中文地址名称
识别算法设计和实现

Chinese address name recognition algorithm design and implementation

领　　域：计算机技术

作者姓名：梁东阳

指导教师：宫秀军　副教授

企业导师：张丁浩 高级工程师

天津大学计算机科学与技术学院

二零一五年五月

独创性声明

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和取得的研究成果，除了文中特别加以标注和致谢之处外，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 **天津大学** 或其他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。

学位论文作者签名：　　签字日期：　　年　　月　　日

学位论文版权使用授权书

本学位论文作者完全了解 **天津大学** 有关保留、使用学位论文的规定。特授权 **天津大学** 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，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。同意学校向国家有关部门或机构送交论文的复印件和磁盘。

（保密的学位论文在解密后适用本授权说明）

学位论文作者签名：　　导师签名：

签字日期：　　年　　月　　日　　签字日期：　　年　　月　　日

摘　要

中文地址名称识别是将用户输入的地址信息转化到地理坐标的过程，是现代汽车导航软件、地理信息和基于位置的服务等系统中最重要的功能之一。百度地图和高德地图等国内厂商大多采用“正向最大匹配分词”或“逆向最大匹配分词”的地址分词算法，当用户的输入在地址库中存在时也能够搜索到正确值。 但当用户输入稍有错误就不能得到满意的结果或不能得到结果。本文通过用户的输入并结合地址词典，利用自然语言处理等技术，合理推断用户最大意图，从而大大增加地址匹配的鲁棒性和智能性。

从原始地图供应商拿到的地址库是固定唯一的，我们首先要保证当用户输入在该原始字典中的情况下，能够正确匹配到地址。我采用了利用原始地址库建立Trie树的数据结构，分词采用动态规划的方法。这样做能够使分词效率和准确率大大增加。

当用户输入的地址在地址库中不存在的情况下，上面方法就会失败。为了解决这个问题，本文对地址词典中的元素（即地址词）中的每一个字建立位置标注（B-开始，E-结束，M-中间，S-单个词），建立隐马尔科夫模型，训练该模型参数。考虑到性能，识别过程采用基于动态规划的维特比算法，这样进一步增大了分词的范围。

当对地址分词后，需要确定单词的属性，也就是该单词属于省、市、县、街道名或兴趣点等，同时也需要满足对用户的输入能够在地址库中校正。本文建立了基于n-gram的倒排索引文件，该文件能够快速索引到与某一单词相关的单词集。之后通过计算相似度并排序，得到对应匹配地址的最大结果。

智能地址匹配最主要的部分就是分词和匹配，如何能够最大程度处理好这两个步骤是本文的关键点。本文是把成熟的自然语言处理技术引入到地理编码领域的一次尝试，实验证明该方法具有一定的可靠性。

关键词：Trie树，隐马尔可夫模型，动态规划，n-gram，维特比算法

THESIS

Chinese address name recognition transforms the user’s information into geographic coordinates and is one of the most important functions in modern auto navigation software, geographic information and location based services system. Baidu map, Gaode map and other domestic companies mostly use "forward maximum matching" or "reverse maximum matching" word segmentation algorithm to address, and could be able to get the correct value when the address exists in the database. But when the user enters with some mistake he/she cannot get satisfactory results or even cannot get one result. In this paper, I infer the user's intention by the address dictionary, using natural language processing technology that greatly increase the address matching robustness and intelligence.

The raw address database from the supplier is fixed only. First of all we should ensure that the input can be properly matched to the address when it is in the original database. I created a Trie tree data structure by using the original address database and did word segmentation by dynamic programming method that increase the accuracy and efficiency greatly.

When the user’s input does not exist in the address database in the case, the above method will fail. In order to solve this problem, this paper created word position tagging (B-begin, E-end, M-middle, S-single) for the whole address database, then created hidden Markov model and trained the model parameters. According to the performance, this paper used the Viterbi algorithm which is based on dynamic programming. All the doings increased the scope of word segmentation.

After the address segmentation, we still need to determine the type for each word and know the word belongs to the province, city, county, street or the point of interest and to correct the use’s input by referencing the address database. This paper created a n-gram based inverted index file which could be fast index related to a set of words. Then get the maximum matching by calculating the similarity and sorting candidates.

Word segmentation and matching are two most important parts for address recognition. How to deal with these two steps is the key point of this paper.

Key words: Trie tree, hidden Markov model, n-gram, Viterbi algorithm

目 录

第一章 绪论

1.1 研究背景及意义

据报道，截至2012第四季度，中国的私人汽车的保有量已经达到93,090,000台。全国乘用车销售量连年增长。随着汽车产业的发展，汽车导航产业也得到快速发展，基于位置的生活信息服务就是其中的一块重要服务内容。汽车导航系统通过地理位置编码技术，将丰富的门牌号地址信息和兴趣点信息等实现系统的空间位置商业标记标注，满足商业软件系统对信息的准确性、丰富性的要求，大大丰富了地图数据的应用。

中文地址名称识别，是指建立给定空间坐标与地理位置信息一致性的过程。通过一段地址位置描述信息，确定该位置的经纬度坐标，也是各大商业GIS系统比较重要的一个功能。

反向地址名称识别，实现了将地球的经纬度坐标转换为标准地址的过程，反向地址名称识别实现了坐标定位地址的功能，辅助用户通过地面上的地理坐标反向查询所在的行政区划，街道和最佳匹配名称等标准信息。

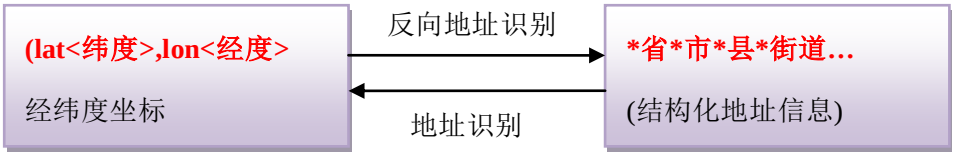

图 1- 1 正反地址识别说明

传统地址识别方法需要提供结构化的输入，也就是需要指定地址信息个各种属性信息。而现代地址识别由于采用了人工智能的技术，用户可以任意输入，由计算机理解用户的意图，提供了更加智能的表现形式。它的处理流程如下：首先对地址进行分词，然后识别每一单词的属性和优化，考虑到用户的输入的名称可能与地图数据库不完全匹配，这时需要利用地图数据库进行多次校准，找到最大可能的地址，成功后返回给地址对应的位置（即经纬度）或返回为空。如下图。

