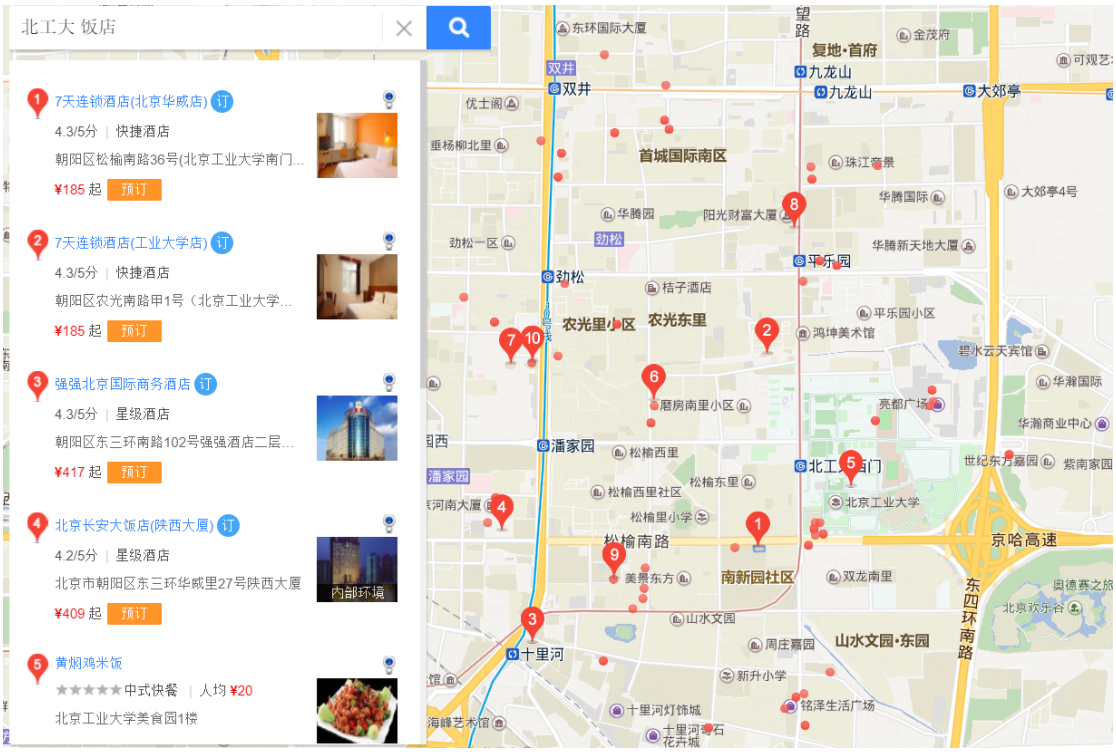


图 1-1 线上应用举例

Figure 1-1 The example of online application

目前，已经存在很多的空间关键词检索技术，大致可以分为以下三类：松散组合、空间优先和文本优先。但是大多数的结果都是按照权值从高到低排列的，这些结果词条可能在某种语义或空间上聚集严重，在用户在不熟悉的位置下，这些技术都无法在语义和空间上给用户返回多样信息。

为了提高语义检索质量，考虑到用户的多重需求，邬等人[10]、王等人[11]和刘等人[12]提出的针对于文本的多样性排序研究和研究现状，但是目前对空间和文本语义结合方面的多样性研究比较欠缺。因此，本文主要在离线计算方面，利用考虑节点数值的 ValueRank 离线权值计算方法计算词条权重；在检索方面，结合空间和文本的多样性和等比例特性来提高搜索结果的质量。在用户给定一个位置坐标前提下，返回以位置坐标为中心，语义上多类别、空间上多方位或符合语义和空间分布结果等比例的多个词条，以此来满足用户的多重需求。

1.2　研究现状

1.2.1　静态离线排序策略

离线计算是指在提前知晓所需的输入数据，且输入数据在之后的操作过程中不会改变的前提下，为了使后续操作快速得到结果而提前对已知数据进行的计算。离线排序策略，顾名思义对所有输入数据根据排序计算策略，通过离线计算而得到的初始权值，根据权值进行排序的策略，在传统信息检索领域，离线计算得到排好序的权值是为后续检索做准备的，也就是，用户给定一个关键词，根据约束和离线排序策略计算好的排序结果返回给用户 k（自然数）条信息。

Google 公司提出的 PageRank[13]算法是最为经典的离线排序策略，传统的网页文本检索是根据用户输入的关键词返回给用户的结果是跟关键词匹配程度较高的网页，所以多数网页都会增加其关键词的数量来达到提升排名的效果，PageRank 的提出主要是为了避免这种传统方法的弊端。此方法地提出主要是依据论文与论文之间的引用作为原型，将这种方法映射到网页与网页的链接关系，也就是说，（1）如果很多的网页都能链接到这个网页 P 的话，则说明 P 是比较重要的网页，所以其相应的权重（PageRank 值，简称 PR 值）也会较高，所以在满足关键词约束的情况下，会更优先地返回 P；（2）如果 P 的 PR 值（PR(P)）很高，它链接到其他的网页（例如 P_1,P_2），那么 P_1,P_2 所对应的 PR(P_1)和 PR(P_2)也会由于 PR(P)而响应地提高。可以看出这一方法地出现使得大家都感兴趣的

词条（文本或位置）取得更高的权值，避免了单纯的基于词频进行加权出现结果分布倾斜的弊端。

后来，A Balmin[14]等人将 PageRank 算法运用到关系型数据集中，像 DBLP 数据集，但是由于 PageRank 中各个链接边的顶点之间的数值流动都为 1，也就是说，如果单个节点的链入数量多，那么它的 PR 值很可能是比较高的，很显然，对于每个节点这并不合理，数量只能作为一方面的考虑因素。所以他们新提出了 PageRank 算法的扩展版—ObjectRank，将其运用到关系型数据集，如书目数据集 DBLP，并将各个顶点之间的链接边的数值流动进行削弱（≤1），不同类别顶点之间的数值流动不同，这样就避免了只考虑链接数量的关系。

然而，由于 ObjectRank 只将其运用到了 DBLP 数据集，而且对于各个顶点的 OR 值的影响，除了数量之外，仅仅依靠于他们之间的数值流动率，而这个数值流动率是固定的，对于一个像 Nothwind 这样的商业类关系型数据集，ObjectRank 算法的思想显然是单一的，例如，对于评价“供应商”属性节点来说，用 OR 或者 PR 只会考虑供应商所提供“商品”的数量，但是却忽略了其单价的问题。所以适用于多类型数据集的通用型离线排序策略有待进一步研究。

1.2.2 语义多样性和等比例特性

信息检索主要是为了满足用户的查询需求而返回用户感兴趣的有序结果集，结果集是由有限个词条组成。传统信息检索方法是匹配用户关键词，返回与关键词相关性较高的词条集合，但是用户在检索的时候都是带有某种意图的，如果意图可知，显然结果是可以通过某种方法进行估算，但大多数情况下这种意图是潜在的、未知的，信息检索系统不能仅仅通过几个关键词与后台已有数据的相似度来对信息进行排序，所以为了更好地满足用户的多样化需求，结果的多样性排序问题成为近几年的研究热点。

目前研究较多的是隐式多样性检索方法，隐式多样性是指用户的意图是不明确的，所以不根据用户的意图对结果进行建模计算排序，而是通过计算已选词条与候选词条的相似度，尽可能地选择与已选词条相似度低的词条加入结果集[15]。关于“多样性”最早可追溯到最大相关边界法[16]（Maximal Marginal Relevance，MMR）的提出，其主要的思想是迭代将候选词条集合加到结果集中，在选择词条的原则是在尽可能保证被选词条与查询关键词相关的前提下，选择候选集中与已选结果集内容相似度较低的词条加入结果集，此方法减少了结果集的数据冗余。MMR 方法的提出对研究搜索结果“多样性”有着重要的而深远的意义，后来出现的大多“多样性”计算方法都是基于 MMR。之后 S.

Gollapudi[17]提出一个生成“多样性”结果的通用框架，包含了八个定理。在[17,18]中，作者提出了 max-sum，max-min，mono-objective 目标函数和算法。[19,20]给出了 *sim* 和 *dis*(详见第二章)函数和目标函数的概率解释。“多样性”直观的概念是 *DisC* diversity[21,22]的提出，使得每一个子集中每一类元素都由一个相似的元素代表，不同类元素之间都不相似，用这种方法来保证了结果的多样性（即 *diversity*）。在[18]中，提出了 LogRank，一种基于权重流动的算法，通过离线算法计算出用户的行为的代表性元组集合，同时通过不同的时间和类别选择其中具有代表性的重要活动的元组，来达到多样性的目的。

关于等比例特性，首先文献[23]提出了基于选举的方法，为了生成按比例的多样化结果，但是，忽略了元素之间的相似性和评分函数，这样就可能导致返回与查询不相关的结果，后来文献[24]通过在目标函数中考虑到相关性来解决这一局限性。

由于本研究需要在保证查询结果候选集是都与检索点相关且在纯文本数据集中还要保证候选集都是与以关键词为根且与其相连的树形结构，所以现有的方法都不适用于。

1.2.3　空间关键词检索

空间关键词查询（SKQ）的出现是因为随着 LBS 的发展，许多数据不仅有空间位置信息，而且还加入了文本信息，所以 POI 是有着空间和文本多重属性的点，在文献[25]中称这种既有地理位置信息又有文本信息的对象为 spatio-textual object，其中一个对象 *o* 由其文本信息（关键字集合）*o.kws* 和其位置坐标信息 *o.loc* 两部分组成。

一个典型的 SKQ 问题 *q* 由以下几个部分组成：

1）给定查询的位置坐标信息 *q.loc*

2）给定查询的关键词组合 *q.kws*

3）约束集合 *q.C*

4）排序公式 *q.f*

5）返回结果的数量 *q.k*

当我们给定一个由多个 *o* 组成的集合 *R* 和一个 *q*，检索技术根据 *q.C* 从 R 中返回 *q.k* 个根据 *q.f* 排序的有序 *o* 组成的队列[26]。空间关键词检索技术大体框架如此，其中技术差别主要体现在 *q.C* 和 *q.f* 上。查询约束方面，分为空间约束和关键词文本约束，空间约束根据 *o.loc* 和 *q.loc* 的关系，分为相交、包含和被包含[25,27~29]等；而关键词文本约束根据 *o.kws* 与 *q.kws* 的关系，可分为完

全匹配、部分匹配、模糊匹配[30]等。排序公式方面，将空间相关性（距离）和关键词文本的相关性按一定比例结合，得到相应排列顺序，例如有的只考虑查询结果的距离[31~35]，有些是考虑空间和关键词文本相关性的线性组合[29,36~39]，还有的是考虑空间距离和关键词文本相关性的比值[40]。本文中在空间检索中，在不考虑关键词的情况下也能生成检索结果，同时还要考虑多样性和等比例特性，在空间上表现为方位，在关键词文本上表现为类别，例如用户搜索某一地点，我们需要返回的结果尽可能包含各个类别的坐标（饭店、KTV、景点等），当然也要考虑到各类别的比例。所以，现有的方法显然并不适用。

1.3 本文研究内容与组织结构

1.3.1 论文的主要研究内容

目前空间关键词检索主要面临三个问题，一是对于每个 POI 的评分标准比较单一，大多数是采取用户评价打分机制，而少有研究考虑到各个 POI 的分数质量问题。二是忽略用户所给关键词的语义多样性和等比例特性问题，目前有的空间关键词检索技术在语义方面大多都是按相似度排序的，而忽略了用户的隐性意图，所以检索结果语义单一。三是忽略用户所给位置的空间分布多样性和等比例特性的问题，在空间分布方面仅仅单纯考虑了距离问题，而没有考虑方位分布，这使得在不了解此位置的用户无法获得更多的信息。考虑到上述问题，本文做了如下研究。

第一，采用 ValueRank 离线排序策略得到 POI 的初始权值（gi，global importance）。本文采用了基于 PageRank 的扩展算法，对每对关系中点到点的数值流动中引入“Value”概念代替 PageRank 中的“1”，而且首次将 ValueRank 引入空间关系数据集的计算中。

第二，实现基于语义多样性和等比例特性的检索方法。对于纯文本数据集来说，本文采用结果树来保证了返回结果都与检索关键词具有一定的联系性，而对于空间数据集，则用指定的检索半径候选范围来保证结果与所检索的点在空间上存在一定的关系。最后在候选结果中根据多样性和等比例特性计算公式来保证结果语义上的多样性和等比例特性。

第三，实现基于空间分布多样性和等比例特性的检索方法。本文用户所检索的位置为地理坐标，故在以此位置为中心的候选范围内，得到候选节点集合，最后根据空间分布多样性和等比例特性计算公式得到结果。

第四，实现基于语义和空间分布结合的多样性和等比例特性检索方法。本文用户所检索的位置为地理坐标，故在以此位置为中心的候选范围内，得到候选节点集合，根据语义和空间分布结合多样性计算得到关于此集合基于语义和空间分布结合多样性的检索结果；同理，根据语义和空间分布结合等比例特性计算得到关于此集合基于语义和空间分布结合多样性的检索结果。

图 1-2 为本文主要研究生成 size-*l* OS（详见第二章）的流程图，首先根据 ValueRank 计算得到各节点权值，然后根据基于语义多样性或等比例特性，空间分布多样性或等比例特性对权值进行削弱得到新的权值，最后按照权值大小进行排序，选择权值较高的节点作为结果（size-*l* OS）输出。

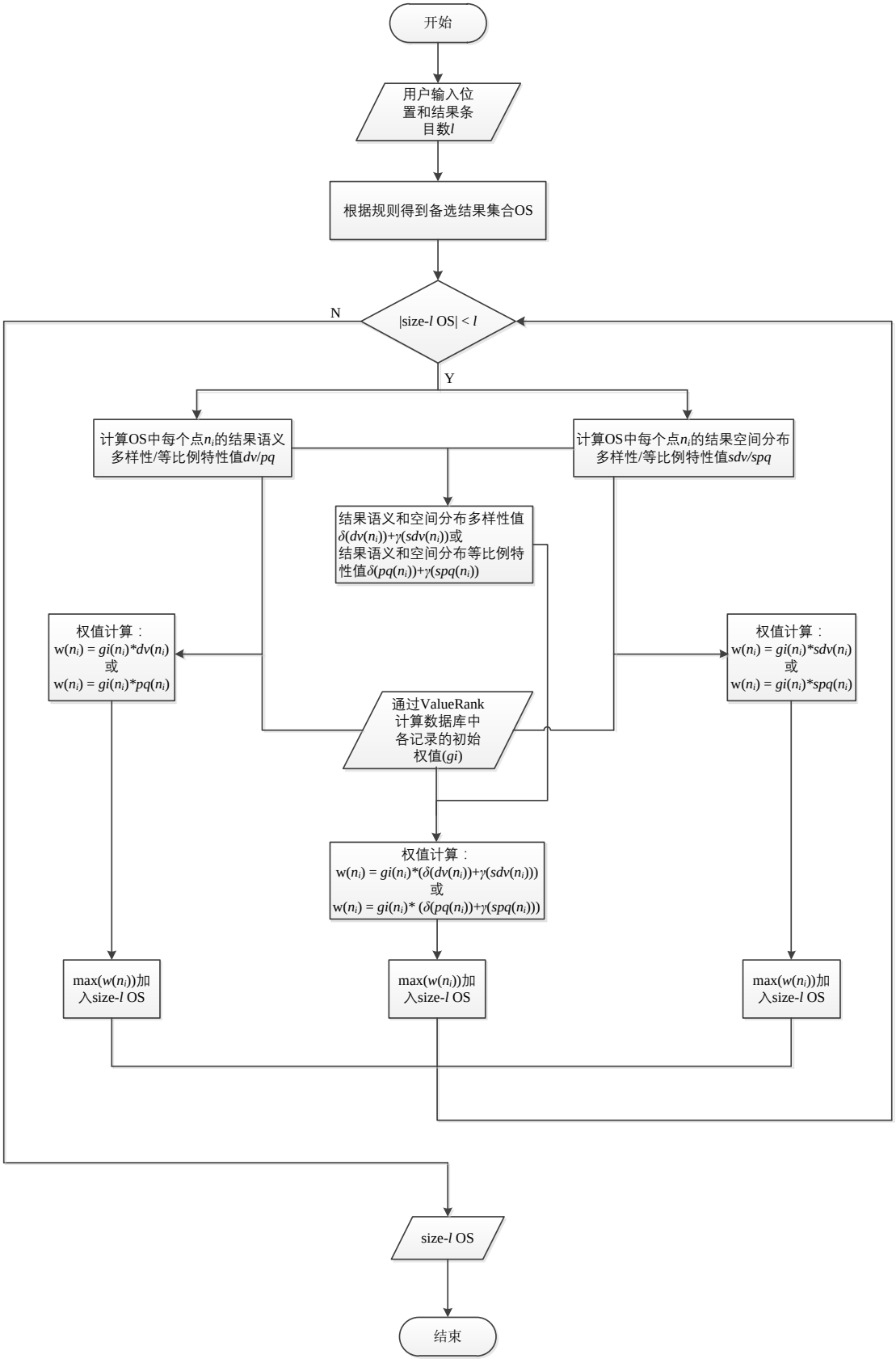


图 1-2 size-*l* OS 生成流程图

Figure 1-2 size-*l* OS generation flow chart

1.3.2　论文的组织结构

第 1 章：绪论。主要阐述了本课题的研究背景和意义，介绍了静态离线排序策略、基于语义多样性和等比例特性和空间关键词检索的研究现状，最后简要说明论文的研究内容和组织结构。

第 2 章：相关研究。主要介绍了本文所用到的理论基础，静态离线权重计算方法（包括 PageRank 和 ObjectRank 算法）和关键词检索所需的理论基础和相关研究现状，为后续提出的新方法做理论支撑。

第 3 章：考虑数值的离线权重计算方法—ValueRank。主要介绍 ValueRank 的理论基础，将其运用到 DBLP 和 NorthWind 数据集的实验结果及分析，以及运用到北京 POI 数据集的结构、计算过程和实验结果分析。

第 4 章：基于语义多样性和等比例特性检索方法。本章主要在第三章通过 ValueRank 计算得到的初始权值的基础上，提出语义多样性和等比例特性计算方法，根据语义多样性或是等比例特性计算公式对检索候选结果集的权值进行削弱，得到新的考虑语义多样性和等比例特性的综合权值，再根据这个权值排序生成最终的检索结果。对于此章的方法，主要在两类数据集中通过实验分析此方法，一类是纯文本的数据集（DBLP 和 Northwind），另一类是复合型数据集（也就是北京 POI 空间数据集），本文主要在纯文本数据集中验证本文所提出的检索方法而在复合型数据集中进行多次实验，得到量化结果。在检索方面两种数据集不同的是备选结果的集合文本数据集是根据结果树的形式选择的备选结果集，而空间数据是根据检索点指定范围内的所有点组成的备选结果集。最后，根据此方法得到检索结果并分析比较。