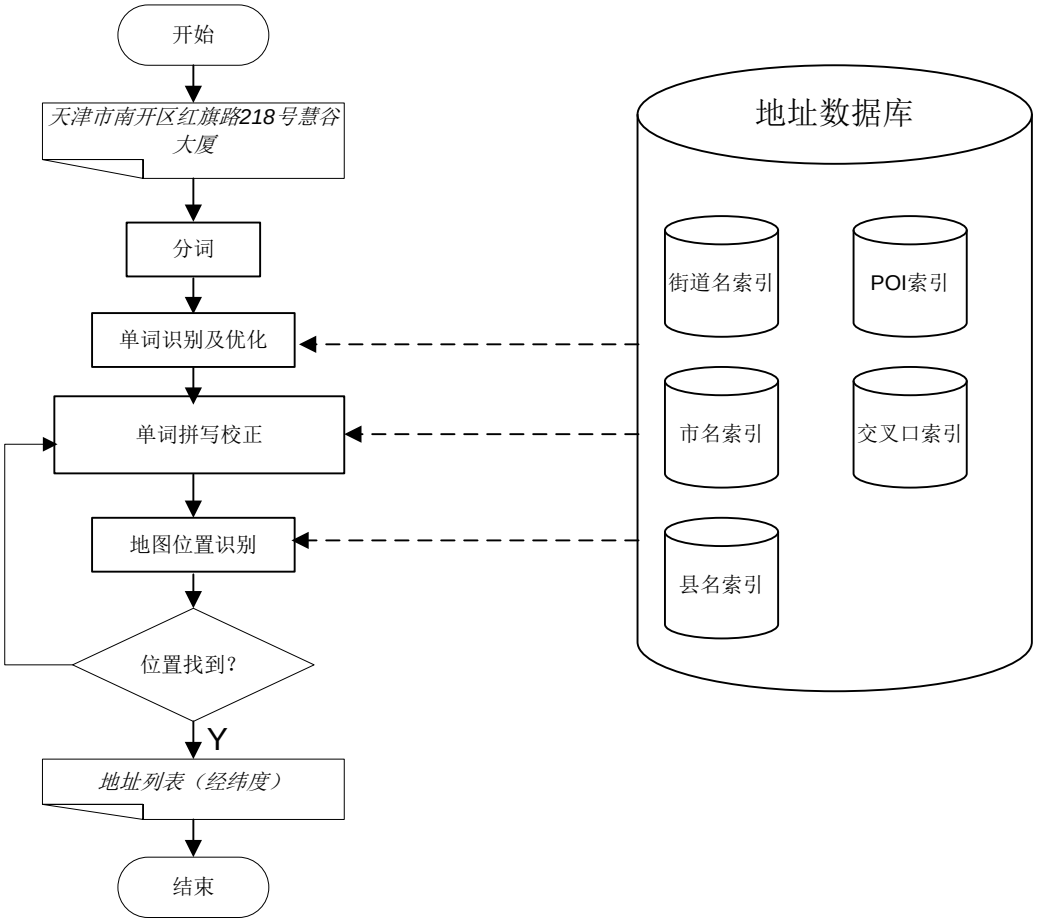

图 1- 2 智能地址匹配流程图

传统地址识别举例：输入“<省>,<市>：天津市，<区/县>：南开区，<街道名>：白堤路，<门牌号>：233号”，则输出应为“lon: 117.156387, lat: 39.109090”。

反向地址识别举例：输入“lat: 39.109090,lon: 117.156387”，正确的输出应该是“天津市南开区白堤路233号三兴望新电子公司”。

现代地址识别举例：输入“天津市南开区白堤路233号三兴望新电子公司”，则输出应为“lon: 117.156387, lat: 39.109090”。

一些大型公司比如Goolgle、Bing、Yahoo、TomTom和Baidu地图等已经实现了基本地址识别（Freefrom Geocoding）技术，并提供了开放的API接口。但在中文方面还不够完善，鲁棒性和智能型还有待加强。地理信息编码也随着智能搜索技术不断完善。

比如打开百度地图，比如我们搜索“南开区福寿堂药店”能够准确搜索到位置。（本次测试执行于2014年9月18日）

图 1- 3 百度地图正确搜索结果

但是当我们输错一个字，输入成“南开区福秀堂药店”时，显示没有结果，界面甚至不能给我一个相近及相关的提示。

图 1- 4 百度地图未能搜索到结果

当我们调用百度 Geocoding 的 API（http://api.map.baidu.com/geocoder?address=南开区福秀堂药店&output=json&key=HetKq4Tuc3IeLQoxOa2zHA9b&city=天津市）时，也没有得到正确结果（从结果我们可以看到给出了两个不同的经纬度，并且通过数据我们知道天津市根本没有“福秀堂药店”）。

“南开区福秀堂药店”	“南开区福寿堂药店”
{	{
"status":"OK",	"status":"OK",
"result":{	"result":{
"location":{	"location":{
"lng":117.162728,	"lng":117.155422,
"lat":39.116987	"lat":39.110989
},	},

图 1- 5 百度地图 API 返回结果

当我们搜索Google地图是也是相同的结果。

图 1- 6 谷歌地图未能搜索到结果

1.2 国内外研究现状

通常意义的地理编码技术大多是针对原始数据进行的二次整理，只是针对某一地理属性字段或某一主题进行编码，形式固定，用户输入受到很多限制，

稍有差错，就搜不到任何结果，如ArcGIS软件中的地理编码，需要对各个字段分别输入才能得到对应的结果。国外在这方面的研究已经比较成熟，也有不少商业化的地址匹配引擎服务，但这些技术都是建立在国外地址模型基础上，对于中文的地址匹配效果并不明显。比如英文单词之间有空格，而中文句子是没有空格的，这就导致中文分词和英文有明显的区别。还有我们输入地址的习惯大部分是正向逐步详细的，而英文恰恰相反等等。

马照亭，孙志刚，孙伟，印洁在《一种基于地址分词的自动地理编码算法》[5]中提出了分词的概念，建立地址模型和地理编码库并在地址匹配时对地理信息的不同属性赋予了不同的权重并给出了总体匹配准确度的计算方法。但是，在该算法中，由于所有的地址编码数据库的元素包括在总的地址的地址元素字典，匹配精度完全取决于地址分词，丰富程度取决于地址。当地址词典中不存时，不能给出任何结果。

孙亚夫，陈文斌在《基于分词的地址匹配技术》[6]中采用的地址识别引擎是基于“正向最大匹配分词”的地址分词算法。但是从分析结果来看，识别引擎对方位词、非标准地址、未登录地址或其他词的识别率很低。这也是由于地址的拆分完全取决于地址词库，不能对词库范围之外的地址要素或单词进行识别，并且也不能解决稍复杂的地址语言。

吴海涛，俞立，张贵军在《基于模糊匹配策略的城市中文地址编码系统》[7]中，提出一种基于K叉地址树模糊匹配策略，将地址数据以K叉树形式进行存储。采用分支定界法来检测和消除无效识别的节点,并对识别结果用模糊识别规则进行过滤及评估，以提高地址识别精度和效率。

小结：前面对一些相关文献的学习和研究可以了解到国内大部分研究及商业地址引擎缺乏鲁棒性的控制，对未登录词和错误地址词的理解能力有限，效率优化方面没有进行充分阐述。依然存在像百度地图或谷歌地图那样存在查不到数据的情况。

1.3 研究内容及组织结构

1.3.1 研究内容

论文研究了解当前地址名称识别发展趋势、运用技术等内容，仔细研究企业单位实际需求情况，将现代自然语言处理技术引入到该领域，设计解决该需求的一整套方案。

前面对一些相关文献的学习和研究，提出了一些自己的优化方案，思路主要体现在这几方面：增加鲁棒性的控制，分词方面本文采用了概率模型和隐马尔可夫模型分词，并利用动态规划搜索，提高了效率。地址匹配方面利用了n-gram模型，通过计算文本相似度，追踪用户最大可能意图。可以这么说，该模型在任何用户输入下都能都按照打分给出结果列表。而不像百度地图或谷歌地图那样存在查不到数据的情况。

1.3.2 论文组织结构

论文的基本组织形式如下：

第一部分：由于地图供应商的不同，导致地图格式也有很大不同，所以需要从不同数据里确定我们需要的地理信息并统一到我们需要的格式。

第二部分：由于不同地图生产商采用不同的数据格式，我们需要抽取我们所需要的部分，并组建自己定义的统一的地址词典格式。

第三部分：地理编码的好坏，分词起决定性作用，我们讨论了基于我们数据格式的采用基于概率语言模型的分词。通过我们的基础数据，对每一个词的频度进行统计，为了增强效率采用Trie树格式存储处理的数据，并用动态规划算法，达到快速分词的目的。

第四部分：基于概率语言模型的分词只是在满足用户完全正确输入单词的情况下才能正确，其实大部分用户会只输入几个关键单词。在这种情况下，我们采用基于隐马尔可夫（HMM）的分词。通过预测用户输入行为，确定隐马尔可夫模型的参数。

第五部分：在上面两次分词结束后，我通过n-gram技术进行输入矫正，和对分词后的每一单词属性的判别。并且由于采用了n-gram我们能够在用户输入过程中实时对用户输入联想，帮助用户最快确定输入名称。并对结果进行测试。