第 5 章：基于空间分布的多样性和等比例特性检索方法。本章主要介绍了对于复合型数据集，如何考虑空间分布多样性和等比例特性的情况下得到检索结果。提出空间分布多样性和等比例特性的计算方法，主要是从“方位”方面对所有候选节点进行权值的削弱，最后根据此方法得到检索结果并分析比较。

第 6 章：基于语义和空间分布多样性和等比例特性的检索方法。本章将第四章和第五章的方法结合，引入参数调节参数，一是将语义和空间分布多样性结合，得到基于语义和空间分布多样性的检索结果，二是将语义和空间分布等比例特性结合，得到基于语义和空间分布多样性的检索结果。

第 2 章　相关研究

2.1　静态离线权重计算方法

PageRank[41]由谷歌提出的有效的网页权重排名算法，此方法地提出主要是依据论文与论文之间的引用作为原型，将这种方法映射到网页与网页的链接关系，后来将此方法运用到关系型数据集中，如 DBLP 数据集，将已经建模好的标记数据图定义为 $D(V_D,E_D)$，根据关系得到其数据模式图 $G(V_G,E_G)$，图 2-1 是 DBLP 数据集的数据模式图，其中 V_G（$v_1,...,v_n$）是元组集，这里的元组代表各类表中每条记录，E_G 是代表边（弧）的集合，E_G＝{<v_i, v_j>| v_i, v_j∈V }，<v_i, v_j>表示从 v_i 到 v_j 的一条边（弧），即 v_i 的信息能够链接到 v_j，令 r 为一个矢量(各个节点 PageRank 值组成的向量)，其中每个元组 v_i 都存在相应的 PR 值 r_i，则通过以下公式来迭代计算矢量 r 为：

$$r = dAr + (1-d)\frac{e}{|V|} \qquad (2\text{-}1)$$

其中 d 是一个（0,1）的阻尼系数，此系数能够保证得到更精确的结果，一般取值为 0.85。A 是一个 $n * n$ 矩阵，n 代表顶点个数，其中若存在从 v_i 到 v_j 的边（弧），则 $A_{ij}=\frac{1}{O(v_j)}$（$O(v_j)$表示 v_j 的出度）；否则为 0。例如，若有三个元组，则 A 是一个 $3*3$ 矩阵，v_0 到 v_1 和 v_2 都有边且 v_1 到 v_2 有边，则 $A_{10}=A_{20}=\frac{1}{2}$ 且 $A_{21}=1$，其余都为 0；e=[1....1]^T；$|V|$为元组总个数。

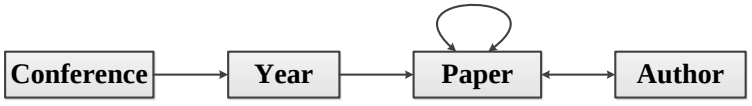


图 2-1　DBLP 数据集模式图

Figure 2-1 The DBLP Dataset Schema Graph

ObjectRank [47]是 PageRank 的扩展，其主要思想是在关系数据集的关系中引入数值流动率的概念（带有数值率的模式图称为数值流动模式图，简称 G^A）。PageRank 中仅将所有的关系数据集映射到一个图是不准确的，因为 PageRank 将每个关系之间的数值流动率都设定为“1”，即关系之间的贡献程度为 1，所以在很大程度上还是由数量决定元组的权重，这时需要一个数值流动模式图来控制各

表中元组之间的数值流动。举例来说，被引用的论文要比引用论文更重要，此时就需要调高被引用论文边的数值流动率而调低引用论文边的数值流动率。

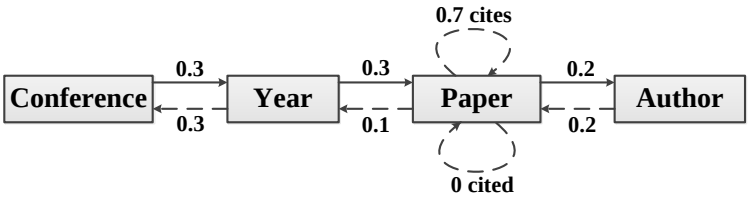


图 2-2 DBLP 数据集数值流动模式图（ObjectRank）

Figure 2-2 The DBLP Authority Transfer Schema Graph G^A (ObjectRank)

具体来说，根据 DBLP 数据集的模式图 $G(V_G,E_G)$，可以得到其相应的数值流动模式图 $G^A(V_G, E^A)$（如图 2-2 所示），图中边上的权重反映相邻表中元组之间的数值流动率，定义 E_G 的每条边为 $e_G = (v_i \rightarrow v_j)$，随后可以得到相应的两条数值流动边，定义其前向数值流动边为 $e_G^f = (v_i \rightarrow v_j)$，后向数值流动边为 $e_G^b = (v_j \rightarrow v_i)$。最终，根据数据图 D 和数值流动模式图 G^A，可以得到数值流动数据图 $D^A(V_D,E_D^A)$，其中，对于 E_D 集合中的每条边，D^A 中有与其对应的两条边分别为 $e^f = (v_i \rightarrow v_j)$ 和 $e^b = (v_j \rightarrow v_i)$，因此得到两条边对应的数值流动率 $\alpha(e^f)$和 $\alpha(e^b)$，其中如果 $OutDeg(u, e_G^f) > 0$ ($OutDeg(u, e_G^f)$ 是节点 u 的 e_G^f 出度)，$\alpha(e^f) = \alpha(e_G^f)/OutDeg(u,e_G^f)$；否则 $\alpha(e^f) = 0$。$\alpha(e^b)$定义相同。

向量 r 由每个节点 v_i 的 ObjectRank 值 r_i 组成，r 由以下公式得出：

$$r = dAr + (1-d)\frac{s}{|S|} \qquad (2\text{-}2)$$

这里用所有元组集 V_D 的一个任意的子集 S 来代替 V_D 作为基本集，因此可以增加与它们相关联元组的数值流动，在计算 ObjectRank 时，S 可以是包含关键词元组的集合。其中，如果在 E_D^A 中存在一条边为 $e = (v_i \rightarrow v_j)$，那么 $A_{ij} = \alpha(e)$；否则为 0。d 是一个（0,1）的阻尼系数，此系数能够保证得到更精确的结果，$s = [s_1, \ldots, s_n]^T$ 是 S 的基本集向量，如果 v_i 属于 S 则 $s_i = 1$，否则 $s_i = 0$。基本集 s 的概念是在[4]中提出的，作为一种进行个性化排名的方法，将 s 设置为用户的书签集。

基本集 S 的概念首次在[41]中提出，文中将 S 设置为用户的书签集，运用到一种进行个性化排名的方法。在[42]，它被用来计算网页中的 PageRank。它还被用在计算 ObjectRank2 中[43]。

2.2　关系数据集中的关键词检索

此节主要介绍关系型数据集中关键词检索所用到的相关理论知识，主要包括 *Object Summary* 和 *size-l OS* 相关概念介绍。

2.2.1　*Object Summary*

在关键词检索中，将包含给定的关键词的元组表定义为数据主体（Data Subject ，简写 DS），当给定关键词后，根据关系数据集的关系特性（如 DS 为 Author 表中元组），可以生成以 DS 为根（R^{DS}），以能与 R^{DS} 有链接关系的表为子孙的树，R^{DS} 的数据主体模式图（G^{DS}）由此建立，它是为了寻找数据模式子集而建立的一颗标记树（如图 2-3 所示）。*Object Summary* ，简称 OS，是根据 G^{DS} 而建立的实例化树形结构，具体来说根据给定的关键词，如 DBLP 数据集中“M. Faloutsos”，可以得到以 Author 表为根节点，Paper、Paper (引用)、Paper (被引用)、Co-author 等为子孙节点的 G^{DS}，然后考虑到关键词“M. Faloutsos”，生成以含有“M. Faloutsos”的元组（$t^{DS} \in R^{DS}$）为根节点，它的相关关系 Paper、Co-author 等表元组递归组成子孙节点的树的集合，即为 OSs。在生成 OS 的过程中为了得到更好的 OS 结果，提出区分 OS 中的每个不同表中元组 t_i 的重要程度，引入了亲和度（*Affinity*，缩写为 *Af*）的概念。文献[44]中讲到如何从 G^{DS} 中遍历计算与 R^{DS} 有的亲和度，深度越浅也就是与 R^{DS} 离得越近的元组具有较高的亲和度。R_i 到 R^{DS} 关系的亲和度由以下公式得出：

$$Af(R_i) = \sum_j m_j \cdot w_j \cdot Af(R_{parent}) \qquad (2\text{-}3)$$

其中 $m_j \in \{m_1, m_2, ..., m_n\}$是某种因素，$w_j \in \{w_1, w_2, ..., w_n\}$是与其对应的权重，$Af(R_{Parent})$ (≤ 1)是 R_i 的父节点到 R^{DS} 的亲和度。R_i 到 R^{DS} 的亲和度因素包括（1）他们之间的距离，（2）他们在数据模式和数据图中的链接属性（具体看[45]）。因此可以得节点的亲和度与其父节点的亲和度单调不增，给定亲和度阈值 θ，认为小于 θ 的元组与 R^{DS} 亲和度较低，所以不会考虑将其作为检索结果输出。给定 θ 后可以得到 G^{DS} 的子集为 $G^{DS}(\theta)$。最终我们可以递归 $G^{DS}(\theta)$来构建 OSs，例如，给定关键词“Christos Faloutsos”（一个作者，将问题简写为 Q1），图四就是根据 Q1 构建的 θ=0.7 的 G^{DS}，可以看到 Author $G^{DS}(0.7)$包含了所有的关系。

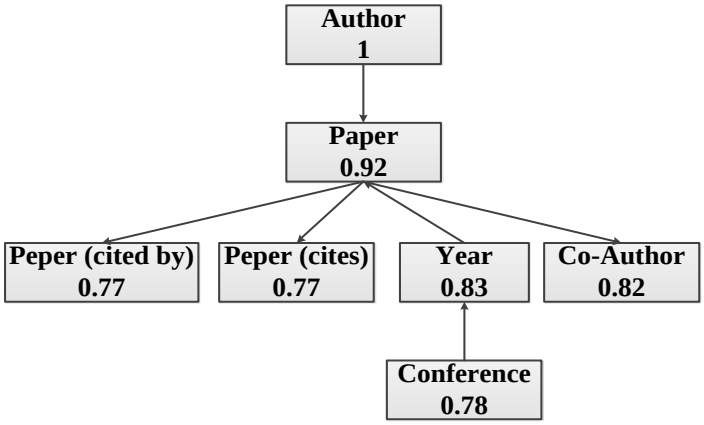


图 2-3 DBLP 数据集 Author 数据主体模式图(带亲和度)

Figure 2-3 The DBLP Author G^{DS} (Annotated with Affinity)

各元组通过 PageRank 等离线权重计算方法计算得到的权重为全局权重（*global importance*，*gi*），OS 中任意节点 t_i 的局部权重（*local importance*，缩写为 *li*）是由这个元组起始的全局权值（*gi*）和这个元组在 OS 中的与 R^{DS} 的亲和度两部分所决定的，所以 t_i 在 OS 中的 *li* 定义为 Im(OS, t_i)，其由以下公式得出：

$$\mathrm{Im}(OS, t_i) = li(t_i) = gi(t_i) \cdot Af(t_i) \qquad (2\text{-}4)$$

其中，$gi(t_i)$是 t_i 的全局权重。所以构建 OS 的步骤如下：

（1）通过离线权重计算方法（PageRank、ObjectRank 等）计算元组的全局权重；

（2）根据公式 2-3 计算出亲和度亦或是人工设定亲和度；

（3）根据公式 2-4 计算出局部权重。

例如，根据图 2-3 和 Q1，得到一个 Paper 表中的一个元组 t_i 为“Efficient and Effective Querying by image Content”，$gi(t_i)$=21.74，$Af(t_i)$ = 0.92，所以 Im(OS, t_i) = 21.74 ×0.92=20。亲和度对于关键词检索技术来说是必要的，因为它有利于得到更准确的检索结果集，例如对于两个元组，Paper 表中的元组“Efficient...”和 Year 表中的元组“1998”，它们的全局权重分别为 21.74 和 21.64，根据公式 4 得到它们在 OS 中的局部权重为 20 (=21.74 ×0.92) 和 18 (=21.64 ×0.83)，Paper 元组的权重大于 Year 元组的权重，所以我们认为 Paper 表中的这个元组对结果来说更有意义，当然，由于 θ 的关系，与根节点亲和度低的元组不会出现在 OS 中。

2.2.2　size-*l* OS

size-*l* OS 是一个由 *l* 个元组（节点）组成的集合，给定一个 OS 和一个整数 *l*，一个候选 size-*l* OS 为 OS 中 *l* 个元组的组成的子集。size-*l* OS 的结果符合以下两个准则：

（1）在 OS 树中，所有的 *l* 个节点都与 t^{DS} 相连；

（2）size-*l* OS 的整体权重 Im(OS, size-*l*)是最大的，也就是其权重为 max(∑ Im(OS, t_i))。

准则（1）保证了 size-*l* OS 中所有 *l* 个节点都是关键词语义上的自描述。在文献[46]中，认为一个较好的 size-*l* OS 是一个跟特定 DS 有关的独立并有意义的最重要节点的集合，用户可以在没有任何冗余信息的情况下很容易地可以理解此结果。因此，*l* 个节点都与 t^{DS} 相连保证了 size-*l* OS 的独立性。例如，在 DBLP 数据集中，对于路径 $R_{Author}\rightarrow R_{Paper}\rightarrow R_{(Co\text{-})Author}$，即使一篇 Paper 的局部权重比一个 Co-author 的小，也不可能去掉 Paper 节点而只留下 Co-author，原因是如果去除了 Paper 节点，那么也就无法保证 Co-author 和 Author 是语义相关联的。

另外，根据准则（1），一个 size-*l* OS 不一定会包含 OS 中前 *l* 个权重最大的节点，比如对于路径 $R_{Author}\rightarrow R_{Paper}\rightarrow R_{Year}\rightarrow R_{Conference}$ 其相应的权重分别为 0.9、0.2、0.7 和 0.6，虽然 Conference 属性节点的权重比 Paper 的高，但是在 size-*l* OS 中可能会去掉 Conference 节点而保留 Paper 节点。所以说 Im(OS, size-*l*)并不代表 *l* 个权重最大的节点的权重之和，是与 t^{DS} 相连的 *l* 个节点组成的集合中最大的权重和。

2.3　本章小结

本章主要介绍了关键词检索技术的相关研究和理论介绍，首先介绍了两种离线权重计算方法，PageRank 和 ObjectRank 的理论概念以及各自的优缺点，为后续提出新方法提供了理论支撑；随后介绍了关系数据集中关键词检索技术中的一些术语和整体的架构流程，也就是当用户给定一个关键词时，如何返回结果集以及返回什么样的结果集，也为后续提出的更深层的检索技术提供了理论支撑。

第 3 章　考虑数值的离线权重计算方法——ValueRank

本章主要介绍一种新的离线权重计算方法—ValueRank，它是 ObjectRank 的扩展。对于数据集中每个元组的 OR 值，除了取决于数量外，仅仅依靠元组之间的数值流动率，而这个值是固定的，忽略了关系之间的语义联系。如在 DBLP 数据集中，Author→Paper 边（如图 2-2），其数值流动率固定为 0.2，假设一位作者 A_1 为计算机领域的权威专家，其 OR(A_1)较高，对于计算机领域的文章，Author 对其 Paper 的数值流动率为固定值是合理的，但是假设 A_1 可能存在几篇群其他领域的 Paper，此时，OR 中考虑固定的数值流动率是比较单一的，故本文提出一种可以考虑节点数值的离线权重计算方法—ValueRank。

3.1　ValueRank 方法介绍

对于关系型数据集，如果其表与表之间影响因素是单一的（如数量等），那么用 PageRank 类方法是很有效的。相反，对于交易类数据集 Northwind（如图 3-1 为 Northwind 数据集模式图），PageRank 虽在一定程度上对其权重的计算给出一定的参考价值，但是其完全忽略了元组之间的数值问题。例如，有两个客户（Costomers）分别为 C_1 和 C_2，其分别下了 100 个订单和 5 个订单，如果 C_2 的五个订单总数值更大，那么 C_2 的权重应比 C_1 更大些。因此，本章提出了一种通用的权重计算方法，它能够考虑关系元组之间的数值问题并且验证此方法适用于任意数据集。

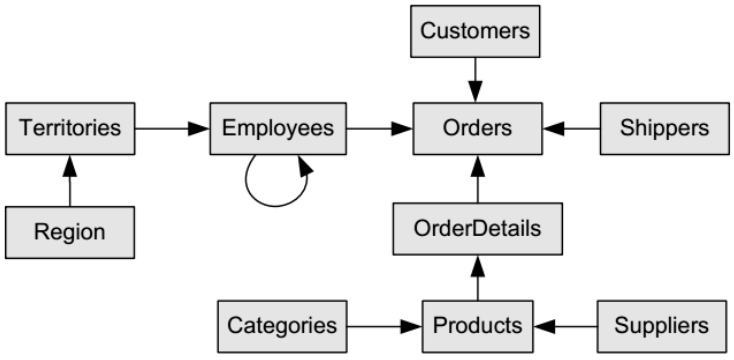


图 3-1 Northwind 数据集模式图

Figure 3-1 The Northwind dataset schema

ValueRank（VR）是 ObjectRank 算法的扩展和延伸，其保留了 ObjectRank 中基本集 S 的概念，将元组之间静态数值流动率扩展为动态计算，也就是通过

VR 得到的权重不仅考虑了元组之间的链接数量关系，还考虑了它们之间的数值关系。基本集 S 中包含的都是对计算其他元组权重值有较大影响的元组。在 DBLP 数据集中，Paper 和 Year 表中的所有元组对其他元组的权重计算都有影响，所以 S 中包含 Paper 和 Year 表中的所有元组。此外，对于 Paper、Year 元组之间的数值流动率可以用归一化函数计算得到。例如，Paper 表中的元组 P_1（2016 年出版，其有一篇为 1996 年发表的参考文献），元组 P_2（2017 年出版，其有一篇为 2016 年发表的参考文献），根据公式算的 Paper 和 Year 之间的数值流动率，然后得到其各自的 VR 值，理想状态下 P_2 得到的 VR 值会比 P_1 高。

ValueRank 具体方法如下：

根据数据模式图 $G(V_G, E_G)$，可以得到其相应的数值流动模式图 $G^A(V_G, E^A)$（如 DBLP 数据集如图 3-2 所示，Northwind 数据集如图 3-3 所示），图中边上的权重反映相邻元组之间的数值流动率，将 E_G 的每条边定义为 $e_G = (v_i \rightarrow v_j)$，随后可以得到相应地两条数值流动边，将其前向数值流动边定义为 $e_G^f = (v_i \rightarrow v_j)$，后向数值流动边定义为 $e_G^b = (v_j \rightarrow v_i)$。最终，根据数据图 D 和数值流动模式图 G^A，可以得到数值流动数据图 $D^A(V_D, E_D^A)$，其中，对于 E_D 集合中的每条边，D^A 中有与其对应的两条边分别为 $e^f = (v_i \rightarrow v_j)$ 和 $e^b = (v_j \rightarrow v_i)$，因此得到两条边对应的数值流动率分别为 $\alpha(e^f)$和 $\alpha(e^b)$，其中如果 $OutDeg(u, e_G^f) > 0$ ($OutDeg(u, e_G^f)$ 是节点 u 的 e_G^f 出度)，$\alpha(e^f) = \alpha(e_G^f)/OutDeg(u, e_G^f)$；否则 $\alpha(e^f) = 0$。$\alpha(e^b)$定义相同。

ObjectRank 值由公式（2-2）得到，但在 ValueRank 中对应每个 v_i 的 s_i 值是由元组的相对权重和由 v_i 的导出的归一化函数计算得到的结果所决定的，在 S 中，每个 v_i 对应的 s_i 值为：

$$s_i = \alpha \cdot f(v_i) \qquad (3\text{-}1)$$

其中 α 是一个调节系数，取值范围为[0,1]，$f(v_i)$是以 v_i 为自变量的归一化函数，其值域为[0,1]，所以 ValueRank 中 s_i 的值域为[0,1]，而 ObjectRank 中 s_i 取值只为 0 或 1。举例来说，对于 Northwind 数据集中 $R_{OrderDetails}$ 中一个元组 v_i，s_i = f(OrderDetails.Price * OrderDetails.Quantity)。s_i 也有可能是相邻元组属性为变量的函数，例如，对于 Orders 表中的元组，s_i = f($\sum$ OrderDetails.Price * OrderDetails.Quantity)。如果 v_i 的入度 >1，那么它的值就是由多个动态数值流动率所决定的。而且同一个元组（出度>1）中的不同属性值可能会影响这个元组不同的链出边，例如，对于边 $R_{Orders} \rightarrow R_{Shippers}$，其数值流动率是 Orders. Freight（Oders 中的 Freight 属性值）为自变量的函数，而对于边 $R_{Orders} \rightarrow R_{Customers}$，其数值流动

率为所有的 Order 的总和（由 UnitPrice * Quantity 计算得出）。ValueRank 中的数值流动率定义为$a(e)'$（包含前向和后向），其可以由以下公式得到：

$$a(e)' = \beta + \gamma \times f(v_i \to v_j) \tag{3-2}$$

其中β 和 γ 是调节系数且 $\beta + \gamma \le 1$，$f(v_i \to v_j)$是自变量为元组 v_i 到 v_j 关系的归一化函数，其值域为[0,1]。图 3-3 为 Northwind 数据集的数值流动模式图。与 ObjectRank 相似，数值流动率、基本集 S 和它们的调节系数需要从实验中得出结论。

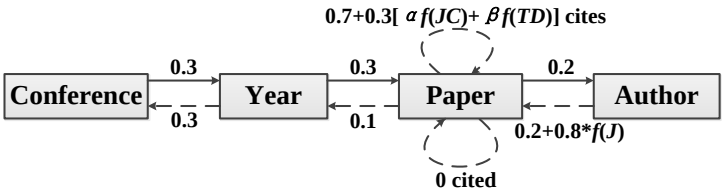


图 3-2 DBLP 数据集数值流动模式图（ValueRank）

Figure 3-2 The G^A for the DBLP dataset(ValueRank)

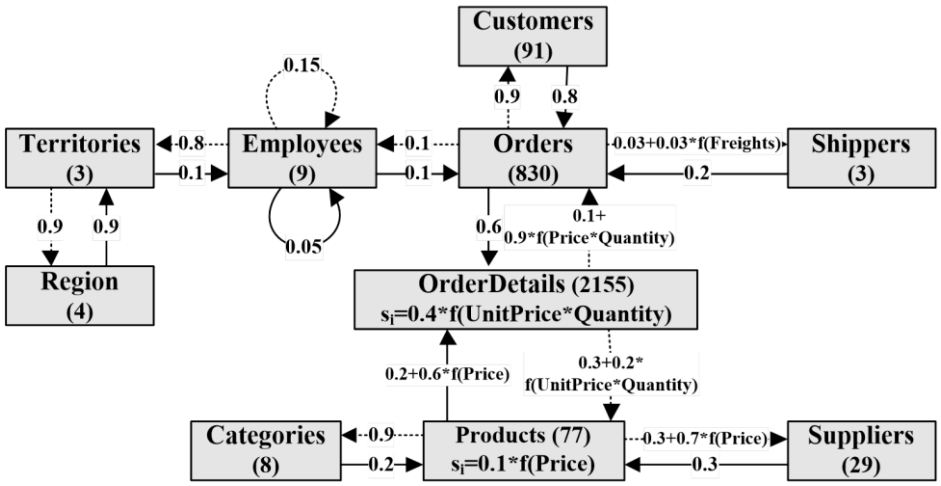


图 3-3 Northwind 数据集数值流动模式图（ValueRank）

Figure 3-3 The G^A for the Northwind dataset(ValueRank)

对于 ValueRank，在 DBLP 数据集中，其数值流动模式图如 3-2 所示，Paper 表中的元组都被认为是可以影响其他元组权重的元组，故 Paper 表中的元组都被加入 S 中，对于边 Paper → Paper，其杰卡德系数（*Jaccard coefficient* ，*JC*）也被认为是可以影响其他元组权重的属性。杰卡德系数主要是用来相似度计算，对于 Paper → Paper，可以根据 *JC* 值来反应论文与被引用论文之间的相似度关系，假设被引用论文有很高的权重且引用论文与被引用论文的 *JC* 很高，那么被引用

论文给引用论文的数值流动率也就相应地较高，故引用论文也将会得到很高的权重。对于 Author → Paper 来说，每个对其主要的领域的论文贡献程度应比次要领域的要高。杰卡德系数公式为：

$$JC(A,B) = \frac{A \cap B}{A \cup B} \tag{3-3}$$

A、B 代表不同的集合，在 Paper 表中由每个元组的论文名称除去停用词之后的所有词组成的集合。假设 Author A_1 发表了三篇文章分别 P_1、P_2、P_3，根据公式 3-3 可以得到 $JC(P_1、P_2)$和 $JC(P_1、P_3)$，将其分别表示为 s_1 和 s_2，若 $s_1 > s_2$，那么 $A_1 \rightarrow P_1$ 的 Jaccard 值为 s_1。数值流动率中的 Jaccard 值 $J(v_i \rightarrow v_j)$计算公式为：

$$J(v_i \rightarrow v_j) = \max[(A-e)(j,:)] \tag{3-4}$$

其中，A 为一个 $n \times n$ 矩阵且 $A_{ij} = JC(n_i, n_j)$（$JC(n_i, n_j)$是 n_i, n_j 的杰卡德系数值），$n_i,\ n_j \in R_{Paper}(n_{Author})$（即一个作者所有的文章）。$e$ 是一个 $n \times n$ 的单元矩阵。$max[(A)(i,:)]$是矩阵 A 中第 i 行所有元素的最大值。J_i 的值域为[0,1)，其取值区间为左闭右开的原因是因为不存在两篇论文完全一样。根据公式(3-2)得到数值流动率 $a(e)'$，其中 $f(v_i \rightarrow v_j)$是 $J(v_i \rightarrow v_j)$的归一化函数。

对于边 Paper → Paper，认为其 Year 属性也对其他 Paper 表中元组的权重计算有很大影响，故提出时间削减率（Time Decrement，简称 *TD*）的概念，一个 Paper 元组 v_i，根据 Paper 引用关系表，得到对其一个引用文章元组 v_j 的时间削减率为：

$$TD(v_i \rightarrow v_j) = \frac{\frac{1}{A_{v_j} + b}}{\sum_{v_j \in P_{v_i}} \frac{1}{A_{v_j} + b}} \tag{3-5}$$

其中，P_{v_j} 是被一个 Paper 元组 v_i 所引用的文章集合，A_{v_j} 是被 v_i 所引用论文 v_j 得年龄，即 $A_{v_j} = y_{v_i} - y_{v_j} + 1$，$y_{v_j}$ 为 v_i 的发表年份，b 是调节系数，用于调整不同年龄的被引用文章的权重，使年龄很小的被引用文章不会获得较高的权重，对于老化快的领域的文章，b 应该取较小值，否则，b 应取较大值，计算机领域一般取值为 $b = 5$。假设 $b = 5$，一篇计算机领域在 1989 发表的名为“*A Knowledge Level Analysis of Belief Revision*”(P_A)的文章有两篇引用文章分别为在 1988 年发表的名

为 “*Investigations into a Theory of Knowledge Base Revision*” (P_B)和在 1986 年发表的名为“*Learning at the Knowledge Level*” (P_C)，故 P_C 的年龄为 4，P_B 的年龄为 2，根据公式（3-5）计算得到的其各自的时间削减率 $TD(P_A \rightarrow P_B)$ 和 $TD(P_A \rightarrow P_c)$ 分别为 0.562 和 0.438。

综上，可以看出数值流动率不一定是由某个属性动态决定，而是由多个属性决定，故可以对 $a(e)'$ 进行改进为 $\alpha(e)$，由以下公式计算得出：

$$a(e) = \beta + \gamma(\sum_{s=1}^{n} a_s f_s(v_i \rightarrow v_j)) \qquad (3\text{-}6)$$

其中， $f_s(v_i \rightarrow v_j)$是属性 s 所对应的归一化函数，a_s 是其对应函数 $f_s(v_i \rightarrow v_j)$（像是 J 或是 TD）对转移率贡献所占的比重系数，且 $\sum_{s=1}^{n} a_s = 1$。

r 是由每个节点 v_i 的 ValueRank 值 r_i 组成的 n×1 的向量，r 由以下公式得出：

$$r = dAr + (1-d)\frac{s}{|S|} \qquad (3\text{-}7)$$

与 ObjectRank 方法不同的是，其中如果 $E_D{}^A$ 中存在 $e = (v_i \rightarrow v_j)$，则 $A_{ij} = \alpha(e)$，s_i 由公式（3-1）得到。$\alpha(e)$由公式（3-6）计算得出。

3.2　ValueRank 与 ObjectRank 对比

本节主要介绍如何在 DBLP 和 Northwind 数据集中利用 ValueRank 算法计算各元祖的权重，以及将 ValueRank 的实验结果与 ObjectRank 的结果进行对比，并对结果进行分析。实验结果证明用 ValueRank 算法得到的权重对检索结果的质量有明显提高，选择这两个纯文本数据集来做比较实验的原因是之前所有的工作都是基于此数据集的实验，故较有可比性。

3.2.1　实验设计

此部分实验主要用了两个数据集，一个为 DBLP 数据集，其模式图如图 2-1 所示，另一个为 Northwind 数据集，其模式图如图 3-1 所示。本文选择这两个数据集来验证 ValueRank 算法的主要原因是它们的模式图比较丰富，其表中有较多的属性，表与表之间有较多的关系。

DBLP 的数据表和关系表及其数据信息如表 3-1 所示。数据表中属性 ID 作用为方便构造关系表。Northwind 数据集表及其对 VR 计算数值流动率有用的属

性信息在图 3-3 中已标出，其中每个表名后括号中的数字为元组个数，也就是对应表的数据总量大小。DBLP 数据集元组总数为 2959511，大小为 513M，而 Northwind 数据集元组总数为 3,209，大小为 1.1M。

表 3-1 DBLP 数据集数据信息表

Table 3-1 The information of DBLP dataset

表名	元组个数	属性
Author	341623	（ID，Name）
Paper	519931	（ID，Name）
Conference	2968	（ID，Name）
Year	11588	（ID，Name）
cy（Conference 与 Year 关系表）	11588	（CY_C，CY_Y）
pa（Paper 与 Author 关系表）	1188553	（PA_P，PA_A）
pp（Paper 与 Paper 关系表）	363329	（PP_P1，PP_P2）
yp（Year 与 Paper 关系表）	519931	（YP_Y，YP_P）

根据不同权重计算方法的不同可知，对元组权重计算有主要影响的为（1）阻尼系数 d，（2）数值流动率。故本实验主要改变阻尼系数 d 和数值流动率中的系数来分析实验结果。

本实验阶段主要用的语言为 Java 语言和 Mysql 数据集，自 2016 年 8 月 1 日至 2017 年 2 月对 DBLP 数据集和 Northwind 数据集进行实验，主要步骤如下：

（1） 通过调整阻尼系数 d 进行对比实验，分析实验得到使结果较好的默认值 d；

（2） 计算各元祖的 ObjectRank 值；

（3） 计算 ValueRank 数值转移率中各参数值，例如 DBLP 库中的 *JC* 和 *TD*；

（4） 调整数值流动率中的系数参数，计算各元组的 ValueRank 值；

（5） 分析比较结果。

步骤（3）的主要作用是降低单次计算的时间复杂度，将 *JC* 和 *TD* 等在 ValueRank 计算之前计算完成减少整体计算的时间复杂度，有助于中期实验结果的保存，在后续计算出现问题时避免重复计算。后期根据 ObjectRank 中用到的调查评分评估方法，来对本实验进行评价打分，具体来说，随机选择我校五位教师和同学来为此次实验结果进行打分，每个参与者在此之前并没有参与本实验且对此权重计算方法毫不了解。每个参与者从所有元组中随机选择 10 个对其权重计算结果进行打分（1 到 10），对于每个元组，还提供了一组元组的描述性细节（属性信息）和统计数据。例如，在 Northwind 数据集中，对于 Employee（或者 Customer），来说，提供了 Oder 的总数、每个 Oder 的大小和数值以及数据

集中所有 Employee（或者 Customer）相应数据的最小值、中位数和最大值。提供这些详细信息有助于评估人员对我们的算法进行更好地评估。

3.2.2　实验对比与分析

本文沿用了 ObjectRank 算法中用到的评价方法，即调整系统参数（阻尼系数 d 或调整数值流动率中的参数）来评价和比较实验结果。

（1） 在保证为同一 G^A（如图 3-2）的情况下，令 d 取值分别为 0.85，0.10 和 0.99（默认为 0.85）。

（2） 然后再保证 d 为同一取值的情况下，改变 G^A 中决定数值流动率的系数（代表不同的 G^A），在 Northwind 数据集中，对三个不同的 G^A 进行实验，而在 DBLP 数据集中，对四个不同的 G^A 进行实验。

对于 Northwind 数据集，G^{A1} 为图 3-3；相应地 G^{A2} 为令所有边中的系数 β 和 γ 都取值为 0.3，也就是令 $\alpha(e) = 0.3+0.3f(.)$；G^{A3} 定义为令所有的 α 设定为 0，得到结果即为 ObjectRank 算法计算的结果。

对于 DBLP 数据集来说，本文提出了两个可以影响元组权重结果的因素，即 JC 和 TD，对于边 Paper→Paper，根据公式（3-6）得到 $\alpha(e)=0.7+0.3(a_1 f_s(JC) + a_2 f_s(TD))$，所以需要调节的参数为 a_1 和 a_2，故 G^{AI}定义为取图 3-2，令 $a_1 = 0.5$ 且 $a_2 = 0.5$；G^{AII}为图 3-2 中令 $a_1 = 0.1$ 且 $a_2 = 0.9$；G^{AIII}为图 3-2 中令 $a_1 = 0.1$ 且 $a_2 = 0.9$。G^{AIV} 为图 3-2 中令所有的 a 都为 0，即 ObjectRank 值。

表 3-2 为系统参数的取值和默认值表，其中在后续实验中默认取 $d = 0.85$。

表 3-2 系统参数的取值和默认值表

Table 3-2　System parameters of ranges and default values

参数	范围
G^A	G^{A1}, G^{A2}, G^{A3}，G^{AI}，G^{AII}，G^{AIII}，G^{AIV}
$d(d_1, d_2, d_3)$	0.85（默认）, 0.10, 0.99

根据上节所述的调查评分评估方法，来对本实验进行评价打分，表 3-3 为所有参与者打分后计算总和的平均分数，其中例如 G^{A1}- d_1 表示表 3-2 中参数 G^A 为 G^{A1}，而参数 d 为 0.85 的实验名称。根据此次调查可以看出参数 d 调整对 G^{A1} 的影响不是很大，调整 d 取值从 d_1 到 d_3，对其结果评分差别很小。根据表 3-3 还可以看出，由 G^{A2} 计算出来的结果也是比较令人满意的，这表明只要我们知道对元组权重影响较大的表中属性，如 Price、Freight 等，那么选择一个可以得到较好结果的 G^A 是很容易的，换句话说，调整数值流动率中的参数对结果满意度的影响其实是比较小的。但是对于 ObjectRank 计算的结果，即通过 G^{A3} 得到的结

果对于参与调查的参与者来说，其结果并不令人满意，这主要是因为 ObjectRank 只考虑到元组间的链接数量关系，却忽略它们之间所存在的数值关系。

表 3-3 ValueRank 算法效果评估

Table 3-3 Evalustion of ValueRank effectiveness

实验参数名称	G^{A1}- d_1	G^{A1}- d_2	G^{A1}- d_3	G^{A2}- d_1	G^{A3}- d_1
评分	7.8	8.1	7.8	7.8	3

表 3-4 给出了在 Northwind 数据集中通过 ValueRank 方法计算的 VR 值，其中阻尼系数 d 取值为默认值（0.85），G^A 选择 G^{A1}，还根据 ObjectRank 的 G^A（即 G^{A3}，相当于对于所有的边取 $\alpha(e) = \beta$）给出了相应元组的 ObjectRank 值，即 OR 值。通过此结果我们可以得到以下结论：在 Northwind 数据集中，其元组的 OR 值与 Orders, OrderDetails 等的数量有较大的关联，即 OR 值与元组所对应的 Orders, OrderDetails 等属性的数量成正比，但其元组的 VR 值却与 Orders, Freight 等的总数值相关联，即 VR 值与元组所对应的 Orders, Freight 等属性的数值总和成正比。例如，对于一个名为 SAVEA 的 Customer 有 31 个 Order，其订单总计为 UnitPrice × Quantity = 115,673.30，而对于另外一个名为 QUICK 的 Customer 有 28 个 Order，其订单总计为 UnitPrice × Quantity = 117,483.39。跟据表 3-4 可以看出前者的 OR 值大于后者的 OR 值，这是因为前者的 Order 总数大于后者，而后者的 VR 值大于前者的 VR 值，这是因为后者的 Order 数值总和大于前者。在表 3-4 中可以看到对于 Product 和 Supplier 可以看到相似的结果。Employee 4 和 Shipper 2 的 OR 值和 VR 值都比较高是因为它们的 Order 数量和 UnitPrice × Quantity 或是 Freight 数值都很大。

表 3-4 Northwind 数据集 ObjectRank 和 ValueRank 权重结果样例（每个关系的最大值用粗体表示）

Table 3-4 Samples of ObjectRank and ValueRank scores in Northwind Dataset (Maximum values per relation are indicated in bold)

Tupple ID	V.R.	O.R.	Orders	{UnitPrice*Quantity, Feight† , Price†† }
Cus_SAVEA	0.654	**0.702**	**31**	115,673.4
Cus_QUICK	**0.691**	0.616	28	**117,483.4**
...				
Shiper 1	0.198	0.359	249	16,185.3†
Shiper 2	**0.272**	**0.470**	326	28,244.8†
...				
Product 38	**1.000**	0.486	24	**149,984.2**
Product 59	0.495	**1.000**	**54**	76,296.0
...				
Employee 4	**0.384**	**0.375**	**156**	**250,187.4**
Employee 3	0.352	0.304	127	213,051.3
...				
Supplier 18	**0.043**	0.086	2	**281.5††**
Supplier 7	0.023	**0.132**	**5**	177.8††
...				