第六部分：针对我采用的如上模型，对该模型做一个技术性的总结，提出该算法的优缺点，和考虑如何能够改进。

第二章 论文涉及关键技术综述

本论文用到了较多的自然语言处理技术，现将它们逐个说明。

2.1 Trie 树的组织结构

数字搜索树采用了逐字散列的方法，可以看成是一种按照字的散列。一个数字搜索 Trie 树的一个节点只保留一个字符。如果一个单词比一个字符长，则包含第一个字符的节点有指针指向下一个字符的节点，依此类推。从而形成一个树的层次结构，第一层包含第一个字符，第二层树包含第二字符，直到最后一个单词，词典中的最长单词的长度即该索引树的最大高度。例如，如下单词序列组成的词典（天津市/河东区/河北区/天塔道/天宝路/21 号/22 号）会生成如下图所示的数字搜索树，叶子节点的数字记录了单词的属性，本文为概率值：

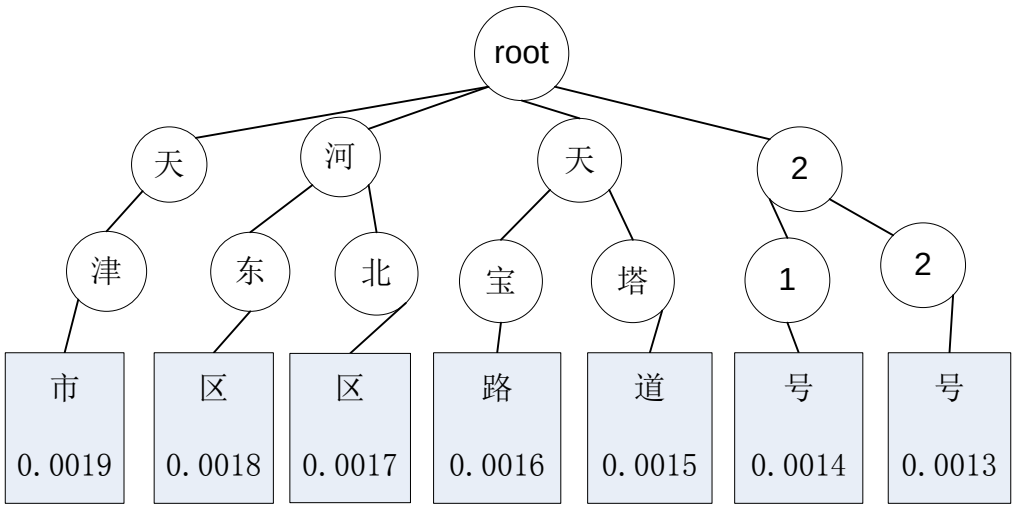

图 2- 1 Trie 树结构图

2.2 基于概率模型的分词

从统计思维的角度来看，分词问题的输入是一个字串 C=C_1,C_2,⋯,C_n，输出是另一个词串 S=W_1,W_2,⋯,W_m，这里 m≤n。对于一个特定的字符串 C，会有相应的多种分割方案，其任务是在在输入字符串的字符序列中找到一个最

大概率分割方案，该方案也就是最有可能的方案。

$$Seg(c)argmax\ P(S|C) = argmax\ \frac{P(C|S)P(S)}{P(C)} \tag{2-1}$$

例如对于输入字符串 “天津市河北工业大学”,有两种切分可能 S_1 和 S_2。

S_1：天津市/河北工业大学/

S_2：天津市/河北/工业大学/

计算条件概率 P(S_1 |C)和 P(S_2 |C)，然后选用概率大的切分方案。根据贝叶斯公式，有

$$\mathrm{P(S|C)} = \frac{\mathrm{P(C|S)\times P(S)}}{\mathrm{P(C)}} \tag{2-2}$$

此中 P(C)是字符串在该词典中存在的概率，为一个固定的值。从单词词串恢复到汉字串只有一种可能，因此 P(C | S)=1。因此，比较 P(S_1 |C)和 P(S_2 |C)的大小变成比较 P(S_1)和 P(S_2)的大小。

概率语言模型分词的目标是：在全切分所得的所有结果中找到一个切分方法 S，使得 P(S)最大。为了便于实现，如果每个词的概率上下文无关的，然后：

$$P(S) = P(W_1, W_2 \cdots, W_m) \approx P(W_1) \times P(W_2) \cdots \times P(W_m) \tag{2-3}$$

为了防止 P(S)下溢出变为零，取 log 后，精确度变大了。

$$P(S) \approx P(W_1) \times P(W_2) \cdots \times P(W_m) \propto logP(W_1) + logP(W_2) \cdots + logP(W_m) \tag{2-4}$$

计算任意一个词出现的概率方法如下：

$$P(W_i) = \frac{W_i\text{在语料库中的出现次数}n}{\text{该语料库中的总词数}N} \tag{2-5}$$

因此：

$$logP(W_i) = \log(Freq_w) - logN \tag{2-6}$$

如果在预处理阶段计算词概率的对数值，那么直接用加法就可以获得结果logP(S)，这样做是由于加法速度大于乘法，提高了效率。

计算最大概率等于求切分词图的最短路径。本文求解最短路径不采用迪杰斯特拉算法，而采用动态规划的方法。由该算法原理可知每个词之间的概率是上下文无关的，与动态规划求解所要求的最优子结构和无后效性相一致。在动态规划求解过程中并没有先生成所有可能的切分路径S_i，而是求出值最大的$P(S_i)$后，利用回溯的方法直接输出S_i。

节点$Node_i$的概率称为到节点$Node_i$为止的最大概率，所以：

$$P(Node_i) = P_{max}(W_1, W_2 \cdots W_i) = \frac{max}{W_j \in prev(Node_i)}(P(StartNode(W_j))) \times P(W_j) \quad (2\text{-}7)$$

如果W_j的结束节点是$Node_i$，则称W_j为$Node_i$的前驱词。节点i的前驱词集合定义为$prev(Node_i)$

其中$StartNode(W_j)$是W_j的开始节点，也是节点i的前驱节点。

因此切分的最大概率$\max(P(S))$就是：

$$P(Node_m) = P(\text{节点 m 的最佳前驱节点}) \times P(\text{节点 m 的最佳前驱词}) \quad (2\text{-}8)$$

首先把切分方案抽象成一个临接矩阵表示的图，然后从左到右计算最佳前驱节点。最后利用回溯法从终止节点找出最大概率的切分路径。

2.3 隐马尔科夫模型

隐马尔科夫模型（Hidden Markov Model, HMM）整合了频率和上下文两方面的特征来取得好的分词结果。具体来说，隐马尔科夫模型同时考虑了词的生成概率和词性之间的转移概率。

词性标注的任务是：给定次序列$W = w_1, w_2, \cdots, w_n$，寻找词性标注序列$T = t_1, t_2, \cdots, t_n$，最大化概率为$P(t_1, t_2, \cdots t_n | w_1, w_2, \cdots w_n)$。

使用贝叶斯公式重新描绘这个条件概率：

$$P(t_1, t_2, \cdots t_n) \times P(w_1, w_2, \cdots w_n | t_1, t_2, \cdots t_n) / P(w_1, w_2, \cdots w_n) \quad (2\text{-}9)$$

忽略掉分母$P(w_1, w_2, \cdots w_n)$，同时做独立性假设，使用N元模型近似计算$P(t_1, t_2, \cdots t_n)$。例如使用二元连接，则有：

$$P(t_1, t_2, \cdots t_n) \approx \prod_{i=1}^{n} P(t_i|t_{i-1}) \tag{2-10}$$

近似计算$P(w_1, w_2, \cdots w_n|t_1, t_2, \cdots t_n)$：假设一个类别中的词独立于它的邻居，则有：

$$\mathrm{P}(\mathrm{w}_1, \mathrm{w}_2, \cdots \mathrm{w}_\mathrm{n}|\mathrm{t}_1, \mathrm{t}_2, \cdots \mathrm{t}_\mathrm{n}) \approx \prod_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{P}(\mathrm{w}_\mathrm{i}|\mathrm{t}_\mathrm{i}) \tag{2-11}$$