表3-5给出了在DBLP数据集中VR值与其对应OR值的比较，令 R^{G1}, R^{G2}, R^{G3} 和 R^{G4} 分别对应其通过 G^{AIV} 得到的 OR 排序值，通过 G^{AI} 得到的 VR 排序值，通过 G^{AII} 得到的 VR 排序值和通过 G^{AIII} 得到的 VR 排序值，阻尼系数 d 取值为默认值（0.85），排序值是指对于某个表所有元组的权重从大到小排序后的排名结果，用排序值来比较结果的主要原因是对于DBLP数据集其OR值和VR值差距较大，所以用排序值来比较更为合理。

Author 表中所有元组总数为 341,623。Author A_1 的 R^{G1} 为 4，R^{G2} 为 2，这是因为 A_1 发表的论文主要是数据库方向的，可以说是其研究方向为数据库，而在 DBLP 数据集中 A_1 的其他领域的论文几乎没有，令与其研究方向不相关领域的论文数量为 n_{ur}，其论文总数为 n_{sum}，则可以得到一个非相关率 $r_i = n_{ur}/n_{sum}$，r_i 值越大也就导致其 VR 值排名比 OR 值排名越低，相反，A_2 的 R^{G2} 比 R^{G1} 高，因为 A_2 的研究方向相对集中，主要为计算机科学与技术领域。Author 的 R^{G3} 和 R^{G4} 根据相应的侧重因素改变，但通过不同 G^A 计算的 VR 值和其排名大致相同。其中 R^{G3} 更侧重于 *TD* 指数较高的元组，而 R^{G4} 则侧重于 *JC* 指数较高的元组。Paper 元组的排序方式如同 Author，Paper A_1 的 R^{G2} 比 R^{G3} 高，因为其 *JC* 高，也就是这篇文章与其参考文献很相关，且其 *TD* 也较高，即这篇文章的参考文献的年龄都

较小，检索到这样的结果对用户很有意义。相应地，若过多关注 *TD*，可以看到 Paper B_1 和 B_2 的 VR 排序值会有相应的改变，若过多关注 *JC*，可以看到 Paper C_1 和 C_2 的 VR 排序值会有相应的改变。更多地实验结果如表 3-5 。

表 3-5 DBLP 数据集 ObjectRank 和 ValueRank 权重结果样例

Table 3-5 Samples of ObjectRank and ValueRank scores in DBLP dataset

Tuple ID	R^{G1}	R^{G2}	R^{G3}	R^{G4}
Author A_1	97,763	210,913	--	--
Author A_2	4	2	--	--
Author B_1	--	45	47	--
Author B_2	--	37,187	35,196	--
Author C_1	--	777	--	765
Author C_2	--	934	--	925
…				
Paper A_1	37	8	--	--
Paper A_2	454	3896	--	--
Paper B_1	--	13	11	--
Paper B_2	--	8	12	--
Paper C_1	--	12	--	6
Paper C_2	--	15	--	23

3.3 复合型数据 ValueRank 计算

本节主要将 ValueRank 用于北京 POI 空间数据，由于 ObjectRank 和 PageRank 只适用于跟数量关系有关的关系数据集，而本文中提出的 ValueRank 适用于任意的数据集，再次利用北京 POI 空间数据构建数据集来证明了此方法的通用性。首先设计数据模式图，由于数据格式的无结构性，故在此需将数据进行清洗、转换、入库等操作，还将原空间数据通过 R 树索引的方法建立商圈，构建数据关系模式图，计算各元组 VR 值，分析实验结果。

3.3.1 实验设计与过程

由于之前文献中只对 ObjectRank 和 PageRank 示例数据集进行研究，此实验将证明本文所提出的 ValueRank 将适用于任意数据集。故，将 ValueRank 思想运用到北京 POI 空间数据集中。步骤如下：

（1） 获取 2016 年百度地图[1]北京 POI 空间数据集，清洗数据，转换数据格式。

表 3-6 北京 POI 空间数据集原始格式

Table 3-6 The old format of Beijing POI spatial set

3960 11600 3961 11601 中餐厅 2016-08-30 17:31:31 {"uid": "3c0d9c8309dd41d86c4760ec", "detail_info": {"service_rating": "3.1", "comment_num": "7", "image_num": "3", "tag": "美食", "environment_rating": "3.1", "di_review_keyword": [],"detail_url": "http://api.map.baidu.com/place/detail?uid=3c0d9c8309dd41d86c4760ec&output=html&source=placeapi_v2", "type": "cater", "overall_rating": "3.9"}, "telephone": "(010)89382390", "detail": 1, "location": {"lat": 39.607719, "lng": 116.004851}, "address": "北京近郊房山房琉路", "name": "腾翼人家"}

数据集中共 570331 条数据，其中每条数据格式如表 3-6 所示，本步骤主要运用 Python 来处理、清洗数据。首先将数据存储为 txt 格式，由于其不利于提取字段，故将每条数据前部分如“3960 11600 3961 11601 中餐厅 2016-08-30 17:31:31”去除，将数据转换成 json 格式。其中每条数据中对接下来实验有用的字段为“tag”、“overall_rating”、“comment_num”、“name”、“type”、“lat”和“lng”。其对每个字段的说明如表 3-7 所示。最后将每条数据中存在这 8 个字段的数据提取出来，若不存在则以空补全。以 json 数据格式存储。

[1] https://map.baidu.com/

表 3-7 北京 POI 空间数据集字段说明

Table 3-7 The interpretation of fields of Beijing POI spatial set

字段名	说明
name	POI 地点名称
tag	地点属性标签
type	地点分类的类型
overall_rating	用户评价分数（最高 5 分）
comment_num	参与评价的用户数量
lat	纬度
lng	经度

（2） 将数据导入 Mysql 数据集中

本步骤首先将 json 格式的数据存储到 mysql 的一张表中，然后根据需求，将表中 name、tag（以分号分开）、type 属性分别存储到另外一张表中，并去重且进行标号（Id）。

（3） 运用 R 树索引进行商区划分

本步骤主要根据 R 树索引来划分北京地区的商区，由于现存的线上应用软件所在的商区都是根据人工标注进行划分的，但是在本文将根据 R 树索计算商区。主要运用 Python 语言进行 R 树索引的构建，构建后的对叶子节点进行分析，去除长宽比较大的叶子节点，因为此类空间节点影响后续商区的划分。根据经纬度距离公式选取相应面积叶子节点作为商区标准，如图 3-4 为所有面积小于 0.018km^2 的部分叶子节点区域（框体为区域边界，带数字框体中的数字为叶子区域的 id），这些叶子节点将作为商圈区域划分的候选区域节点是符合事实情况的，其中这些区域除了是地铁沿线再多数都是我们已经了解过的比较繁华的区域。

关于商圈 Business Circle(BC)的计算算法如下：

BC 计算算法

1） 将所有 POI 数据按坐标构建 R 树索引
2） 分析叶子节点，去除长宽比小于 1/4 的叶子节点
3） 去除面积大于 0.018km^2 的叶子节点
4） 计算所有剩余叶子节点区域的中心坐标 o_i
5） 遍历所有剩余叶子节点，对于每个叶子节点 r_i 做如下操作：
6） 若 r_i 没有加入商圈则：
找到以 o_i 为中心 300m 范围内所有叶子节点中心集合 O = $\{o_1,o_2,o_3,...,o_k\}$ 将此集合中所有中所有以中心点 o_j，$o_j \in O$ 为中心的叶子节点和以 o_i 为中心的叶子节点加入同一个商圈
7） 否则：
continue
8） 返回所有商圈

最终构建所有商圈的总数为 387 个。根据叶子节点和商圈的关系，找到商圈内所有的 POI，构建 POI 与商圈之间的关系。

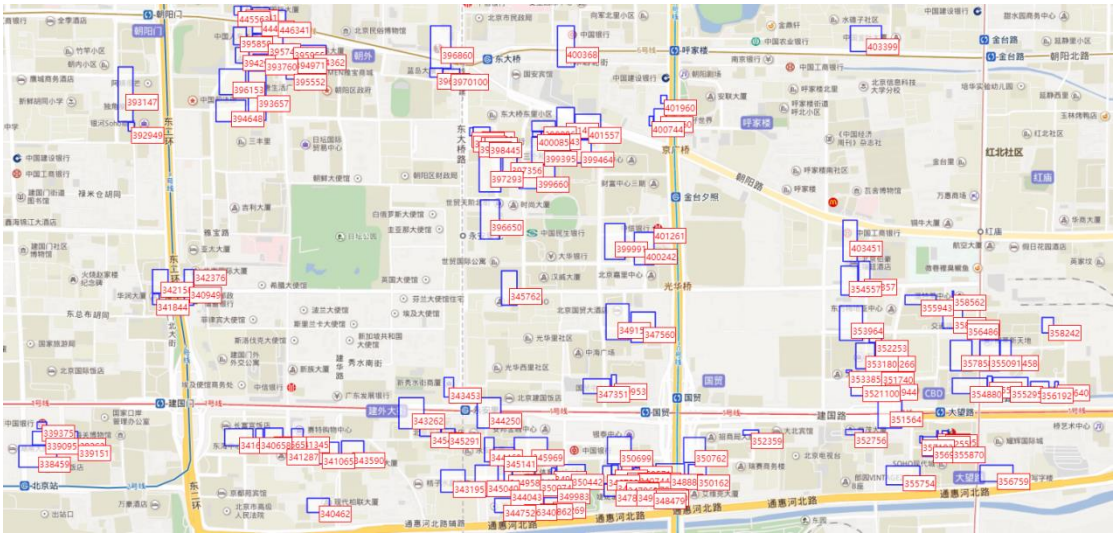


图 3-4 北京 POI 空间数据 R 树部分叶子节点（商圈备选节点）

Figure 3-4 Some R-tree leaf nodes of Beijing POI spatial data (commercial circle candidate node)

（4） 构建数据表关系，得到 G^A

至此，得到所有的基础表，根据他们之间的关系构建关系表（id 与 id 之间），此处用 id 构建关系也是为了方便计算。其构建完的北京 POI 空间数据集相关数据信息表如表 3-8 所示，随后根据数据集中表与表之间的关系构建的 ValueRank 的 G^A 如图 3-5。最终，根据 G^A 计算其 VR 值与起始 over_rating（即用户评分）进行分析比较。

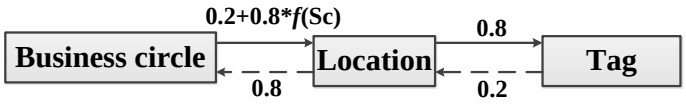


图 3-4 北京 POI 空间数据集数值流动模式图（ValueRank）

Figure 3-4 The G^A for the Beijing POI spatial dataset(ValueRank)

在北京 POI 空间数据集中，对于边 Business circle→Location，本文在其用户评价分数的基础上考虑其商圈内综合评分 *Sc*，其中 *f*(*Sc*)由以下公式得出：

$$f(Sc)=\frac{r}{5.0}*\frac{n}{\max(n_i)},n_i\in B_{jt} \qquad (3\text{-}8)$$

其中 *r* 为此 POI 即 Location 的用户评价分数，取值为[0,5]，*n* 为此 Location 用户评价条目数，max(n_i)为此 Location 所在的 Business circle B_j 中 type *t* 中最大的用户评价条目数，故 *f*(*Sc*)的值域为[0,1]。每个 POI 的 type 为大类别，共十类分别为（cater，life，house，enterprise，shopping，education，hospital，hotel，beauty，

scope）。这里将商圈中不同 type 之间计算是非常有意义的，因为每个商圈中可能存在多个 type，但是每个 type 之间若只用一个标准其实并没有可比性，比如商圈 B_1 中 type 为 shopping 的 max(n_i)为 788，其对应的 over_rating（用户评价分数）为 4.0，但 type 为 hotel 的 max(n_i)为 5，其对应的 over_rating 为 5.0，故本文将不同 type 分开进行比较计算，这样可以得到更有比较意义的结果。

表 3-8 北京 POI 空间数据集数据信息表

Table 3-8 The information of Beijing POI spatial dataset

表名	元组个数	属性	描述
l	462437	（id，name）	POI 的 id 与名称对应关系表
l_datail	490601	（l_id，over_rating，comment_num，type）	POI 具体属性表
l_co	490601	（l_id，lat，lng）	POI 具体坐标表
tag	158	（id，name）	POI 属性标签
lt	480398	（l_id，t_id）	POI 与标签的关系表
lb	52181	（l_id，l_id）	POI 与商圈的关系表

3.3.2 部分实验结果分析

上节实验中已经对 d 的取值做了实验比较，故本节直接沿用上节实验的结果，取 d 为 0.85。本节主要是为了计算北京 POI 空间数据集中各 POI 的初始权值，为以后检索做准备，故本实验所要比较的是用 ValueRank 计算得到的权值和原来的以用户评价分数的权值相比较。由于没有量化方法对此方法进行分析，所以本节中只是随机选择一个商圈的部分节点来说明本方法的效果和有效性。

在一个备选范围内所有点按照权值降序排列到列表中，检索系统首先选则的是权值较高的备选点，故虽然用户评价分数 r 和其 VR 值的值域并不相同，但可以通过备选范围内权值的降序排序结果来比较其权值的变化结果。表 3-9 为选择商圈为 1114 的部分点的实验结果。首先提取商圈所有点的信息，然后去除 r 为空（None）的点，剩余点个数为 247，将这 247 个点分别以 r 和 VR 值降序排序，表 3-9 中 s_r 为按 r 值排序后的排名，同理 s_V 为按 VR 值排序后的排名。

表 3-9 北京 POI 空间数据集数据部分点 VR 值与 *r* 值

Table 3-9 VR and r of some points of Beijing POI spatial dataset

id	*r*	n	type	VR	s_r	s_V
14904	4.1	**788**	cater	0.486	15	2
288397	3.9	84	cater	0.167	23	196
71892	5.0	12	cater	0.167	4	233
16722	2.4	1	cater	0.167	68	238
29629	4.1	**469**	life	0.236	14	4
207704	5.0	6	life	0.176	3	41

根据表 3-9 可以看出 id 为 14904 的点属于 cater 类型，其 *r* 值为 4.1，但 n= =788 为此商圈的最高的 *n* 值（加粗的值为此商圈中此 type 中最大 *n* 值），此时，*f*(*Sc*) = 4.1/5.0 * 788/788 = 0.82，而对于 id 为 71892 的点，其 *r* 值为 5.0 但 *n* = 12 故此时 *f*(*Sc*) = 5.0/5.0 * 12/788 = 0.0152，由此可见根据图 3-4 的 G^A 对于 VR(cater)对这两个点的数值流动率分别为 0.856 和 0.21216，故 id 为 14904 的点对用户来说可能有更好的参考价值。显然根据 *r* 和 VR 值得排序情况来看，id 为 14904 得点从 15 上升到 2，而 id 为 71892 的点从 4 跌到 233（点的总数为 247）。可见排序的结果跟本文之前预想的是一样的，表 3-9 还列出了对于 life 类型的两个点，结果与意义与 cater 类型是一样的，在此不做过多阐述。

3.4　本章小结

本章主要提出 ValueRank 方法来计算元组的初始全局权重，为以后检索得到结果排序做准备。

首先说明了 ObjectRank 和 PageRank 存在的不足，介绍 ValueRank 的理论思想和相关计算原理。在 ValueRank 中引入了元组数值的概念，例如评价一个消费者，不应以订单的总数来评价，而还与每个订单的数值来计算其权重，随后讲述了如何在 DBLP 数据集中和 Northwind 数据集中计算每个元组的 VR 值，在构建 DBLP 的 G^A 时，针对 Paper→Paper 边提出了 *TD*（时间削减率）和 *JC*（杰卡德相似系数）的概念，削弱 Paper 元组中引用文献集合年龄较大的权重，也对与引用文献不太相关 Paper 元组削弱其权重。而对于 Author→Paper 边也引入 *JC*（杰卡德系数）来削弱与 Author 主要研究领域不太相关的 Paper 元组的权重。

随后，通过对 DBLP 数据集和 Northwind 数据集的实验说明 ValueRank 比 ObjectRank 计算得到的元组权重更有意义。而且 ValueRank 更适用于一般的数据集，而不像 ObjectRank 只适用于数据表之间有数量关系的数据集。

最后，使用北京 POI 空间数据集进行实验，首先根据 R 树的叶子节点计算得到商圈，而不是根据之前人工标注得到商圈的范围，计算得到的商圈比人工标注划分更有意义和说服力，随后为了计算各点的 VR 值，提出 *Sc*（商圈综合评分）的概念，根据每个商圈和点计算得到的 *Sc* 不同，每个点得到的数值流动率也不同，也就是说是动态变化的，最终得到每个 POI 的初始化权重。说明此方法得到的权重，一是避免了传统线上应用单一以用户评价作为其初始权重，二是还考虑到商圈对权重的影响，动态分配数值流动率，这比静态指定的数值流动率更有意义。此章主要用于计算元组的初始权重，对检索结果排序有很大部分的参考价值，在接下来的章节主要对语义和空间检索方法进行研究。

第 4 章　基于语义多样性和等比例特性的检索方法

本章主要介绍针对语义提出基于语义多样性和等比例特性的检索方法。现有的检索方法得到的检索结果都是根据与关键词（在空间数据检索中为“坐标位置”无“关键词”）相关的词条按权重排列得到的（在空间数据检索中为按距离远近排序，忽略 POI 之间的语义关系），而这种方法的缺点是，在用户描述比较模糊的时候，并不能准确地返回精准结果，也就是无法了解用户意图。根据上述需求，提出基于语义多样性和等比例特性方法，以此得出的检索结果可以将多样性的词条推送给用户，此时用户可以根据多样性语义的结果选取自身的需求。除此之外，，在空间中，本文考虑在用户不给出关键词的情况下，也能根据用户所给的坐标位置返回给用户基于语义和空间分布多样性和比例性的检索结果（以下亦同）。本章还提出基于语义的等比例特性的检索方法，比如某个区域的小吃较多，但景点较少，那么此处可能是小吃街，所以在结果中也考虑每个类别的等比例特性，使返回结果尽可能保证原来的结果集中的占比。

4.1　多样性和等比例特性

通过上章所述离线权重计算方法 ValueRank 得到各个元组 v_i（$v_i \in V$）的的全局权重即 $gi(v_i)$，根据 2.2 节节对 OS 和 size-l OS 的介绍，当用户给定关键词，即得到 DS，进而得到 R^{DS}，随后根据 G^{DS} 可以得到完整的 OS，由于 OS 为树形结构，所以令它包含的节点为 $n_1,\ldots, n_k$（在第二章中节点用 t 表示），根据图 2-3 所示，OS 中的节点可能有多次重复的可能。例如 n_i 与 n_j 相同，即 $n_i = n_j = v_i$ 且 $gi(n_i) = gi(n_j) = gi(v_i)$。在此定义 $fr(v_i)$（也为 $fr(n_i)$）为 v_i 在给定 OS 中出现的次数（频率(fr)），例如在 DBLP 数据集中，对于关键词为“M. Faloutsos”生成的 OS 中包含 12 次“C. Faloutsos”（v_k），即 fr（v_k）=12。而这种多次重复的且权重较大的元组在结果集中对结果质量影响较大，由于现在用户检索的意图都是隐形意图，故在此考虑结果的多样性和等比例特性，以此来削弱此类元组的权值。

基于多样性的检索通常通过权衡查询问题 q 与结果集的相似性来提高查询结果的质量，相似性在文本关系数据集中表现为与查询问题 q 语义上的相似关系，而在空间数据集中查询问题为坐标位置，相似性与 q 距离特定范围内。通常，给定一个查询 q 和结果集合的规模 k（自然数），可以得到一个按 $sim(s_i, q)$从大到小排好序的结果集 S，$S = \{ s_1, s_2, \ldots, s_n \}$，其中 $n \geqslant k$。多样性的结果是指从 S 中查找一个子集 R，R 满足以下两个条件：

1) R 的大小为 k

2) R 中每个元素都与 q 相似（写为 $sim(q, s_i)$），于此同时 R 中的每个元素之间都尽可能不相似（写为 $dis(s_i, s_j)$）。

一般情况下 $dis(s_i, s_j) = 1- sim(s_i, s_j)$。

本文根据 2.2.2 提出两种类型的 size-l OS，即基于语义多样性得到的 *se*Dsize-l OS 和基于语义等比例特性得到的 *sp*Psize-l OS，两者都是在 size-l OS 定义的基础上分别引入了多样性和等比例特性的概念。一个有意义的 *se*Dsize-l OS 和 *sp*Psize-l OS 应该分别将各个节点的多样性得分、等比例特性得分和局部权值（li）巧妙地结合。因此，本文提出对于 OS 中的每一个节点，其相应的权值为结合这三个因素（可表示为 $dv(n_i)$, $pq(n_i)$ 和 $li(n_i)$（由公式（2-4）得到，也可写为 Im(OS, n_i)））最终得到一个单独的权值（$dw(n_i)$ 和 $pw(n_i)$，如果在没有歧义的情况下可以简写为 w）。最终我们选择的规模为 l 的结果集为整体权重和最大（即 max($\sum w_i$)）的集合。其中每个节点的 li 如 size-l OS（2.2.2）中介绍，由公式（2-4）得出，故在此我们只讨论多样性和等比例特性得分的计算方法。

在介绍方法之前需要对节点之间的相似性的评定做一个规定，本文中定义的相似性即 $sim(s_i, s_j)$为 s_i 完全等于 s_j，目前对于相似度的计算方法有很多，但是在本文中一是主要的研究方法不在于此，二是对于北京 POI 空间数据集，本文只对“tag”（即 POI 的属性类别）字段进行相似度计算，而“tag”都是人工标注好的，故并不需要进行相似度计算；对于 DBLP 数据集，对“Author”表进行相似度计算也无任何意义。故在以下研究中只认为如果 $s_i = s_j$，则 $sim(s_i, s_j) = 1$；否则 $sim(s_i, s_j) = 0$。

4.1.1 语义多样性（*se*Dsize-l OS）

本文中还建议结果集中 l 个节点应该具备多样化的特性，这样就可以防止重要节点的控制整个 size-l OS。例如在北京 POI 空间数据集中，距离查询点越近的点具有更高的权值，而可能附近的 POI 都为餐饮类，则根据权值排序返回的结果集并不具有代表性。衡量多样性的一个标准是最大化每个节点之间的相异性之和。因此，对于给定的 OS 中的节点，提出估计多样性系数的方法如下：

$$dv(n_i) = 1 - \sum_{n_j \in S, n_i \neq n_j} \frac{sim(n_i, n_j)}{l-1} = 1 - \frac{z(n_i) - 1}{l-1} \qquad (4\text{-}1)$$

如果 $n_i = n_j$，则 $sim(n_i, n_j) = 1$;否则为 0。因此，当前从完整候选 OS 选择到的 size-*l* 片段中与 n_i 的相似节点的总和为 $z(n_i) - 1$，其中 $z(n_i)$是片段中 n_i 出现的次数。除以 *l*-1 是为了将 $dv(n_i)$归一化在[0,1]的范围内。将任何节点的多样性表示为 *dv*[*z*]，其表示它在片段中出现 *z* 次。例如，对于所有节点都只出现一次，那么 dv [1] = 1。在例如，考虑 *l* = 10 并且“C. Faloutsos”出现 2 次（即 *z* = 2），则 dv [2] = $1 - \frac{2-1}{10-1} = \frac{8}{9} = 0.89$。注意这个分数对应于图节点 n_i，因此这两个节点将具有共同的 *dv*，即 0.89（另一种方式是将 n_i 第一次出现 *dv* 为 1，第二次出现 *dv* 为 0.78，因为 1 + 0.78 = 0.89 + 0.89）。这个公式受以下规则所启发：

（1）最大化多样性，指相关性最大化的一个集合中不相似性元组分数的总和最大化；

（2）公式将元组与关键词之间的相关性和元组之间的相异性结合成一个单一的值，作为此元组的值。

值得注意的是此公式是用来削弱备选 OS 中类元组（即完全相似的元组，在 DBLP 数据集中指“Author”字段，空间数据集中指“tag”字段）的局部权值，而并不是将元组的 *li* 和 *dv* 进行按系数地相加，因为有可能元组的 *li* 的值域为[0,10]，但是 *dv* 的值域总是在[0,1]范围内的。因此与大多数多样化方法不同，我们考虑的并不是多样性与其局部权值得总和，其结合后得到得权值计算公式将在 4.1.3 中提出，关于多样性的实验结果将在 4.2 和 4.3 中写到。

4.1.2　语义等比例特性（*se*Psize-*l* OS）

我们观察到，一个备选的 OS 中会有频繁出现的元组（节点），例如在 DBLP 数据集中，“Michalis Faloutsos” OS 中，有 37 个“S. Krishnamurthy”节点，12 个“C. Faloutsos”，有 18 个“INFOCOM and Computer Com. R.”Conference 节点，根据上面介绍定义其频率为 *fr*。有时候这些节点的局部权值相对较低，但是它们在 OS 中的频率也代表着它们与 DS 之间有着较强的联系，所以它们的这种关系应该将被考虑在被选中的有效片段中。因此，本文还提出考虑频繁节点的等比例特性。在不考虑局部权值因素（即考虑到所有节点具有共同的 *li*）的情况下，本文认为这样的频繁节点应该按比例表示在最终的 size-*l* OS 结果中。即在理想情况下，如果一个节点在 1259 个 OS 节点中出现 37 次，那么它应该出现 $l \cdot \frac{37}{1259}$ 次在相应的 *se*Psize-*l* OS 中（这实际上不可能，因为也可能需要中间节点）。

由于 l 的规模较小，因此我们需要一些增量选择策略将不同的频繁集合按比例分布，即在每次迭代时选择一个不同的频繁节点（某种程度有利于循环选择）。为此，我们建议使用的等比例系数如下：

$$pq(n_i)=\frac{fr(n_i)}{\alpha\cdot z(n_i)+1} \qquad (4\text{-}2)$$

其中，$z(n_i)$为从完整 OS 选择到 size-l OS 片段中 n_i 已经存在的次数（也就是在用公式（4-2）计算之前，它已经被选择了 z-1 次），$fr(n_i)$为 n_i 在完整 OS 中出现的次数，α 是一个可以调整的比例参数。这个公式的灵感来源于 Sainte-Laguë 算法[23]（其中 α = 2），根据经验，认为此公式对本文要解决的问题很有效。对一个给定的 $z(n_i)$，可以得到其对应的等比例特性系数 $pq[z]$，此公式的原理为促使当向片段中增加一个频繁节点时，其 $pq[z]$会显著削弱，这样它类的节点可能被选择的概率会增加。此方法其实也促成了结果的多样性。例如，对于“C. Faloutsos”，其 fr = 12 且令 α = 2，那么首次选择这个节点时，可以得到 $pq[1]$ = 12/3 = 4，而第二次为 $pq[2]$=12/5=2.4。

在此还需要说明以下节点添加次序的准则，选择节点的顺序将决定其相应节点相应的 $pq[z]$，2.2.2 节中讲过，我们将选择整体权值最高的节点集合作为最终的 *se*Psize-*l* OS。因此，将会选择整体权值最大化即按 Im(*PSl*)（如公式（4-4））大小顺序来选择节点。例如，考虑到节点 $n_i = n_j$，它们的局部权值相等，如果先选择 n_i，那么可以得到 $pq(n_i)[1]$，那么此时整体权值为 Im(PSl_1)；但如果先选择 n_j，那么可以得到 $pq(n_j)[1]$，那么此时整体权值为 Im(PSl_2)，然后将选择致使整体权值最大化的那个节点，即选择 *argmax*(Im(PSl_1), Im(PSl_2))。如果没有这样的约束，那么我们可能得不到准确的整体权值（即对于相同的 *se*P 或 *se*Dsize-*l* OS，如果选择相同节点的次序不同，那么可能会产生不同的整体权值），即避免了整体权重的随机性。

4.1.3 *se*Dsize-*l* OS 和 *se*Psize-*l* OS 定义

基于以上讨论，对于 *se*Dsize-*l* OS 中的每个节点提出以下公式来计算结合后的语义多样性权值：

$$dw(n_i)=li(n_i)\cdot dv(n_i) \qquad (4\text{-}3)$$

其中 $li(n_i) = af(n_i) \cdot gi(n_i)$（如公式（2-4））即 n_i 的局部权值，dv 是多样性系数（如公式（4-1））。

定义 1（*se*Dsize-*l* OS）：给定一个 OS 和 *l*，一个 *se*Dsize-*l* 是 OS 的一个满足以下条件的子集：

（1） *se*Dsize-*l* OS 的大小为 *l*（其中 *l*≤|OS|）

（2） 在文本数据集中所有的节点都与根节点（n^{DS}）相连，在空间数据集中，所有节点都在以根节点（n^{DS}）为圆心，指定半径的圆的范围内

（3） 对于每个节点可以根据公式（4-3）得到一个权值 $dw(n_i)$

（4） 对于一个 *se*Dsize-*l* OS 整体权值可以由以下公式计算：

$$\mathrm{Im}(DSl)=\sum_{n_i\in DSl} dw(n_i) \qquad (4\text{-}4)$$

令 OS 中任意一个满足上述条件（1）-（3）的子集为备选的 *se*Dsize-*l* OS，其最佳的 *se*Dsize-*l* OS 为所有备选子集中 Im(*DSl*)为最大值的子集。

类似可以得出 *se*Psize-*l* OS 定义。

4.2　实验结果与分析

在之后的实验中需要注意的是，首先我们在文本型数据集中对本文所提出的基于语义的多样性和比例性检索方法进行实验测试，经过测试得到的部分实验结果证明本文所提出的方法有效，但本文的主要的研究主线为在包含空间信息的复合型数据集中进行实验，故在此只对复合型数据集中的实验进行量化分析。除此之外，本文所提出的基于语义或者空间的多样性和比例性检索方法，在复合型数据集中，检索点的信息只为‘位置坐标’信息，因为本文主要的意图是为了，在用户给定很少信息的前提下，能够通过备选检索结果集中的点与检索点之间的关系更好地解释检索点，可以让用户在并不了解此检索点的前提下，能从语义上多类别、空间分布上多方位地了解此检索点。

4.2.1　相关方法在文本关系型数据集中的测试

根据第三章 VR 静态离线权值计算方法计算每个元组的全局权值后，再根据 2.2.1 中计算 OS 中各元组的局部权值，本节主要根据得到的 OS 将元组多样性和等比例特性的系数加入实验中，讲述在引入多样性和等比例特性时，在 DBLP 数据集中如何从 OS 中得到 *se*Dsize-*l* OS 和 *se*Psize-*l* OS，以及抽取实验的部分结果来证明本方法的有效性。

根据 2.2.2 可知，给定一个完整的 OS 和 *l*，在 DBLP 数据集中，如何从 OS 中选择 size-*l* OS 是一个难题，因为 OS 在文本数据集中是一个树形结构，而在空

间数据集中只要指定相应地规则即可。最简单的方法是用动态规划，但是无论从空间还是时间复杂度上讲，代价都是很大的，在此我们采用贪心算法 *k*-LASP。

k-LASP(*k-Largest Averaged Score Path*)即 *k* 个节点的最大平均值路径（也就是一条路径上的 *k* 个节点权值的平均值），本节为了方便表示，将 *dw* 和 *pw* 值统称为权值 *w*，即 OS 中的每一个节点 n_i 都有一个权值 $w(n_i)$，与之对应的 n_i 与其先辈节点（个数 *t*, *t*=*max*(*k*—1, 实际长度)）的平均权值定义为 $AP(n_i)=\frac{\sum_{i=1}^{t} w(n_i)}{t}$。

在生成OS的过程中，需要一个哈希表来记录每个节点的一些信息，用*HFr*表示，*HFr* 包括三个部分，一是哈希表的 key 值即 v_i，二三为其对应的 value 值，二是 v_i（v_i 指的是数据集中的元组节点，假如 v_1 为 “C. Faloutsos”，且在 OS 中 $n_2 = n_4 = v_1$，故 v_1 在 OS 中出现次数为 2 次）在 OS 中出现的次数 $fr(v_i)$，三是 v_i 在 size-*l* OS 中出现的次数 $z(v_i)$。为了更好地管理 OS 中节点和对应的 *AP* 值，建立一个队列 W 来保存这些信息，在这个队列里节点的顺序按相对应的 *AP* 值递减排列。*k*-LASP 算法如下：

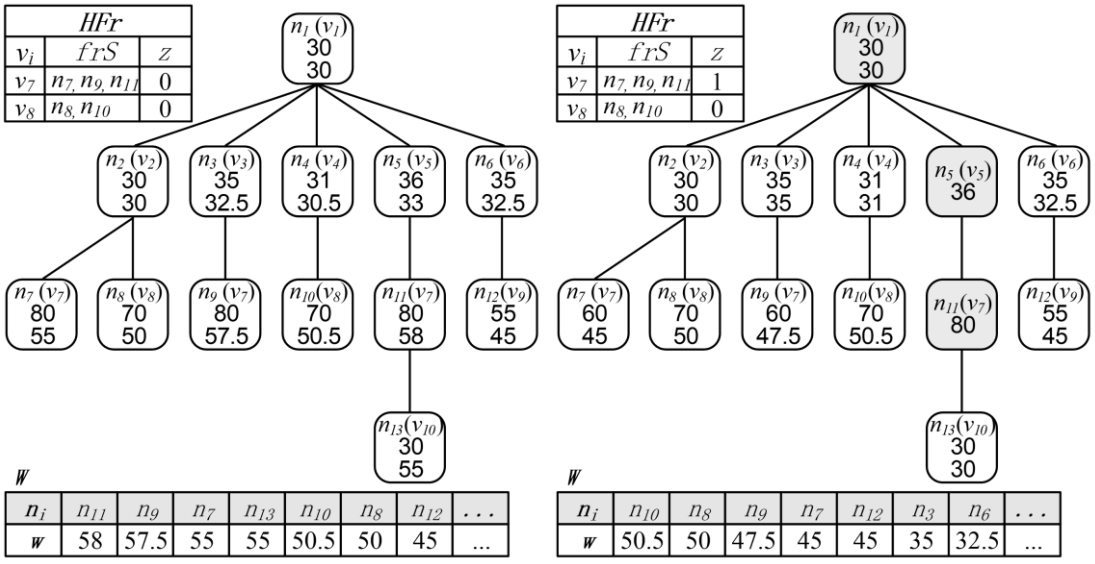
k-LASP 算法

1） 生成 OS，包括构建 *HFr*、计算 $AP(n_i)$和生成 W
2） 若|size-*l*|<*l*，转 3），否则转 11）
3）p_i 表示当前 W 中拥有最大 *AP* 值的节点到根节点的路径，
　将 p_i 里的前 *l*-|size-*l*|个节点加入到 size-*l* OS 中
4）如果|size-*l*|<*l*，转 5），否则转 10）
5）将所选 p_i 中的 *l*-|size-*l*|个节点从 OS 和 W 中移除
6）对于 p_i 每个节点的子孙节点（个数 *t*,*t*=*max*(*k*—1, 实际长度)）n_j 做如下更新：
　在 OS 和 W 中更新 $AP(n_j)$值
7）对于 p_i 中的每个节点 v_i，若 v_i 在 *HFr*，转 8），否则转 10）
8）$HFr(v_i).z$++
9）对于 OS 中每个使得 $n=v_i$ 的节点 *n* 做如下更新：
　对于每一个以节点 *n* 为根的子树中的每个节点 n_i 做：
　　用 $HFr(v_i).z$ 值在 OS 和 W 中更新 $AP(n_i)$值
10）转 2）
11）返回 size-*l* OS