寻找最有可能的词性标注序列实际的计算公式：

$$P(t_1, t_2, \cdots t_n) \times P(w_1, w_2, \cdots w_n|t_1, t_2, \cdots t_n) \approx \prod_{i=1}^{n} P(t_i|t_{i-1}) \times P(w_i|t_i) \tag{2-12}$$

这里把词w称之为显状态，词性t称为隐状态。条件概率$P(t_i|t_{i-1})$叫做转移概率，条件概率$P(w_i|t_i)$叫做发射概率。

2.4 维特比算法

维特比（Viterbi）算法实际是用动态规划解隐马尔科夫模型预测问题，即用动态规划(Dynamic Programming)求概率最大路径（最优路径）。这时一条路径对应着一个状态序列。

依据动态规划原理，最优路径具有以下特点：如果最优路径在时刻t通过结点i_t^*，那么这一路径从结点i_t^*到终点i_T^*的部分路径，对于从i_t^*到i_T^*的所有可能的部分路径来说，一定是最优的。

首先导入两个变量δ和ψ。定义在时刻t状态为i的全部单条路径$(i_1, i_2, \cdots, i_\mathrm{t})$中概率最大值是

$$\delta_t(\mathrm{i}) = \max_{i_1, i_2, \cdots, i_{\mathrm{t}-1}} \mathrm{P}(i_\mathrm{t} = \mathrm{i}, i_{\mathrm{t}-1}, \cdots, i_1, o_\mathrm{t}, \cdots, o_1|\lambda), \quad \mathrm{i} = 1,2, \cdots, \mathrm{N} \tag{2-13}$$

定义在时刻t状态为i的全部单条路径$(i_1, i_2, \cdots, i_{\mathrm{t}-1}, i)$中概率最大的路径的第t-1个结点为

$$\psi_t(\mathrm{i}) = \arg\max_{1\le \mathrm{j}\le \mathrm{N}}\left[\delta_{\mathrm{t}-1}(\mathrm{j})\mathrm{a}_{\mathrm{ji}}\right],\quad \mathrm{i} = 1,2,\cdots,\mathrm{N} \tag{2-14}$$

下面介绍维特比算法。

输入：模型$\lambda = (A,B,\pi)$和观测$O = (o_1,o_2,\cdots,o_T)$；

输出：最优路径$I^* = (\mathrm{i}_1^*,\mathrm{i}_2^*,\cdots,\mathrm{i}_\mathrm{T}^*)$

（1） 初始化

$$\delta_1(i) = \pi_i b_i(o_1),\quad i = 1,2,\cdots,N \tag{2-15}$$

$$\psi_1(i) = 0,\quad i = 1,2,\cdots,N \tag{2-16}$$

（2） 递推，对t=2,3,…,T

$$\delta_t(i) = \max_{1\le j\le N}\left[\delta_{t-1}(j)a_{ji}\right]b_i(o_t),\quad i = 1,2,\cdots,N \tag{2-17}$$

$$\psi_t(i) = \arg\max_{1\le j\le N}\left[\delta_{t-1}(j)a_{ji}\right],\quad i = 1,2,\cdots,N \tag{2-18}$$

（3） 终止

$$P^* = \max_{1\le i\le N}\delta_T(i) \tag{2-19}$$

$$i_T^* = \arg\max_{1\le i\le N}[\delta_T(i)] \tag{2-20}$$

（4） 最优路径回溯。对$t = T-1,T-2,\cdots,1$

$$i_t^* = \psi_{t+1}(i_{t+1}^*) \tag{2-21}$$

求得最优路径$\mathrm{I}^* = (i_1^*,i_2^*,\cdots,i_T^*)$。

2.5 文本相似性的度量

当用户输入部分错误或输入过程中，我们需要根据当前的片段，找到与该单词匹配度最大可能的名称，并对每一个可能值进行打分。所以我们需要判断文本相似度的指标。

2.5.1 Dice 系数(Sørensen–Dice coefficient)

给定关键词集合X 和Y ，定义相似度为共同信息(重叠部分)的两倍除以基数的总和。当作为一个字符串之间的相似性度量计算两个字符串之间的系数时，X和Y，使用二元模型的计算公式如下

$$S = \frac{2n_t}{n_x+n_y} \tag{2-22}$$

其中n_t是两个字符串共有的bigrams的个数，n_x是 x中bigrams的个数，n_y是y中bigrams的个数。例如要计算下面两个字符串之间的相似度:

“红旗路”

“红旗南路”

我们可以得出以下二元组集合，为了区分起始和结束字符，在开始和结束字符串添加标志字符$和^:

{$红，红旗，旗路，路^}

{$红，红旗，旗南，南路，路^}

两个集合分别有4和5个元素，它们有三个相同的元素: {$红，红旗，路^}。代入公式我们可以计算出：2 * 3 / (4 + 5) = 0.67。

2.5.2 编辑距离(Damerau–Levenshtein distance)

是指两个字符串之间，由一个转换到另一个所需的最少编辑操作次数。允许的编辑操作包含将一个字符替换成另一个字符，插入一个字符和删除一个字符。

例如将“马栏拉面”一字转成“兰州拉面”:

马栏拉面 （栏→兰）

兰拉面 （马→）

兰州拉面 （→州）

于是可以得出字符串“马栏拉面”和“兰州拉面”的编辑距离是3。

2.5.3 最长公共子串(Longest Common Subsequence, LCS)

假设 s1 = {a，b，c，b，d，a，b}，s2 = {b，d，c，a，b，a}，则从前往后找，s1 和 s2 的最长公共子串是 LCS（s1，s2）= {b，c，b，a}，如下图所示。

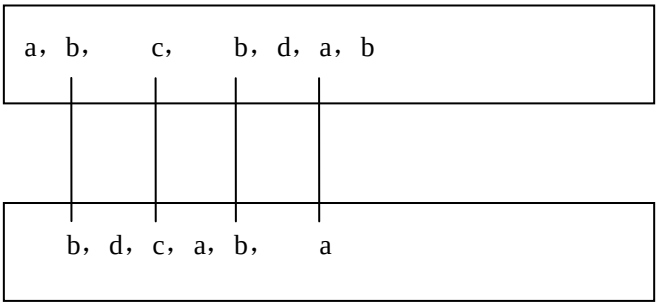

图 2- 2 LCS 计算结果

第三章　地址信息的获取与地址库的构建

由于原始地图数据类型不一，并且大部分是为导航和地图渲染准备的，故需要从原始地图数据中抽取只用于地理编码的，并规格化。我们的算法应该对这些数据是透明的，故我们需要把这些数据改成标准格式作为算法的输入，为方便起见，我们采用文本文件（.txt）格式。

3.1 电子数据地图的数据格式

Shapefile格式，SIF+格式，MapInfo格式和RDF地图格式是比较流行的地图提供商提供的地图的数据格式。现在的地图导航都是依赖于这些地图提供商提供的地图数据，地图导航企业需要对地图数据进行进一步的处理，把数据处理后，让它更适合导航的使用。在地图显示方面，快速的从改造后的地图数据中，提取有用的地图数据，并把数据转换成显示用的矢量地图数据格式，在计算导航路径上，地图数据的调用效率更高，速度更快。

从数据信息的详细程度看，SIF+具有比较长时间的发展历史，而且，数据的成熟度比较高，支持它的导航厂商也比较多。在本文所做的系统，它就是其中一个支持的数据类型。Shapefile是现在TomTom数据公司主推的一种数据格式，它的特点是全球覆盖面广，现在已经支持了全球大部分的国家，并且跟Iphone和BlackBerry等都有合作。