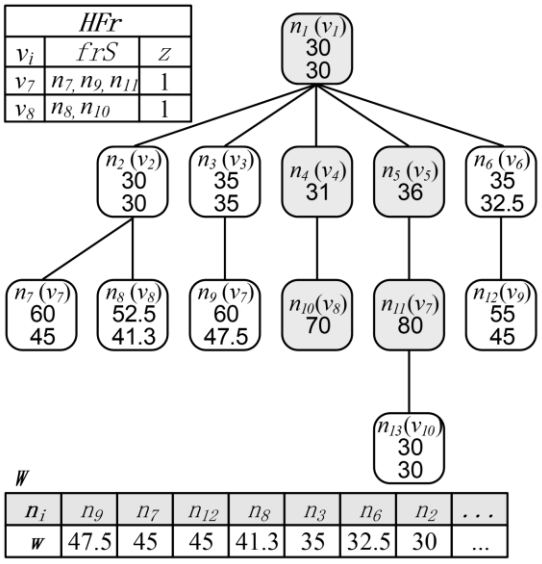
图 4-1 为 *2*-LASP 算法生成 *se*Dsize-*5* OS 的例子，图 4-1（a）为初始的 OS，令输入的 *l*=5。根据 *k*-LASP 算法，令 *k* = 2，首先选择 W 中权值最高的节点即 n_{11}，选择 n_{11} 到根节点这个路径 p_1，路径 p_1 中有 3 个节点，先将这 3 个节点添加到 size-*l* OS 中；根据算法第 4 步，0<5，转到第 5 步，将这三个节点从 OS 和 W 中移除；然后对这三个节点的孩子节点更新 $AP(n_i)$值：

根据算法第 7）行，*HFr* 中有节点 v_i，故转到 8）令 *HFr*(v_i).z++，则 $w(n_9)=w(n_7)=li(g(v_7).dv[2]=80\cdot(1-\frac{2-1}{5-1})=60$，再根据公式 $AP(n_i)=\frac{w(n_{parent})+w(n_i)}{2}$ 来更新 $AP(n_i)$值，如 $AP(n_{10})=\frac{AP(n_4)+AP(n_{10})}{2}=\frac{31+70}{2}=50.5$，其余节点也是按此计算。

到此第一次更新完毕且选择 3 个节点到 *se*Dsize-*l* OS，结果如图 4-1（b）所示，随后再根据上述步骤继续迭代更新权值并选择节点，直到生成最终的 *se*Dsize-*l* OS，最终结果如图 4-1（c）所示。如此便得到了最终的 *se*Dsize-*5* OS。



（a）初始 OS　　（b）初次迭代更新



（c）最终结果（*se*Dsize-*5* OS）

图 4-1 *k*-LASP 算法得到 size-*l* OS 示例（$k = 2$，$l = 5$, 类型为 *se*Dsize-*l* OS）

Figure 4-1 The example of *k*-LASP Algorithm: size-*l* OS ($k = 2$, $l = 5$, type is *se*Dsize-*l* OS)

表 4-1 描述了当 l =10 且 q = “Michalis Faloutsos”时，生成 *se*Dsize-*l* OS 情况下 Author 表中各元组权值的变化情况，主要根据 OS 中 Author 元组的局部权值 *li*，然后根据公式（4-1）、（4-3）的出结果（按 *dw*[1]值递减排列）。

表 4-1 基于作者信息元组的多样性权值变化

Table 4-1 The change of Diversity Weight based on Author’s Tuples

Name	*li*	*dv*[1]	*dw*[1]	*dv*[2]	*dw*[2]	*dv*[3]	*dw*[3]
C.Faloutsos	1.8	1.0	1.8	0.9	1.6	0.78	1.4
S.Madden	1.6	1.0	1.6	0.9	1.4	0.78	1.2
M.Mitzenmacher	1.4	1.0	1.4	0.9	1.2	0.78	1.1
G.Varghese	1.4	1.0	1.4	0.9	1.2	0.78	1.1
J.Cui	0.8	1.0	0.8	0.9	0.7	0.78	0.6
T.Karagiannis	0.7	1.0	0.7	0.9	0.6	0.78	0.5
S.Krishnamurthy	0.6	1.0	0.6	0.9	0.5	0.78	0.5
K.Papagiannaki	0.6	1.0	0.6	0.9	0.5	0.78	0.5
M.Chrobak	0.3	1.0	0.3	0.9	0.3	0.78	0.2
J.Eriksson	0.2	1.0	0.2	0.9	0.1	0.78	0.2

根据此表中各 Author 元组的权值变化可以看到，当 C.Faloutsos 和 M.Mitzenmacher 各将被选中一次的时候，它们的权值分别为 1.8×1 = 1.8 和 1.4×1 = 1.4，但当 C.Faloutsos 将被选择三次时，权值同 M.Mitzenmacher 将被选择一次时的权值是相等，都为 1.4，由此可以看出此方法所得到的结果可以保证输出元组信息的多样性，避免了重复信息多次出现。

还以上述 q 为例，令 l=10，表 4-2 描述了生成*se*Psize-*l* OS 情况下 Author 表中各元组权值的变化情况，主要根据 OS 中 Author 元组的局部权值 *li*，然后根据公式（4-2）、（4-4）得出结果（按 *pw*[1]值递减排列）。

表 4-2 基于作者信息元组的等比例特性权值变化

Table 4-2 The change of Proportional Weight based on Author's Tuples

Name	*li*	*fr*	*pq*[1]	*pw*[1]	*pq*[2]	*pw*[2]	*pq*[3]	*pw*[3]
S.Krishnamurthy	0.6	37	12.3	7.4	7.4	4.4	5.3	3.2
C.Faloutsos	1.8	12	4.0	7.2	2.4	4.3	1.7	3.1
J.Cui	0.8	11	3.7	3.0	2.2	1.8	1.6	1.3
T.Karagiannis	0.7	10	3.3	2.3	2.0	1.4	1.4	1.0
M.Mitzenmacher	1.4	3	1.0	1.4	0.6	0.8	0.4	0.4
G.Varghese	1.4	2	0.7	0.9	0.4	0.6	0.3	0.4
K.Papagiannaki	0.6	4	1.3	0.8	0.8	0.5	0.6	0.4
S.Madden	1.6	1	0.3	0.5	0.2	0.3	0.1	0.2
M.Chrobak	0.3	4	1.3	0.4	0.8	0.2	0.6	0.2
J.Eriksson	0.2	7	2.3	0.4	1.4	0.2	1.0	0.2

根据此表中各 Author 元组的权值变化可以看到，S.Krishnamurthy 和 C.Faloutsos 原来的局部权值（*li*）分别是 0.6 和 1.8，它们的差值为 1.8-0.6 = 1.2，但是 *fr*(S.Krishnamurthy) - *fr*(C.Faloutsos) = 25，根据前面的描述，在比例方面可以认为 S.Krishnamurthy 比 C.Faloutsos 更重要，所以从表中可以看出当其都将被选择到 *se*Psize-*l* OS 三次的时候，其权值差只为 0.1，且 *w*(S.Krishnamurthy) > *w*(C.Faloutsos)。由此可以看出，对于一个可能拥有较弱的局部权值但出现频率较高的元组节点，它的频率表明在最终能得到的 *se*Psize-*l* OS 中此节点与 n^{DS} 存在重要的联系，由此可以保证这个元组节点能够在最终生成的 *se*Psize-*l* OS 中处于一个更合适的位置。

4.2.2　相关方法在复合型数据集中的实验结果及分析

根据上节实验我们在关系型数据集中的部分测试结果来看，本章所提出的方法切实有效，故在本节空间数据集中，对于基于语义多样性的评定方法为用 *se*Dsize-*l* OS 中所包含的 tag 多样性与备选范围内所有 tag 的个数之比，即计算公式为 $\frac{|t_{size-l\,os}|}{|t_{os}|}$ 作为评价检索结果是否考虑语义多样性的评价标准，所有的评价结果用 Approximation Quality（*AQ*）量化表示，即在本实验中 $AQ = \frac{|t_{size-l\,os}|}{|t_{os}|}$，其中 $t_{\text{size-}l\ \text{OS}}$ 为结果中 tag 的总数，同理 t_{OS} 为备选点中 tag 的个数，用此方法来评价检索结果中语义多样性的质量。参照实验为只按照初始节点权值选择的 size-*l* OS 中所包含的 tag 个数与备选范围内所有 tag 的个数之比。具体实验方法如下：

（1） 随机选择 10 个检索点，定义 *l*（为自变量，取值 5，10，15，20），对于每个点做以下（2）-（4）操作：

（2） 定义以检索点经纬度为中心±0.002 经纬度的范围内所有点为 OS（备选节点）

（3） 计算 OS 中 tag 种类总数，考虑 *dv* 计算 *se*Dsize-*l* OS 中 tag 种类总数，得到 *se*Dsize-*l* OS 的质量

（4） 不考虑 *dv* 情况下计算 size-*l* OS 中 tag 种类总数，得到 size-*l* OS 的质量

（5） 对于 *se*Dsize-*l* OS 计算各个检索点得到质量的平均值作为实验的 *AQ*

（6） 对于 size-*l* OS 计算各个检索点得到质量的平均值，即 *AQ*

图 4-2 为 *se*Dsize-*l* OS 和 size-*l* OS 的检索结果质量平均得分折线图，由图中可以看出，*se*Dsize-*l* OS 的 *AQ* 总是比 size-*l* OS 高，而且随着 *l* 的增加 *se*Dsize-*l* OS 的 *AQ* 接近于 1。

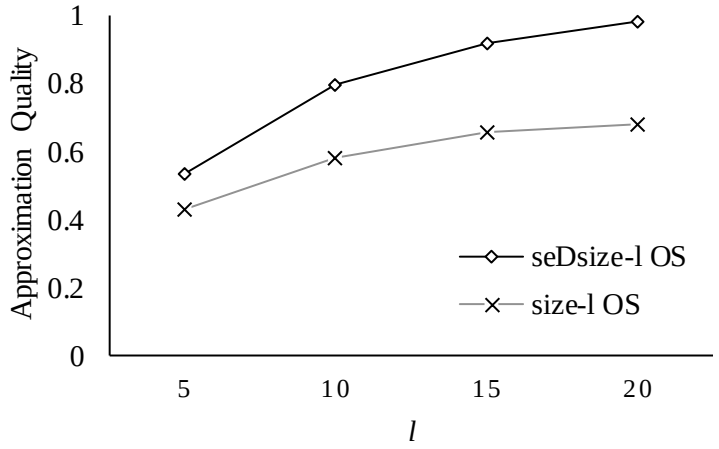


图 4-2 北京 POI 空间数据集中 *se*Dsize-*l* OS 和 size-*l* OS 的 *AQ* 值比较

Figure 4-2 *AQ* of *se*Dsize-*l* OS and size-*l* OS On Beijing POI spatial dataset

这里要注意的是选取以检索点为中心，经纬度±0.002 经纬度的范围内（换算成距离为 2km 左右）所有点作为 OS 的备选节点主要是因为经过观察，在大多数地点，此范围内的 POI 个数为 20~90 左右之间，故在选择结果时更加节省时间，而且 *l* 的选择一般不会超过 20。当然，本实验所随机选择的 10 个点中，最小的|OS|为 7，最大的 88。

在空间数据集中，关于基于语义的等比例特性评定方法为用 *se*Psize-*l* OS 中每类 tag 与 *l* 的比值与 OS 中此类 tag 与|OS|的比值的差绝对值，将每类遍历完后得到的值相加取归一化，作为 *se*Psize-*l* OS 的 *AQ*，即 $\sum 1-|\frac{|t_i|_{size-l\,OS}}{|size-l\,OS|}-\frac{|t_i|_{OS}}{|OS|}|/|t|$，其中$|t_i|_{\text{size-}l\,\text{OS}}$为类别 t_i 在 size-*l* OS 中的个数，$|t_i|_{\text{OS}}$同理，|*t*|是除以|*t*|是为了将此值域控制在[0,1]范围内。参照实验为只按照初始节点权值选择的 size-*l* OS 中每类 tag 与 *l* 的比值与 OS 中此类 tag 与|OS|的比值的差的绝对值，将每类遍历完后得到的值相加取归一化，作为 size-*l* OS 的质量。实验步骤类似于 *se*Dsize-*l* OS 的步骤，也是随机选择十个点，最后计算平均值作为 *AQ*。

图 4-3 为 *se*Psize-*l* OS 和 size-*l* OS 的检索结果质量平均得分折线图，由图中可以看出，*se*Psize-*l* OS 的 *AQ* 总是比 size-*l* OS 高，而且随着 *l* 的增加 *se*Psize-*l* OS 的 *AQ* 接近于 1。

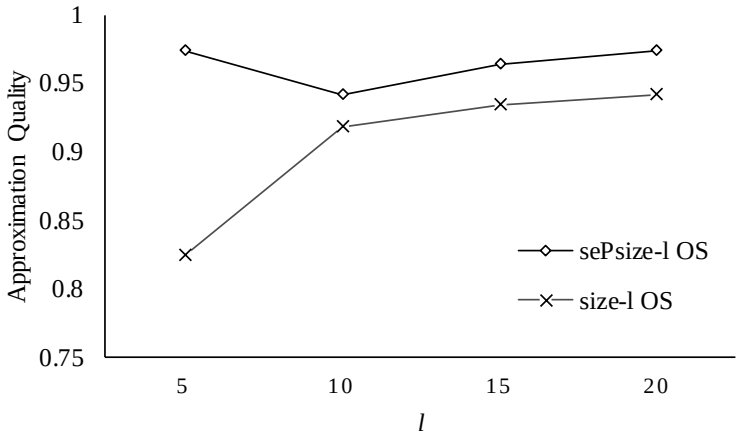


图 4-3 北京 POI 空间数据集中 *se*Psize-*l* OS 和 size-*l* OS 的 *AQ* 值比较

Figure 4-3 *AQ* of *se*Psize-*l* OS and size-*l* OS On Beijing POI spatial dataset

4.3 本章小结

本章主要介绍了基于语义多样性和等比例特性的检索方法，首先介绍了基于语义多样性和等比例特性的意义，主要是为了在检索时，用户表达意图可能不明显，这时，本文希望在结果中返回更全面的信息，本文提出语义多样性系数和等比例系数来削弱经过 ValueRank 算法计算得到的局部权值，使检索结果更有意义。随后分别介绍了语义多样性和等比例特性的计算公式和相关定义。在最后，首先

在 DBLP 数据集中随机选择几个样例测试本章所提出地方法有效，随后在北京 POI 空间数据集中进行实验，得到其实验结果结果并进行分析，在北京 POI 空间数据集中无论是生成 *se*Dsize-*l* OS 还是生成 *se*Psize-*l* OS，判定其检索质量的 *AQ* 值随着 *l* 的增加都接近于 1，这也就证明本文方法的确在基于语义多样性和等比例特性方面的效果是显著的。

第 5 章　基于空间分布的多样性和等比例特性检索方法

本章主要介绍基于空间分布的多样性和等比例特性检索方法，由于本文研究此方法的目的是为了在用户给定一个坐标位置的情况下，尽可能全面地（多样性和等比例特性）解释这个点，即从空间分布来讲为各个方位（多样性）或方向按比（比例特性）返回给用户 *l* 个位置点。

5.1　多样性和等比例特性

现有的空间检索技术，大多都按离检索点的远近来排序，取前 *l* 个点作为结果来返回，显然对于并不了解此地的用户来说这并不合理，在本节就将讨论基于空间分布的多样性和等比例特性检索方法。

5.1.1　空间分布多样性（*sp*Dsize-*l* OS）

空间数据不像文本数据检索那样可以构建树形结构，且按文本相似度去计算多样性，空间数据只有位置和距离信息。在空间中，我们把多样性定义为分布多样性，也就是多个方位，故我们可以根据三角形的特性得到以下公式用于计算空间分布多样性系数：

$$sdv(n_i)=\frac{2\sum_{n_j\in S,n_k\neq n_j}1-\frac{dis(n_k,n_j)}{dis(n_i,n_j)+dis(n_i,n_k)}}{|z|(|z|+1)} \qquad (5\text{-}1)$$

其中，$dis(n_i, n_j)$为点 n_i 到 n_j 的直线距离，z 表示已被选择的位置点集合，n_k 和 n_j 都为已被选择的且不相同的位置点，S 为包含检索点在内的所有被选择到 *sp*Dsize-*l* OS 的所有点的集合，故加上检索点将 S 中个点进行连线可以得到 $\frac{|z|(|z|+1)}{2}$ 条不同的线段，分析其分子，其值域为[0, $\frac{|z|(|z|+1)}{2}$]，故除以分母之后可以将其值域控制在[0,1]之间。这样做的目的一是为了方便计算，二是为了最后生成 *sp*Dsize-*l* OS 时可以方便结合空间分布多样性系数和语义多样性系数（语义多样性系数的值域也为[0,1]）。由于此系数和语义多样性系数都是对用 ValueRank 算法算得的 *li* 进行削弱，故需要控制在同一个值域。与语义多样性权值定义相似，对于 *sp*Dsize-*l* OS 中的每个节点 n_i，提出以下公式来计算结合后的空间分布多样性权值：

$$sdw(n_i) = li(n_i) \cdot sdv(n_i) \quad (5\text{-}2)$$

值得注意的是，在运用公式（5-1）计算时，z 中至少存在两个已经被选择的点，关于前两个点的选择将在第六章给出详细定义，关于此小节的实验及结果详见 5.2 节。

5.1.2 空间分布等比例特性（*sp*Psize-*l* OS）

受基于语义等比例特性的启发，在研究空间分布的等比例特性时，本文以检索点为中心，将空间分为东南（d_1）、东北（d_2）、西南（d_3）、西北（d_4）四类，若为正东、南、西、北方向（即经度或纬度相等）那么随机附属于上述四类。随后根据公式（4-2）得到点 n_i 关于空间分布的等比例特性系数为：

$$spq(n_i) = \frac{fr(d_{n_i})}{\alpha \cdot z(d_{n_i}) + 1} \quad (5\text{-}3)$$

其中 d_{n_i} 为点 n_i 所在空间的类别 $fr(d_{n_i})$ 为 n_i 所在空间类别 d_i 在总的备选点集合中出现的频率而 $z(d_{n_i})$ 为 d_i 在已选结果集的频率。α 为可调整的比例参数，一般为 2。同样，*sp*Psize-*l* OS 中的每个节点 n_i，提出以下公式来计算结合后的空间分布等比例特性权值：

$$spw(n_i) = li(n_i) \cdot spq(n_i) \quad (5\text{-}4)$$

关于此小节的实验及结果详见 5.2 节。

5.2 实验结果与分析

对于空间数据集空间分布多样性的实验，由于没有公式可以对此实验结果量化，故对于空间分布的多样性，本文沿用 ObjectRank 中的调查评分评估方法，即随机给定 10 个点，选择我校 5 名教师和同学（每个参与者在此之前并没有参与本实验且对此多样性计算方法毫不了解），然后分别对相应得到的 *sp*Dsize-*l* OS 和没有考虑空间分布多样性，只是按照初始权值排序得到的 size-*l* OS 效果进行评价打分，分数值域为[1,10]，然后对每个检索点所有评分取平均值，表 5-1 即为评分结果。

表 5-1 *sp*Dsize-*l* OS 效果评估

Table 5-1 Evalustion of *sp*Dsize-*l* OS effectiveness

检索点 id	|OS|	*sp*Dsize-*l* OS 评分	size-*l* OS 评分
112101	288	8.8	5.7
120270	181	8.6	6.0
122385	50	7.0	4.2
452422	115	8.9	6.5
447423	72	8.5	5.2
200440	424	9.5	4.0
222430	13	5.5	5.5
223446	286	8.5	7.2
256989	48	7.6	5.0
302812	103	8.9	6.0
平均值		8.18	5.53

由表 5-1 中可以看出，在考虑空间分布多样性的情况下生成的 *sp*Dsize-*l* OS 效果几乎总是比按权值生成的 size-*l* OS 效果要好很多，而且我们可以看到，对于检索点生成的|OS|越大的位置，生成的 *sp*Dsize-*l* OS 效果越好，相反，效果越差，且和 size-*l* OS 效果接近。图 5-1 为以箭头所指点为检索点生成的 *sp*Dsize-5 OS 和 size-5 OS 在地图中的展示，显然，*sp*Dsize-5 OS 在空间分布多样性方面的效果要比 size-5 OS 好。

(a) *sp*Dsize-5 OS　　(b) size-5 OS

图 5-1 *sp*Dsize-5 OS 和 size-5 OS 例子（箭头所指为检索点）

Figure 5-1 An example of *sp*Dsize-5 OS and size-5 OS (the point of retrieval is that the arrow refers to)

对于空间数据集考虑基于空间分布等比例特性的实验，将以检索点为中心，将空间分为东南（d_1）、东北（d_2）、西南（d_3）、西北（d_4）四类，故在此也参考基于语义等比例特性所用的实验结果评价方法，即用 *sp*Psize-*l* OS 中空间分

布每类 d_i（i∈{1，2，3，4}）中的点的个数，与 l 的比值与 OS 中 d_i 的个数与|OS|的比值的差绝对值，将每类遍历完后得到的值相加取归一化，作为 *sp*Psize-l OS 的 AQ，即 $\sum 1-|\frac{|d_i|_{size-l\ OS}}{|size-l\ OS|}-\frac{|d_i|_{OS}}{|OS|}|/4$，其中$|d_i|_{\text{size-}l\ \text{OS}}$为空间分布类别 d_i 在 size-l OS 中的个数，$|d_i|_{OS}$ 同理，除以 4 是为了将此值域控制在[0,1]范围内。参照实验为只按照初始节点权值选择的 size-l OS 中 d_i 类中点的个数与 l 的比值，与 OS 中 d_i 类中与|OS|的比值的差的绝对值，将每类遍历完后得到的值相加取归一化，作为 size-l OS 的质量。实验步骤类似于 *se*Dsize-l OS 的步骤，也是随机选择十个点，最后计算平均值作为 AQ。

图 5-2 为 *sp*Psize-l OS 和 size-l OS 的检索结果质量平均得分折线图，由图中可以看出，*sp*Psize-l OS 的 AQ 总是比 size-l OS 高，而且随着 l 的增加 *sp*Psize-l OS 的 AQ 接近于 1。

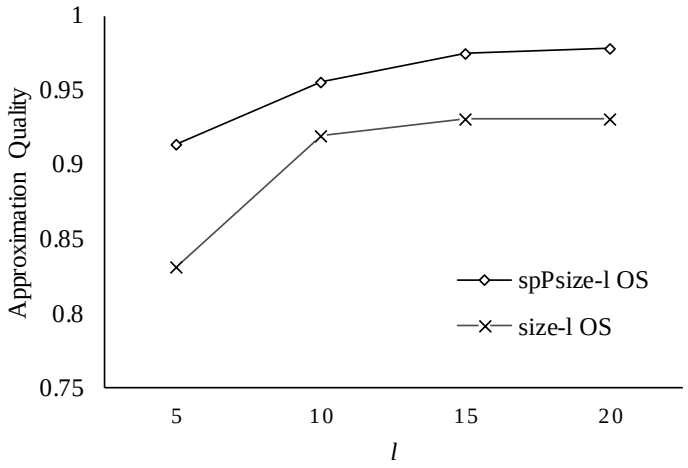


图 5-2 北京 POI 空间数据集中 *sp*Psize-l OS 的 AQ 值比较

Figure 5-2 Approximation quality of *sp*Psize-l OS On Beijing POI spatial dataset

5.5 本章小结

本章主要讲述了基于空间分布的多样性和等比例特性检索方法。基于空间分布的检索和文本不同的是，文本数据集检索，关键词与被检索集合可以根据关系构建成树形结构，而空间数据只能根据空间范围进行检索，所以这也简化了计算过程。在本章中，首先介绍了空间分布的多样性和等比例特性的计算方法和理论

内容，根据空间距离特性给出公式，削弱各点的局部权值。最后给出实验结果，并对结果进行分析，在考虑空间分布的多样性和等比例特性方面，检索结果的质量有较大的提升。

第 6 章　基于语义和空间分布多样性和等比例特性的检索方法

第三章介绍了如何计算待检索数据集中元组的初始权值（全局权值）。第四章主要在全局权值的基础上，在语义的多样性和等比例特性上对其权值进行削弱，在此，分为纯文本数据集和有空间位置信息的复合型数据集两种，在纯文本数据集中，由于文本检索可以构成以检索节点为根，与其有关系的节点为子孙节点的树，故可以根据这些子孙节点与根节点之间的亲密度，可以计算出元组的局部权值，然后根据多样性或等比例特性计算方法来得到相应地 size-*l* OS；而在复合型数据集方面并没有树形结构，故本文将以检索点（位置坐标）为中心，指定距离为半径的范围内的所有点作为检索的候选节点，而它们之间也无亲密度关系，故在复合型数据集中并无局部权值的概念，而是直接选择全局权值作为空间语义结果多样性和等比例特性的初始权值，随后再根据多样性和等比例特性公式来对其进行削弱，得到结果。第五章介绍了复合型数据集中基于空间分布的多样性和等比例特性检索方法。本章主要结合上述三章所有内容，但只在复合型数据集中说明当用户给定一个位置坐标，如何分别得到语义和空间分布多样性和等比例特性的检索结果。

6.1　生成 Dsize-*l* OS 和 Psize-*l* OS 方法介绍

空间检索的多样性计算主要考虑 POI 的语义多样性和空间分布的多样性，而这两方面分别在第四章和第五章有过介绍，故本节将结合第四章基于语义的多样性和第五章空间分布的多样性，对元组的初始权值进行削弱，通过实验分析实验结果，等比例特性亦然。

在此本文只考虑空间数据集，输入为空间坐标点 d（检索点）、检索半径 r（即以检索点 d 为中心，半径为 r 的圆范围内的 POI 为备选点（完整 OS））、检索结果集的规模 l，根据多样性计算，输出为 Dsize-*l* OS。在生成 Dsize-*l* OS 时，空间分布和语义多样性结合后的权值 w 计算公式为：

$$w(n_i) = li(n_i) \cdot (\delta \cdot dv(n_i) + \gamma \cdot sdv(n_i)) \qquad (6\text{-}1)$$

其中，$dv(n_i)$为点 n_i 的语义多样性系数，由公式（4-1）得出，$sdv(n_i)$为点 n_i 的空间分布多样性系数。δ 和 γ 为可调节的系数，且 $\delta + \gamma = 1$，表示削弱权重的占比。

生成 Dsize-*l* OS 的算法如下：

Dsize-*l* OS 生成算法

1) 输入为空间坐标点 *d*，半径 *r*，结果元组个数 *l*≥1；
2) 以 *d* 为圆心，*r* 为半径圆内所有点组成 OS，若|OS|<*l*，则返回|OS|，否则，转 3)；
3) 队列 W 中为按初始权重大小排列的 POI，选择队首 POI 到 Psize-*l* OS，将此点从 W 中移除，并更新 *HFr*（具体看第四章）；
4) 如果|Dsize-*l* OS| = *l*，则转 6)，否则对于所有剩余 OS 点做以下更新：
   根据 *HFr* 中 $z(n_i)$值、公式（4-2）、公式（5-2）和公式（6-2）更新 OS 和 W 中的权值 *w*；
5) 转 2)
6) 返回 Dsize-*l* OS

通过此方法返回的 Dsize-*l* OS 为考虑基于语义多样性和空间分布多样性相结合的 *l* 个 POI 信息。

同理，按照上述类似步骤可以生成 Psize-*l* OS，在生成 Psize-*l* OS 时，空间分布和语义等比例特性结合后的权值 *w* 计算公式为：

$$w(n_i) = li(n_i) \cdot (\delta \cdot pq(n_i) + \gamma \cdot spq(n_i)) \qquad (6\text{-}2)$$

其中，$pq(n_i)$为点 n_i 的语义多样性系数，由公式（5-1）得出，$sdv(n_i)$为点 n_i 的空间分布多样性系数。δ 和 γ 为可调节的系数，且 $\delta + \gamma = 1$，表示削弱权重的占比。

生成 Psize-*l* OS 的算法如下：

Psize-*l* OS 生成算法

1) 输入为空间坐标点 *d*，半径 *r*，结果元组个数 *l*≥1；
2) 以 *d* 为圆心，*r* 为半径圆内所有点组成 OS，若|OS|<*l*，则返回|OS|，否则，转 3)；
3) 队列 W 中为按初始权重大小排列的 POI，选择队首 POI 到 Psize-*l* OS，将此点从 W 中移除，并更新 *HFr*（具体看第四章）；
4) 关于第二个 POI 的选择，更新 OS 中剩余的所有节点的权值，计算各个 POI 的语义多样性，按公式（4-3）得到的权值来更新 W，此时选择队首 POI 到 Psize-*l* OS，将此点从 W 中移除，并更新 *HFr*；
5) 如果|Psize-*l* OS| = *l*，则转 7)，否则对于所有剩余 OS 点做以下更新：
   根据 *HFr* 中 $z(n_i)$值、公式（4-1）、公式（5-1）和公式（6-1）更新 OS 和 W 中的权值 *w*；
6) 转 2)
7) 返回 Psize-*l* OS

通过此方法返回的 Psize-*l* OS 为考虑语义等比例特性和空间分布等比例特性性相结合的 *l* 个 POI 信息。

综上，介绍了 Dsize-*l* 和 Psize-*l* 的生成方法。在此要注意的是生成 Dsize-*l* OS 时，*sdv* 的计算的前提为 $l > 2$，故，只有当 $l > 2$ 时才能用到公式（5-1）。下节主要从北京 POI 数据集中验证本文提出的方法的有效性，根据实验结果进行分析。

6.2　实验结果与分析

本章还是沿用调查评估评分方法对实验进行评价。

空间数据集生成 Dsize-*l* OS 部分检索结果如表 6-1 所示，其中|OS|为以此检索点半径 2 公里内所有点的个数，$|t|_{OS}$ 为 OS 中 tag 类别的数量，同理$|t|_{Dsize\text{-}l\ OS}$为结果中 tag 类别的数量，$|t|_1$、avg_w_1 和 S_1 为公式（6-2）中令 $\delta = 0.1$，$\gamma = 0.9$ 所得到结果的 tag 类别总数、平均权值（即结果中每个点的权值总和的平均数）和所有参与者打分的平均值；$|t|_2$、avg_w_2 和 S_2 为公式（6-2）中令 $\delta = 0.5$，$\gamma = 0.5$ 所得到结果的 tag 类别总数、平均权值和所有参与者打分的平均值；$|t|_3$、avg_w_3 和 S_3 为公式（6-2）中令 $\delta = 0.9$，$\gamma = 0.1$ 所得到结果的 tag 类别总数、平均权值和所有参与者打分的平均值。表中最后一行为打分类分数所取得平均值（四舍五入）后的结果。

表 6-1 Dsize-*l* OS 实验结果（$l = 10$）

Table 6-1 The result of Dsize-*l* OS （$l = 10$）

检索点	|OS|	$\|t\|_{OS}$	$\|t\|_1$	avg_w_1	S_1	$\|t\|_2$	avg_w_2	S_2	$\|t\|_3$	avg_w_3	S_3
172002	6	5	5	1.525	8.3	5	1.560	8.6	5	1.595	8.3
184672	99	15	5	10.419	7.6	4	10.889	7.3	4	11.382	8.2
208492	87	11	4	39.410	7.9	4	39.730	8.3	4	40.231	8.1
238475	16	6	5	1.795	8.6	4	1.990	8.5	4	2.165	8.0
283759	10	3	3	1.783	9.2	3	1.883	8.8	3	1.980	9.5
316493	94	12	4	3.306	7.3	3	3.448	7.5	4	3.938	7.7
358294	25	12	6	2.547	7.9	5	2.667	8.1	4	2.768	8.5
376002	44	8	5	1.959	9.0	4	2.067	9.2	4	2.181	7.9
420356	84	12	5	31.517	8.2	4	31.706	8.4	4	31.872	8.1
450294	86	14	3	5.864	8.8	3	6.101	8.5	3	6.287	8.5
平均值					8.3			8.3			8.3

从表中可以看出参与者对参数调整的满意度几乎是相同的（平均值都为 8.3），从表中数据来看总是 $avg\_w_3 > avg\_w_2 > avg\_w_1$，由此可见，空间分布多样性对节点的权值削弱相对较大。根据用户的评分，本文提出的 Dsize-*l* OS 计算方法使用户在不了解检索点的前提下，返回给用户多样信息去诠释检索点有很大的帮助。图 6-1 为当 $\delta = 0.5$，$\gamma = 0.5$ 时，以箭头所指位置为检索点（lng = 116.48195 ，lat = 39.908966）所得到的 Dsize-*l* OS （$l = 10$）结果，其中$|t|_{OS}$ =12，$|t|_2$ = 4。

图 6-1 Dsize-10 OS 例子（Q 为检索点）

Figure 6-1 An example of Dsize-10 OS (the point of ‘Q’ is that the arrow refers to)

表 6-2 为上述生成 Dsize-10 OS 结果中各个点的详细信息，其中 id 为图 6-1 中红色标注的编号，且为选择到 Dsize-10 OS 的顺序序号。根据图 6-1 展示和表 6-2 中的数据，与图 6-2 和表 6-4 的结果可以看出此实验得到的结果在语义和空间分布多样性上具有良好的表现，说明本文提出的生成 Dsize-*l* OS 的方法具有可行性。

表 6-2 图 6-1 中各 POI 详细信息

Table 6-2 The detail information of POIs in Figure 6-1

id	lat	lng	tag	name
检索点	39.908966	116.48195	-	-
1	39.910824	116.48265	交通设施	收费停车场
2	39.907692	116.480314	购物	依客便利店
3	39.907279	116.483212	购物	雪花花艺生活馆
4	39.907021	116.482727	购物	CHI ZHANG
5	39.908835	116.481066	交通设施	北京国家广告产业园区-地下停车场
6	39.908396	116.480833	交通设施	通惠国际传媒广场-停车场
7	39.909811	116.4838	美食	通惠小镇
8	39.908737	116.482422	美食	老北京家常菜
9	39.909465	116.483366	美食	依客家
10	39.909475	116.480174	出入口	北京国家广告产业园区 B 座-北二门

同理，空间数据集生成 Psize-l OS 部分检索结果如表 6-3 所示，其中|OS|为以此检索点半径 2 公里所有点的个数，AQ_t 为根据第四章评价语义等比例特性质量公式$\sum 1-|\frac{|t_i|_{size-l\,OS}}{|size-l\,OS|}-\frac{|t_i|_{OS}}{|OS|}|/|t|$得到的语义质量值，同理 AQ_d 为第五章评价空间分布等比例特性质量公式$\sum 1-|\frac{|d_i|_{size-l\,OS}}{|size-l\,OS|}-\frac{|d_i|_{OS}}{|OS|}|/4$得到的空间分布质量值。

AQ_{t1}、AQ_{d1}、avg_w_1 和 S_1 为公式（6-3）中令 $\delta=0.1$， $\gamma=0.9$ 所得到结果语义等比例特性质量值、空间分布等比例特性质量值、平均权值（即结果中每个点的权值总和的平均数）和所有参与者打分的平均值；AQ_{t2}、AQ_{d2}、avg_w_2 和 S_2 为公式（6-3）中令 $\delta=0.5$， $\gamma=0.5$ 所得到结果语义等比例特性质量值、空间分布等比例特性质量值、平均权值（即结果中每个点的权值总和的平均数）和所有参与者打分的平均值；AQ_{t3}、AQ_{d3}，avg_w_3 和 S_3 为公式（6-3）中令 $\delta=0.9$， $\gamma=0.1$ 所得到结果语义等比例特性质量值、空间分布等比例特性质量值、平均权值（即结果中每个点的权值总和的平均数）和所有参与者打分的平均值。表中最后一行为打分类分数所取得平均值（四舍五入）后的结果。

表 6-3 Psize-*l* OS 实验结果（$l = 10$）

Table 6-3 The result of Psize-*l* OS （$l = 10$）

检索点	|OS|	AQ_{t1}	AQ_{d1}	avg_w_1	S_1	AQ_{t2}	AQ_{d2}	avg_w_2	S_2	AQ_{t3}	AQ_{d3}	avg_w_3	S_3
172002	6	1.0	1.0	0.09	9.0	1.0	1.0	0.086	9.2	1.0	1.0	0.08	9.0
184672	99	0.30	0.98	7.04	7.4	0.36	0.98	4.63	7.8	0.39	0.98	2.26	8.2
208492	87	0.42	0.98	11.74	8.2	0.41	0.98	9.04	7.3	0.56	0.98	6.41	8.8
238475	16	0.65	0.94	0.21	8.8	0.65	0.94	0.19	8.5	0.65	0.94	0.17	8.5
283759	10	1.0	1.0	0.13	9.2	1.0	1.0	0.14	9.2	1.0	1.0	0.16	9.2
316493	94	0.31	0.93	1.35	8.3	0.33	0.93	1.31	8.4	0.48	0.93	1.31	8.5
358294	25	0.31	0.72	0.30	7.2	0.30	0.72	0.28	7.5	0.39	0.72	0.26	7.8
376002	44	0.21	0.71	0.43	8.5	0.34	0.71	0.49	6.3	0.43	0.71	0.54	7.2
420356	84	0.47	0.97	4.56	8.5	0.47	0.97	4.27	8.6	0.56	0.94	4.00	9.0
450294	86	0.19	0.95	1.55	8.9	0.26	0.95	1.38	8.2	0.54	0.92	1.27	9.2
平均值		0.48	0.92	2.74	8.4	0.51	0.92	2.18	8.1	0.60	0.91	1.65	8.5