中国方面最主要的两大地图生产商是四维图新和高德，他们分别采用SIF+和Shapefile的文件格式。

3.2 地理信息的分类

大体上地理信息分为三种：点(POI, Point Address, Intersection)，线(Road)，面(Area)。每一个对象都会从属于某一行政区域（省，市/直辖市，县/区）。

从百度地图截取一段（如下地图），从该地图我们可以看到：点（华美里、郁美净孔雀园、天津银行红旗路支行等）；线（黄河道、红旗路、长江道等）；面（长虹生态园等）。

图 3- 1 地图上的地理信息

从用户搜索的习惯上区分，对于点又可分为三类：

1）兴趣点（POI:Point Of Interest），比如“天津市南开区慧谷大厦”。

2）点地址（Point Address），比如“天津市南开区红旗路218号”

3）交叉路口（Intersection），比如“天津市南开区红旗路与鞍山西道交口”

3.3 地理信息的抽取和地址库的构建

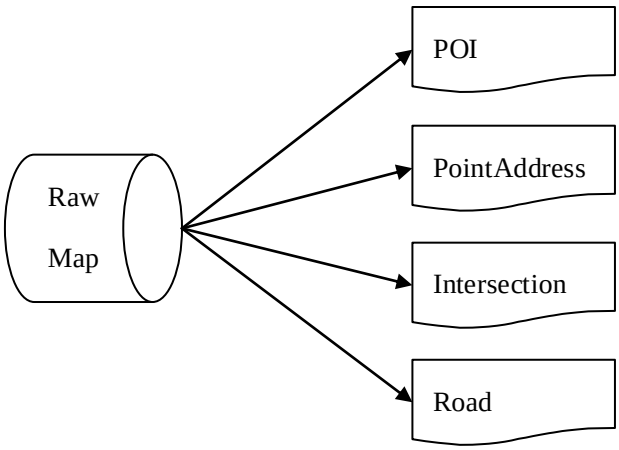

图 3- 2 地理信息数据的抽取

3.3.1 地址词属性分类

从原始数据可以看到，地址词属性大体可分类如下类别：

省，市，县，邮编，街道名，街道名+门牌号，门牌号，兴趣点，其他

3.3.2 地址信息的清洗

原始地址名称等还存在冗余，需要对原始信息进行再处理，包括去掉前缀、后缀和括号等等。比如

1) 去掉前缀

116.856474,39.643774|天津市|武清区|||**武清区**高村乡永兴庄村民委员会|

→

116.856474,39.643774|天津市|武清区|||高村乡永兴庄村民委员会|

2) 去掉括号

117.251057,39.986500|天津市|蓟县|||易捷便利店（**蓟县邦均中学南**）|

→

117.251057,39.986500|天津市|蓟县|||易捷便利店|

3.3.3 测试数据的说明

本论文测试数据采用了高德地图2011Q1，抽取了天津的数据。地理信息的每个元素用‘|’分开。测试数据组织如下：

1）兴趣点（tj=123112条记录）：

117.117239,39.134423|天津市|南开区|延安医院|

2）地址点（tj=33166条记录）：

117.117239,39.134423|天津市|南开区|南泥湾路|10号|

3）交叉路口（tj=7622条记录），一个路口可能会有多条公路交叉，根据用户输入习惯加入连词（与）和位置词（交口）：

117.190315,39.145859|天津市|南开区|东马路|与|北马路|与|大胡同|与|通北路|交口

4）道路（tj=4500条记录），由于它代表了一系列坐标，我们只抽取了第一个坐标，以利于搜索。

117.150665,39.116692|天津市|南开区|鞍山西道|

第四章　精确匹配地址库算法

4.1 问题的描述及需求分析

首先考虑用户正确输入时，系统能够正确分词。比如用户输入已经在地址库中的地址 “天津市南开区红旗路慧谷大厦”时，我们希望首先得到这样的分词结果“天津市/南开区/红旗路/慧谷大厦/”。考虑到地址库中的地址数目也是很可观的，我们需要设计一个结构，能够保证所有这些地址能够达到最大程度的正确。

由于原始地图厂商采集数据的不规范性，从地址库可以看到，由于好多地址名称与地域名称存在叠加情况，比如地址库中经常会出现如下数据，我们的程序还需要考虑到这种情况。

117.163042,39.497596|天津市|武清区|||天津市武清区佳龙工艺品厂|

给定地址库进行分词的方法有以下两类方法。

1）　机械匹配方法：例如正向或逆向最大长度匹配的方法。

2）　统计方法：如基于最大熵和概率语言模型的中文分词方法。

比较简单的实现是基于正向最大长度匹配的分词方法。每次从词典中搜索与待匹配串的前缀最长匹配的词，假如找到相应的匹配词，就把这个词作为切分词，同时从待匹配串中减去该词。如果词典中没有词与其匹配，则按单字切分。逆向就是采用相反的顺序，其他逻辑和正向一样。因为汉语的主干成分大部分在句子后面，所以逆向最大长度切分精度比正向稍高。

从地址库可以看到，由于好多地址名称与地域名称存在叠加情况，采用“正向最大匹配分词”或“逆向最大匹配分词”的方法往往使分词结果不够充分。比如当对地址“天津市武清区佳龙工艺品厂”进行分词时，句子被完全覆盖，不能得到“天津市/武清区/佳龙工艺品厂”的分词结果。

机械匹配的方法没有考虑到地址单词的频率特性。

从地址库中可以看到，每个地址单词在地址库中出现的频率是不同的，比如广域范围的地址词“天津市”、“红旗路”等出现的次数是很多的，而非常确

定的地址词比如“慧谷大厦”则次数很少，这种在地址库中的次数就构成了每个词的频率。进而能够得出该地址词的概率。

程序也需要同时考虑效率因素。

4.2 解决方案综述

本章采用通过已知地址库构造分词词典，然后根据分词词典构造Trie树，并采用基于概率语言模型的分词方法进行分词。对于用户输入的一段地址描述信息，我们首要是要对这句话进行分词，然后确定每一元素（word）确定其属性（省、市/、县/区、兴趣名称、地址名称等）。本章采用基于概率语言模型的分词方法。该模型考虑的词的频率因素。

4.2.1 构造地址分词词典

字典格式，是可以方便的人工查看和编辑文本文件格式。每行一个单词，对应的频率和词性是最基本的文本文件格式。基于词典的中文分词方法的基础是词典匹配。词典中单词的规模通常在几十万词以上。为了保证分词速度，选择一个好的查找词典方法是成败的关键。本文采用Trie树组织词典结构，分词采用最大概率查找的方法。

通过抽取的地理信息文件通过统计，构造词典。其中词典文件的每一行对应一个单词，由三部分组成：名称，频数，词性（n代表名词）。具体结果举例如下：

```
…
德悦洗车行 1 n
德悦达大酒店 1 n
红旗精品时装 1 n
红旗自行车有限公司 1 n
红旗自选商店 1 n
红旗路 193 n
…
```