从结果来看，由于空间分布等比例特性的类别只有四类，而结果语义等比例特性的类别平均比较多，故 AQ_d 分值都相对与 AQ_t 的要高，且对于参数的调整，AQ_d 的变化并不大，而 AQ_t 的变化较大，这也和结果类别的数量有关。对于平均权值 avg_w，整体上 $avg\_w_1 > avg\_w_2 > avg\_w_3$，可见结果语义等比例特性对节点权值削弱较大。从用户的评分来看，本文提出的 Psize-*l* OS 的生成方法是有效的。图 6-4 为当 $\delta = 0.5$，$\gamma = 0.5$ 时，以箭头所指位置为检索点（lng = 116.48195 ，lat = 39.908966）所得到的 Psize-*l* OS （$l = 10$）结果，其中|OS| =90， $|d_{3OS}| = 61$，$|d_{4OS}| = 15$， $|t_{公司企业OS}| = 52$。

图 6-2 Psize-10 OS 例子（Q 所指为检索点）

Figure 6-2 An example of Psize-10 OS (the point of 'Q' is that the arrow refers to)

表 6-4 为上述生成 Psize-10 OS 结果中各个点的详细信息，其中 id 为图 6-2 中红色标注的编号，且为选择到 Psize-10 OS 的顺序序号。以 tag_d = d_3 为例，$||d_{3OS}|/|OS|-|d_{3size\text{-}10\ os}|/|size\text{-}10\ OS||=|61/90 – 8/10| = 0.12$。相对应于图 6-1 的结果来看，明显在结果的空间分布等比例特性上有所体现，这也说明本文提出生成 Psize-l OS 方法的可行性。

表 6-4 图 6-2 中各 POI 的详细信息

Table 6-4 The detail information of POIs in Figure 6-2

id	lat	lng	tag_t	tag_d	name
检索点	39.908966	116.48195	-		-
1	39.91082	116.4827	交通设施	d_2	收费停车场
2	39.90841	116.4805	公司企业	d_3	北京明德天行管理咨询有限责任公
3	39.90874	116.4811	公司企业	d_3	北京第四极科技有限公司
4	39.90805	116.4805	公司企业	d_3	北京优美达电梯销售有限公司
5	39.90805	116.4805	公司企业	d_3	华诚万家科技(北京)有限公司
6	39.90837	116.4804	公司企业	d_3	F 团(Ftuan)北京分公司
7	39.90874	116.4824	美食	d_4	老北京家常菜
8	39.90846	116.4809	房地产	d_3	通惠国际传媒广场
9	39.90811	116.48	文化传媒	d_3	北京北视英特维文化传播有限公司
10	39.90894	116.4804	公司企业	d_3	乐语

生成 Psize-l OS 和 Dsize-l OS 的时间复杂度最差的情况下都为 $O(l(n+nlog_2n))$，其中 l 为结果集中元组的个数，n 为候选结果集中元组的个数，即每次在选择结果集元组的时需要对候选结果集中所有元组进行更新其时间复杂度为 $O(n)$，排序算法时间复杂度为 $O(nlog_2n)$，故最差的情况下需要选择 l 次，故时间复杂度为 $O(l(n+nlog_2n))$。

6.3 本章小结

本章主要根据四、五章提出的方法，将空间分布与语义相结合，生成基于语义和空间分布多样性的 Dsize-l OS 和生成基于语义和空间分布等比例特性的 Psize-l OS，结果表明此方法生成的检索结果对于参与者来说是比较满意的（平均打分 8 分以上），而且例子证明本文提出的方法是有效的。

结 论

目前，很多的空间检索技术，大多数都是根据用户所在根据位置和给定的关键词按照权值从高到低排列的，在权值方面大多数都是按照线上应用的用户评分作为检索权值，而忽略各属性本身存在的关系，而在结果方面一是用户必须给定关键词才能得到的结果，二是所得到的结果可能在某种语义或空间上聚集严重。所以在用户在不熟悉的位置下，这些技术都无法了解用户的隐含意图，也就不能在语义和空间上给用户返回多样化信息。本文针对目前的空间位置检索技术，做了以下三方面研究工作：

第一，提出一种新的离线权重计算方法—ValueRank，来计算各词条的初始权值，它是 PageRank 和 ObjectRank 的扩展，它在旧的方法的基础上引入了动态“数值”的概念，即不仅仅是依靠数量来评定词条的全局权值。在实验方面主要运用两种数据集来进行实验，一种是纯文本数据集，像 DBLP 和 Northwind，根据结果来看，通过 ValueRank 计算得到的全局权值比 ObjectRank 的更加合理；另外一种是空间数据集，即本文最终结果所用的北京 POI 空间数据集，在进行此项实验时，为了构建其数据模式图，本文运用基于 R 树的方法进行商圈的划分，与以往人工划分商圈不同的是，此方法更有数据依据，随后通过实验证明，用 ValueRank 计算得到的各词条的初始权值比只按用户评价作为初始权值更为合理。

第二，提出基于语义多样性和等比例特性的检索方法。在检索结果中本文认为不能只按初始权值排序作为唯一的标准返回结果，这样在语义方面可能会导致在某一语义上产生严重的聚集，而我们在不了解用户意图时，返回这样的结果明显是不合理的，故提出此计算方法来返回多样化的结果。首先在多样性方面，本文也是根据两类（纯文本和空间）数据集进行实验，本文认为在不了解用户意图的情况下，应尽可能地返回语义多样性地结果，在纯文本数据集中，由于结果候选集是以关键词为根的树形结构，故还提出结合多样性或等比例特性计算的 *k*-LASP 结果集选择方法，此方法为贪心算法，在效率上比暴力方法要快得多，由于纯文本数据集的结果候选集为树形结构且节点较多，无法量化计算结果，但从实验给出的例子来看，本文提出的方法是有效的；在空间数据集中，本文只对其 tag 作为语义多样性的标准，由于空间数据集中备选结果集中的点的个数较少，且 tag 数量较少，容易记录，故可以用公式经过量化计算，实验结果表明随着 *l* 的增大语义多样性结果的质量越好且趋近于 1，而只按权值排序得到的 size-*l* OS 结果多样性只能趋近于 0.6。从等比例特性方面，本文认为频繁出现的词条对于结果来说可能比较重要，于是提出了结果等比例特性计算方法，在纯文本数据集

中通过实验结果例子表明本文提出等比例特性计算方法的有效性，在空间数据集中，结果表明随着 *l* 的增大语义等比例特性结果的质量越好且趋近于 1，而只按权值排序得到的 size-*l* OS 结果等比例特性总比上述结果要差。

第三，提出基于空间分布多样性和等比例特性的检索方法。在空间检索中大多数只按照距离远近将候选集的点排序，与语义结果类似，传统方法可能造成结果在某一空间分布上的聚集，故提出基于空间分布多样性和等比例特性的检索方法，计算方法与语义类似，在空间分布多样性方面对实验结果评价采用调查评分评估方法，最终在考虑空间分布多样性生成的 *sp*Dsize-*l* OS 平均评分为 8.18，而只按权值排序得到的 size-*l* OS 结果为 5.53；在空间分布等比例特性方面用量化计算法得到的 *sp*Psize-*l* OS 的质量总比 size-*l* OS 的高。

最后结合空间分布和语义的多样性和等比例特性，分别生成 Dsize-*l* OS 和 Psize-*l* OS，通过调节控制空间分布和语义的参数进行实验，实验还是采用调查评分评估方法，最终结果都为 8 分以上，故可以得出本文所提出的计算方法的有效性。

本文不仅在语义和空间分布多样性和等比例特性上提出新的计算方法，而且在计算词条权值方面也提出新的计算方法，避免了人工标注的复杂性，在语义和空间分布多样性和等比例计算方面也存在些许不足：

第一，在空间数据集上没有将点按意图进行分类，例如一个想来旅游的用户可能并不得到附近的公司信息；

第二，在计算空间分布多样性上，只考虑了四个方位，这导致在最后生成 Psize-*l* OS 时空间分布多样性值较语义多样性大，对结果得影响也相对较大；

第三，最后在生成 Dsize-*l* OS 和 Psize-*l* OS 上，没有好的量化计算方法去评价实验结果。

针对上述三个方面得不足，我们将在今后的工作和学习中进行进一步研究，针对第一点，将问卷调查用户意图，将数据集细分为几类，使结果能够按照用户意图展现；针对第二、三点希望在后期得研究和调查中有好的新的方法可以运用，使本文得结果更有说服性。

参 考 文 献

[1] Aditya B, Bhalotia G, Chakrabarti S, et al. BANKS: browsing and keyword searching in relational databases[C]// Proceedings of the 28th international conference on Very Large Data Bases. VLDB Endowment, 2002:1083-1086.

[2] Hulgeri A, Nakhe C. Keyword Searching and Browsing in Databases using BANKS[C]// International Conference on Data Engineering. 2002:431-440.

[3] Hristidis V, Papakonstantinou Y. DISCOVER: Keyword Search in Relational Databases[J]. Vldb, 2003, 26(2):670-681.

[4] Ying Z A, University E C N, Shanghai. Location-Based Services: Architecture and Progress[J]. Chinese Journal of Computers, 2011, 34(7):1155-1171.

[5] Cong G, Jensen C S, Wu D. Efficient retrieval of the top-k most relevant spatial web objects[J]. Proceedings of the Vldb Endowment, 2009, 2(1):337-348.

[6] Yao B, Li F, Hadjieleftheriou M, et al. Approximate string search in spatial databases[J]. 2010, 41(3):545-556.

[7] Cao X, Cong G, Jensen C S, et al. Collective spatial keyword querying[C]// ACM SIGMOD International Conference on Management of Data, SIGMOD 2011, Athens, Greece, June. 2011:373-384.

[8] Li G, Feng J, Xu J. DESKS: Direction-Aware Spatial Keyword Search[C]// IEEE, International Conference on Data Engineering. IEEE Computer Society, 2012:474-485.

[9] Basu Roy S, Chakrabarti K. Location-aware type ahead search on spatial databases: semantics and efficiency[C]// ACM SIGMOD International Conference on Management of Data, SIGMOD 2011, Athens, Greece, June. 2011:361-372.

[10] 邬艳艳，周新科. 信息检索结果隐式多样化排序方法研究[J]. 电子科技，2016, 29(8):106-109.

[11] 王莹,罗准辰,于洋.基于排序学习模型的微博多样性检索问题研究[J].计算机工程,2017,43(11):152-160.

[12] 刘兴林. 信息检索多样化排序算法研究综述[J]. 中国科技信息, 2014(16):33-35.

[13] Brin S, Page L. The anatomy of a large-scale hypertextual Web search engine[C]// International Conference on World Wide Web. Elsevier Science Publishers B. V. 1998:107-117.

[14] Balmin A, Hristidis V, Papakonstantinou Y. Objectrank: authority-based keyword search in databases[C]// Thirtieth International Conference on Very Large Data Bases. VLDB Endowment, 2004:564-575.

[15] SANTOS RLT， MACD Santos RLT, Macdonald C, Ounis I. Exploiting query reformulations for web search result diversification[C]// International Conference on World Wide Web. ACM, 2010:881-890.

[16] J. G. Carbonell and J. Goldstein. The use of mmr,diversity-based reranking for reordering

documents andproducing summaries. In SIGIR, pages 335－336, 1998.

[17] S. Gollapudi and A. Sharma. An axiomatic approach for result diversification. In WWW, pages 381－390, 2009

[18] H. L. Vieira, M. R. amd Razente, M. C. N. Barioni,M. Hadjieleftheriou, D. Srivastava, A. J. M. Traina, and V. J. Tsotras. On query result diversification. In ICDE, pages 1163－1174, 2011.

[19] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In WSDM, pages 5－14, 2009.

[20] A. Angel and N. Koudas. Efficient diversity-aware search. In SIGMOD, pages 781－792, 2011.

[21] M. Drosou and E. Pitoura. Disc diversity: result diversification based on dissimilarity and coverage. PVLDB, 6(1):13－24, 2012.

[22] M. Drosou and E. Pitoura. The disc diversity model. In EDBT/ICDT Workshops, pages 173－175, 2014.

[23] V. Dang and W. Croft. Diversity by proportionality: an election-based approach to search result diversification. In SIGIR, 2012

[24] L. Wu, Y. Wang, J. Shepherd, and X. Zhao. An optimization method for proportionally diversifying search results. Advances in Knowledge Discovery and Data Mining,70(2):390－401, 2013.

[25] Zhou YH, Xie X, Wang C, GongYC, Ma WY. Hybrid index structures for location-based Web search. In: Proc. of the CIKM. NewYork: ACM Press, 2005. 155 162. [doi: 10.1145/1099554.1099584]

[26] 空间关键词搜索研究综述*刘喜平 1,2, 万常选 1,2, 刘德喜 1,2, 廖国琼 1,2

[27] Chen YY, Suel T, Markowetz A. Efficient query processing in geographic Web search engines. In: Proc. of the ACM SIGMOD.New York: ACM Press, 2006. 277 288. [doi: 10.1145/1142473.1142505]

[28] Christoforaki M, He J, Dimopoulos C, Markowetz A, Suel T. Text vs. space: Efficient geo-search query processing. In: Proc. of the CIKM. New York: ACM Press, 2011. 423 432. [doi: 10.1145/2063576.2063641]

[29] Li ZS, Lee KCK, Zheng BH, Lee WC, Lee DL, Wang XF. IR-Tree: An efficient index for geographic document search. IEEE Trans.on Knowledge and Data Engineering, 2011,23(4):585 599. [doi: 10.1109/TKDE.2010.149]

[30] 胡骏, 范举, 李国良,等. 空间数据上 Top-k 关键词模糊查询算法[J]. 计算机学报, 2012, 35(11):002237-2246.

[31] Zhang CY, Zhang Y, Zhang WJ, Lin XM. Inverted linear quadtree: Efficient top K spatial keyword search. In: Proc. of the ICDE. Washington: IEEE, 2013. 901 912. [doi: 10.1109/ICDE.2013.6544884]

[32] Cary A, Wolfson O, Rishe N. Efficient and scalable method for processing top-k spatial

Boolean queries. In: Proc. of the SSDBM. LNCS 6187, Berlin, Heidelberg: Springer-Verlag, 2010. 87 95. [doi: 10.1007/978-3-642-13818-8_8]

[33] Felipe ID, Hristidis V, Rishe N. Keyword search on spatial databases. In: Proc. of the ICDE. Washington: IEEE, 2008. 656 665. [doi: 10.1109/ICDE.2008.4497474]

[34] Li GL, Feng JH, Xu J. DESKS: Direction-aware spatial keyword search. In: Proc. of the ICDE. Washington: IEEE, 2012. 474 485. [doi: 10.1109/ICDE.2012.93]

[35] Wu DM, Yiu ML, Cong G, Jensen CS. Joint top-K spatial keyword query processing. IEEE Trans. on Knowledge and Data Engineering, 2012,24(10):1889 1903. [doi: 10.1109/TKDE.2011.172]

[36] Cong G, Jensen CS, Wu DM. Efficient retrieval of the top-k most relevant spatial Web objects. Proc. of the VLDB Endowment, 2009, 2(1):337 348. [doi: 10.14778/1687627.1687666]

[37] Huang WH, Li GL, Tan KL, Feng JH. Efficient safe-region construction for moving top-K spatial keyword queries. In: Proc. of the CIKM. New York: ACM Press, 2012. 932 941. [doi: 10.1145/2396761.2396879]

[38] Rocha-Junior JB, Gkorgkas O, Jonassen S, Nørvåg K. Efficient processing of top-k spatial keyword queries. In: Proc. of the 12th Int’l Conf. on Advances in Spatial and Temporal Databases. Berlin, Heidelberg: Springer-Verlag, 2011. 205 222. [doi: 10.1007/978-3-642-22922-0_13]

[39] Rocha-Junior JB, Nørvåg K. Top-k spatial keyword queries on road networks. In: Proc. of the EDBT. New York: ACM Press, 2012. 168 179. [doi: 10.1145/2247596.2247617]

[40] Wu DM, Yiu ML, Jensen CS, Cong G. Efficient continuously moving top-k spatial keyword query processing. In: Proc. of the ICDE. Washington: IEEE, 2011. 541 552. [doi: 10.1109/ICDE.2011.5767861]

[41] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In WWW Conference, pages 107117, 1998.

[42] T. H. Haveliwala. Topic-sensitive pagerank. In WWW '02, pages 517526, 2002.

[43] R. Varadarajan, V. Hristidis, and L. Raschid. Explaining and reformulating authority ow queries. ICDE, pages 883892, 2008.

[44] G. J. Fakas. A novel keyword search paradigm in relational databases: Object summaries. Data Knowl. Eng., 70(2):208 – 229, 2011.

[45] Fakas, G. J. (2008). Automated generation of object summaries from relational databases: A novel keyword searching paradigm. IEEE, International Conference on Data Engineering Workshop, IEEE Computer Society, pp.564-567.

[46] Fakas, G. J., Cai, Z., & Mamoulis, N. (2014). Versatile size-l object summaries for relational keyword search. IEEE Transactions on Knowledge & Data Engineering, Vol.26 No.4, 1026-1038.

[47] Balmin A, Hristidis V, Papakonstantinou Y. ObjectRank 1 : Authority-Based Keyword

Search in Databases[C]// Thirtieth International Conference on Very Large Data Bases. ELSEVIER, 2004:564–575.

攻读硕士学位期间所发表的学术论文

1. 才智，兰许，曹阳. 一种基于多样性和比例特性的关键词检索方法，国家发明专利：201610218405.1。
2. 才智，李彤，兰许，曹阳，丁治明. 基于多样性的地理空间兴趣点检索方法，国家发明专利：201611254804.X
3. Zhi Cai，Xu Lan，Yang Cao. ValueRank Keyword Search of Object Summaries Considering Values. KSII Transactions on Internet and Information Systems（已投）

致 谢

三年光阴，白驹过隙。毫不留意，就要匆匆的写下句号。回顾研究生的学习生活，内心感慨万千。在此对之前在学习和生活中给予我帮助的人们表示最诚挚的谢意。

首先由衷的感谢才智老师在研究生期间对我生活上的照顾和学业上的指导。才老师有着严谨的学术态度和尽职尽责的工作精神，他将学术和学习生活融为一体，提醒我们注意思维的锻炼和看待问题解决问题的方式。才老师的谆谆教诲不仅指导了研究生三年，我将在以后都谨记心中。

同时还要感谢实验室的同学！感谢实验室的师兄师姐们对我的帮助，还有刘超、穆红章、方皓达等同学在学业与生活上的关怀与帮助。

感谢我的朋友们，是缘分让我们相聚在这美丽的北京工业大学，和你们共同度过的欢乐时光会永远地保存在我的校园记忆中！

还要要特别感谢挚爱的父母，在父母的鼓励和陪伴下我走过这十多年的求学之路。感谢我的家人们一直以来的支持与信任让我对学业的追求不曾犹豫！

最后，感谢评阅、评议论文和答辩委员会的各位老师在百忙之中给予我的指导。

岁月悠悠，惟愿天厚泽人！