图 4- 1 地址词典文件举例

4.2.2 基于概率语言模型的地址分词方法的算法设计

基于概率模型的算法流程图如下：

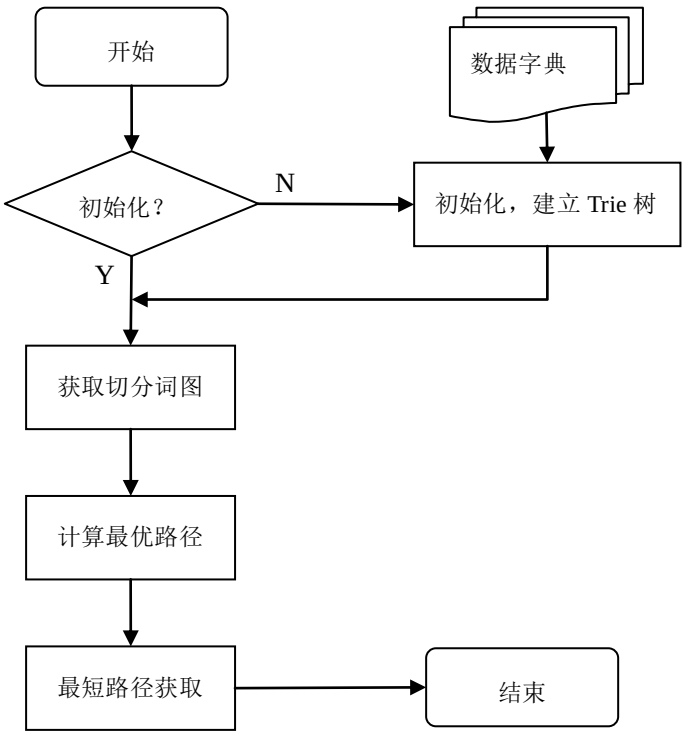

图 4- 2 概率分词算法流程图

基于概率语言模型的分词算法如上图所示，具体算法步骤如下：

1）首先获取用户输入地址信息。

2）判断地址字典是否已经初始化，没有则根据地址库信息，统计单词频率并建立Trie数据字典。

3）用户输入地址利用地址字典建立切分词图（即DAG，有向无环图），记录所有的切分可能。

地址切分词图的定义如下：如果字符串分割有m个字符，每个字符的左边和右边的位置，有相应的M + 1个点，点的编号从0到m。把候选词当作边，能够按照字典生成一个切分词图，该切分词图实际上是一个正权重的有向图。比如“南开区红旗路”这句话的切分词图如下所示。

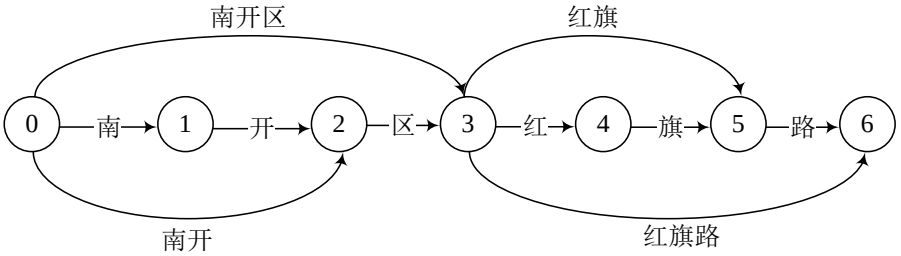

图 4- 3 切分词图生成

在“南开区红旗路”的切分词图中：“南”这条边的起点是0，终点是1；“南开”这条边的起点是0，终点是2，以此类推。切分方案就是从起点0到终点6之间的路径，该示例字符串共存在如下四条切分路径。

路径1：0—3—5—6 对应的切分方案S1：南开区/红旗/路

路径2：0—3—6 对应的切分方案S2：南开区/红旗路

路径3：0—2—3—6 对应的切分方案S3：南开/区/红旗路

路径4：0—2—3—5—6 对应的切分方案S4：南开/区/红旗/路

切分词图中的边即词典中存在的单词，边的起点和终点分别对应该单词的起始和结束的位置。

4）由于每个单词有频率属性，采用动态规划（即DP，Dynamic Programming）的方法在切分词图上找到最大概率的切分路径。由于该算法比较简单，本论文不再赘述。

4.3 解决方案效果评价

通过对天津的141799条地址记录，通过本算法与逆向最大长度匹配算法比较，得出基于概率的分词算法效率低于逆向最大长度匹配算法30%左右，这主要是由于动态规划算法慢于直接长度比较，但也是可以忍受的。而由于原始数据地址的不规则性，导致准确率方面要高于逆向最大长度匹配算法50%左右。

4.4 实验设计及结果分析

该算法对已经在字典里的单词，分词成功率是很高的。对未收录在字典里的单词分词往往失败。

1）成功分词的例子

天津市滨海新区吉祥经营部
天津市滨海新区盛源香烟精品店
天津市静海县胜利路与陈大公路交口
天津市南开区红旗路107号
分词结果：
天津市/滨海新区/吉祥经营部
天津市/滨海新区/盛源香烟精品店
天津市/静海县/胜利路/与/陈大公路/交口
天津市/南开区/红旗路/107/号

2）当我们对句子进行细微改动时，可以看到分词基本是错误的。
天津市滨海区吉祥经营部　　[去掉“海”]
天津市宾海新区盛元香烟精品店　　[源　元，滨　宾]
分词结果：
天津市/滨海/区/吉祥经营部
天津市/宾/海/新/区/盛/元/香/烟/精/品/店

3）利用该分词对天津所有的地址数据（123112(POI) + 11065(Address) + 7622(Intersection) = 141799条）按不同比率进行交叉验证已知数据得到的结果如下。从结果可以看到随着测试数据的比率的增加，分词的准确率是逐步增大的。

测试数据数：141799×10%=14179条
样本数据数：50%，60%，70%，80%，90%，100%
测试结果如表4-1所示。

表 4- 1 准确率测试图表

样本数	样本百分比	准确率 1	准确率 2	准确率 3	平均准确率
708995	50%	52.13%	51.98%	52.22%	52.11%
850794	60%	57.75%	57.99%	57.90%	57.88%
992593	70%	64.51%	63.86%	63.78%	64.01%
1134392	80%	71.18%	70.94%	70.77%	70.96%
1276191	90%	78.68%	78.14%	78.60%	78.47%
141799	100%	85.85	85.53%	85.82%	85.73%

从趋势图也可以得到准确率是基本和样本百分比成正比的。

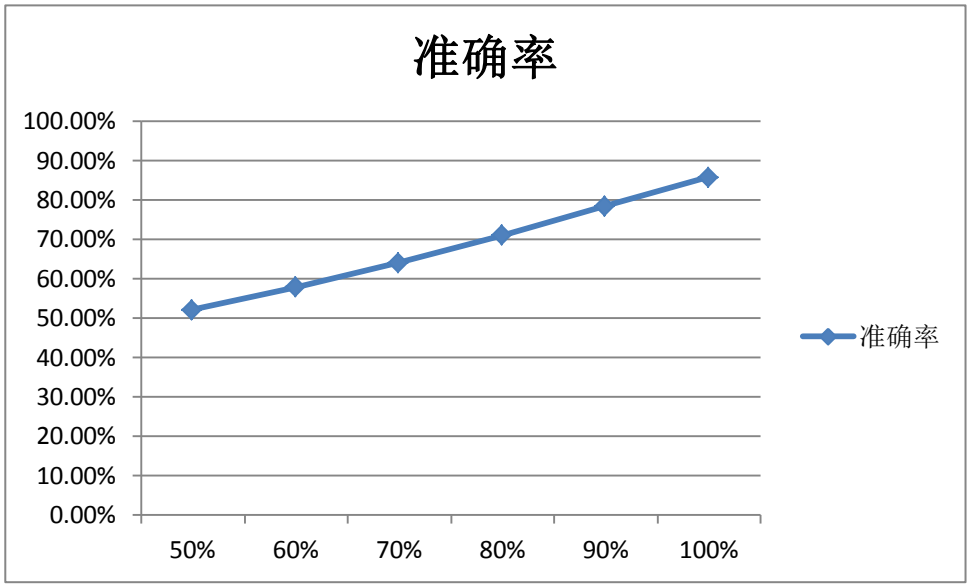

图 4- 4 准确率趋势图

4.5 讨论及本章小结

本章介绍了查找词典分词算法。查字典最早用首字母或者散列表实现，然后采用Trie树的方法开始流行。本章还介绍了如何从原始地图数据抽取出构造数据字典文件。正向最大长度匹配（Forward Maximum Match）和逆向最大长度匹配（Reverse Maximum Match）的方法效率高，实现起来相对简单，但没有考虑

到词的频率因素，精度要大打折扣。而基于概率的模型，频率（即概率）对分词起着主要作用，效率上要低于上面两种，但由于采用了Trie树和动态规划的算法，速度上是可以接受的，同时精度上优于上面两种。

通过测试我们还能看到，基于概率的分词在输入是字典中存在的数据时，分词精度还是比较准确的，但当用户输入稍微错误一点，就不能得到正确的结果，鲁棒性还有待加强。我们还需要新的算法来处理这个需求。

第五章　模糊匹配地址库算法

5.1 问题的描述及需求分析

基于概率语言模型的地址分词方法的前提条件即假定每个词之间的概率是上下文无关的，但实际情况并不是这样。

考虑到实际的需求，用户输入过程是五花八门的，不一定按照地址库的名称输入，并且用户。需要总结用户输入模式，并增大分词的鲁棒性。用户的常见错误输入大致可以分类如下（比如正确地址为“天津市南开区红旗路慧谷大厦”），我们的目标就是设计一个算法，能够在即使出现某种错误情况下，也能够比较准确的分词。

1）　输入部分：天津南开红旗慧谷

2）　输入错别字：天津市南开区红旗路**汇**谷大厦

3）　输入文字错误：天津市**南风**区红**齐**路**虎骨**大厦

基于概率的语言分词首先要求单词必须在地址库里存在，当出现上面情况下，采用该分词得到的结果如下（分词失败得到的结果都是一串单字），

1）　天/津/南/开/红/旗/慧/谷/

2）　天津市/南开区/红旗路/**汇**/谷/大/厦/

3）　天津市/南/风/区/红/秋/路/虎/骨/大/厦/

对此需求对分词应采用另一种方法，需引进上下文的概念。第一种情况下，这是由于在字典里不存在如下单词“红旗”、“南开”、“红旗”和“慧谷”四个，即属于“未登陆词”。但是引入上下文后，考虑由于词典存在 “南开区”，故可以判断“南”在有些情况下是分词的开头，而“区”是属于单词的结束，当在概率在某种情况下满足时，分词只需要知道“南”即可认定是一个新单词的开始。

5.2 解决方案综述

为了使分词的鲁棒性增加，本章分词引入隐马尔科夫模型，通过地址库构建马尔科夫参数并建立模型。单词在地址库中按照四个状态 BMES 来标记，其中 B 是开始（begin）位置， M 是中间（middle）位置， E 是结束（end）位置，

S 是单独成词（single）的位置，无前后。也就是说，地址库中的每一个单词采用了四种状态 (B,M,E,S)来给该地址单词做标记，比如红旗路可以标注为 BME，即红/B 旗/M 路/E，表示红是开始位置，旗是中间位置，路是结束位置。

这样整个地址库又变成了一个按照BEMS分类的状态库。

5.2.1 通过地址库利用最大似然估计法估计隐马尔可夫模型的参数

由上面分析可知，隐马尔科夫模型需要三个参数：初始概率、转移概率和发射概率三部分组成。本文将采用原始地址库，利用最大似然法估计该参数。

首先将地址库中的所有地址按照如下转换为地址状态库，如下：

117.251057,39.986500|天津市|蓟县|||易捷便利店|

→

117.251057,39.986500|BME|BE|||BMMME|

1）字状态间的转移概率的估计如下，该结果为4 x 4矩阵。

设样本中时刻t处于状态i时刻t+1转移到状态j的频数为A_{ij}，那么状态转移概率a_{ij}的估计是

$$a_{ij} = \frac{A_{ij}}{\sum_{j=1}^{N} A_{ij}}, \quad i = 1,2,\cdots,4 ; j = 1,2,\cdots,4 \tag{5-1}$$

2）发射概率的估计，假设地址库中的所有单子的个数为M，则结果为4xM的矩阵。

设样本中状态为j并观测数为k的频数是B_{jk}，那么状态为j发射为k的概率$b_j(k)$的估计是

$$b_j(k) = \frac{B_{jk}}{\sum_{k=1}^{M} B_{jk}}, j = 1,2,\cdots,4; \ k = 1,2,\cdots,M \tag{5-2}$$

3）状态概率的估计，该值为1x4的矩阵。

初始状态概率π_i的估计为S个样本中初始状态为q_i的频率。

4）测试数据计算的马氏参数

为了防止下溢出为零，本文仍然对概率进行了取log处理，概率为零的元素定为-3.14e+100。通过对测试例子，计算出的模型参数如下

```
#初始状态概率
-0.26268660809250016 -3.14e+100 -3.14e+100 -1.4652633398537678
#转移概率（BEMS）
-3.14e+100 -2.980478650491838 -0.052102598277761605 -3.14e+100
-0.04779069873908266 -3.14e+100 -3.14e+100 -3.0647244320282105
-3.14e+100 -1.0207277900252474 -0.4468069224753539 -3.14e+100
0.0 -3.14e+100 -3.14e+100 -3.14e+100
#发射概率（BEMS）
迷:-10.516131,石:-8.065126,迪:-9.146644,快:-8.933722,迦:-12.531034,……
迹:-13.224181,念:-12.125569,石:-10.826286,迪:-11.614743,忧:-13.224181,……
迹:-13.094194,忻:-13.49966,迸:-14.192806,迷:-12.246897,矶:-14.192806,……
与:0.0,
```

图 5- 1 地址词典训练的隐马尔科夫参数

5.2.2 采用维特比算法得到状态序列并进行分词

利用BEMS四个状态的初始状态，转移概率和发射概率进行分词。假设需要分词的句子为“南开区红旗路”。则每个阶段的每个隐藏状态和显示状态形成一个二维矩阵，即维特比求解栅格。

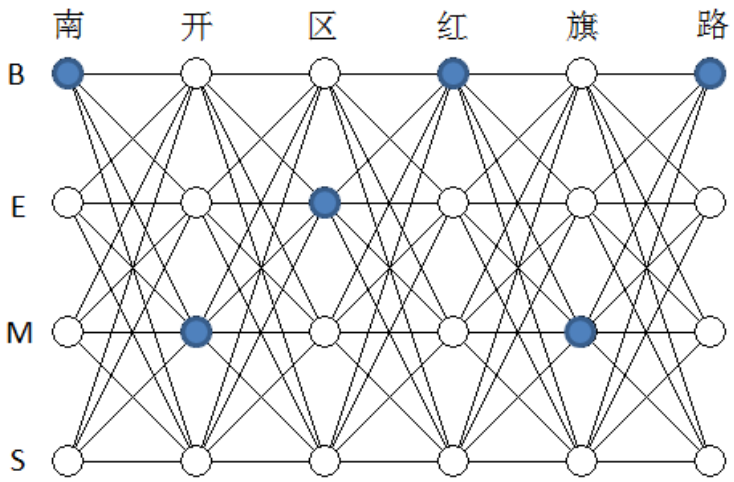

图 5- 2 维特比算法结果举例

举例如上图，句子“南开区红旗路”采用动态规划的方法求解得到最佳隐状态序列（蓝色圆球）为“BMEBME”，则根据隐状态含义应该分词为“BME/BME”，带入原始代词即得到分词结果“南开区/红旗路”

下面验证一下鲁棒性，比如假如用户不小心将该地址输入为“南开区洪旗路”，即用户将字符‘红’错输成‘洪’时，根据马氏参数计算得到最终求解栅格权值矩阵如表5-1所示：

表 5- 1 权值矩阵

	南	开	区	洪	旗	路
B	-3.612	-∞	-∞	-18.057	-36.525	-41.636
E	-∞	-∞	-10.444	-26.917	-32.316	***-28.858***
M	-∞	-8.026	-13.365	-23.313	-25.274	-29.772
S	-∞	-∞	-∞	-∞	-∞	-∞

路径回溯（即前一状态）矩阵如表5-2所示：

表 5- 2 回溯路径矩阵

	南	开	区	洪	旗	路
B		E	E	***E***	E	E
E		E	***M***	M	B	***M***
M		***B***	M	M	***B***	M
S		E	E	E	E	E

最后从最后一个字符利用上面两个矩阵进行回溯（粗斜黑体），计算得到最优状态为BMEBME，分词后得BME/BME，带入原始字符即得：南开区/洪旗路。也能得到比较正确的分词。

5.3 解决方案效果评价

通过模拟用户输入，对以下几类错误输入，算法基本上都能够给出正确结果

1） 天津南开红旗慧谷

转换后生成的状态序列如下

BMBMBMBM

根据状态含义，在B（Begin）的前面应该分词，得结果如下

BM/BM/BM/BM，即

天津/南开/红旗/慧谷

2） 天津市南开区红旗路汇谷大厦

转换后生成的状态序列如下

BMEBMEBMEMMME

根据状态含义，在E（End）的后面应该分词，得结果如下

BME/BME/BME/MMME，即

天津市/南开区/红旗路/汇谷大厦

3） 天津市南风区红秋路虎骨大厦

转换后生成的状态序列如下

BMEBMEBMEMMME

根据状态含义，在E（End）的后面应该分词，得结果如下

BME/BME/BME/MMME，即

天津市/南风区/红秋路/虎骨大厦

从上面的分析可以看到，由于只划分了这四个状态，分词的标志是在如下位置：状态B（Begin）之前、状态E（End）之后和S（Single）之前或之后。该算法不能保证任意输入都分词成功，但至少也对相当一部分的输入进行了正确的划分，这些情况在采用上一章的算法中是不能实现的。

5.4 实验设计及结果分析

我们对采用概率语言模型句子分词失败的句子进行测试，可以看到分词是正确的。我们可以得出结论，该算法使得分词的鲁棒性大大增加。

1）成功分词的例子

天津市滨海区吉祥经营部　　[去掉“海”]

天津市宾海新区盛元香烟精品店　　[源　元，滨　宾]

分词结果：

天津市/滨海区/吉祥经营部

天津市/宾海新区/盛元香烟精品店

2）结果统计分析如下。从结果来看当样本百分比大于60%后，结果就比较稳定了。

表 5- 3 结果准确率表

样本数	样本百分比	准确率 1	准确率 2	准确率 3	平均准确率
708995	50%	62.16%	63.10%	63.56%	62.94%
850794	60%	64.40%	64.19%	64.32%	64.30%
992593	70%	64.22%	64.26 %	64.26 %	64.25%
1134392	80%	64.02%	64.19%	63.93%	64.05%
1276191	90%	63.65%	64.32%	64.65%	64.21%

141799	100%	64.04%	64.00%	64.03%	64.02%

趋势图如下

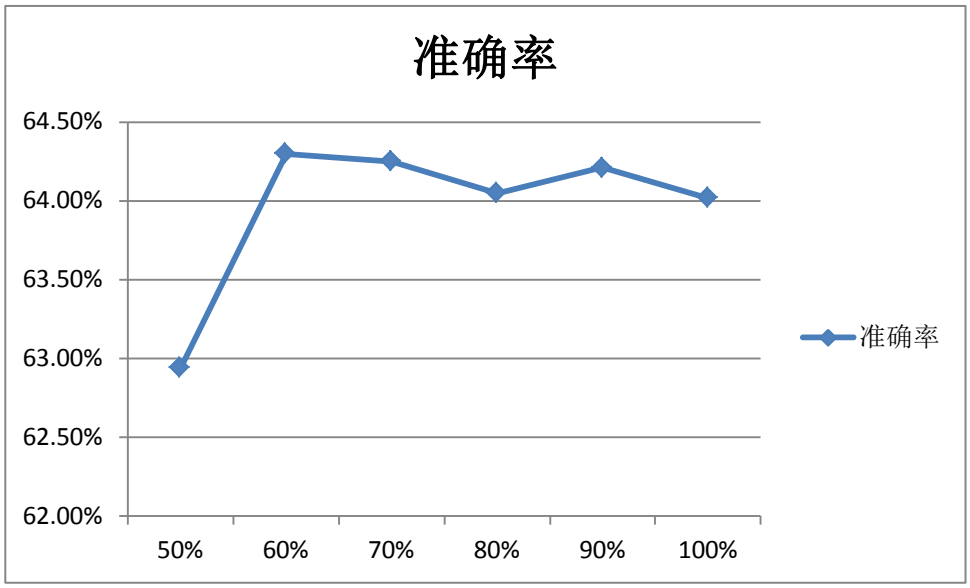

图 5- 3 结果准确率趋势图

5.5 实验设计及结果分析

标注问题还可以选用基于转换的学习方法（Transformation Basd Learning，TBL），先把每个字标注上最可能的词性，然后通过转换规则不断修正错误的标注，以提高标注精度。但最有名的还是基于隐马尔科夫的模型。

本文讨论了基于隐马尔科夫模型在分词中的应用。通过对每个字的状态分类为BEMS，将分词问题转化为标注问题。充分利用了隐马尔科夫模型的上下文相关性，使分词的鲁棒性大大增强。另外通过动态调整模型参数，使模型在应用过程中能够逐渐增强其准确性。本文还讨论了如何利用数据字典训练马氏模型的参数以及用维特比算法快速分词的过程。

第六章　地址单词属性标注及单词校正技术

6.1 问题的描述及需求分析

地理信息的单词属性大体可分为如下8种。每个单词对应一个或多个可能属性（如下），

```
typedefenum AddressComponent
{
    eState        = 0x00000001,//省
    eCity         = 0x00000002,//市
    eCounty       = 0x00000004,//县
    eZip          = 0x00000008,//邮编
    eStreet       = 0x00000010,//街道名
    eStreetHNo    = 0x00000040,//街道名+门牌号
    eHNo          = 0x00000080,//门牌号
    ePoi          = 0x00000100,//兴趣点
} AddressComponent;
```

图 6- 1 地址词类别属性

当分词完毕还不能真正满足用户需求，还有三点需要考虑。

1）属性判断：正确分词后，需要对每一单词的类别进行判断，从而能够大致在语义上理解用户的意图。

2)输入矫正：我们知道采用隐马尔科夫可以对用户输入错误地址进行分词，但分词后如何矫正每一个地址词，程序需要矫正的单词区进行地址库的匹配。

3）联想功能：用户输入过程中如何对每一个字符过程中及时的提供给用户建议。

以上三点在系统运行时近乎实时的特点，如何构造一个快速的索引结构，也是本章考虑的重点。

6.2 解决方案综述

根据用户中文输入习惯，基本采用关键语素后置，采用层层推进的方式，可将标准输入模式（即地址格式，Address Format）列表设置如下。用户输入满足下面格式称为一个“完美匹配”（Perfect Match）。程序通过遍历该格式列表，选择满足最大的匹配。

```
#天津市南开区红旗路
eCity/eCounty/eStreet
#天津市南开区红旗路 218 号
eCity/eCounty/eStreetHNo
eCity/eCounty/eStreet/eHNo
eCity/eCounty/ePoi
#天津市南开区红旗路与鞍山西道交口
eCity/eCounty/eStreet/与/eStreet/交口
eCity/eCounty/eStreet/eStreet
……
```

图 6- 2 地址格式模板文件

6.2.1 Bigram 倒排索引文件的生成

n-gram即N元语言模型，相对应的bigram即是二元语言模型。每一个单词被分为多个相邻的二元词元。为了特殊考虑将首尾分别加上特殊字符$和^。

比如：

红旗路　$红，红旗，旗路，路^

红旗南路　$红，红旗，旗南，南路，路^

对所有二元词元建立索引文件如下。索引文件与主文件是多对多的对应关系。如下图所示，左边的是索引文件（geograms.bin）。右边是主文件（geoadmins.bin），每个元素后面记录了其属性。

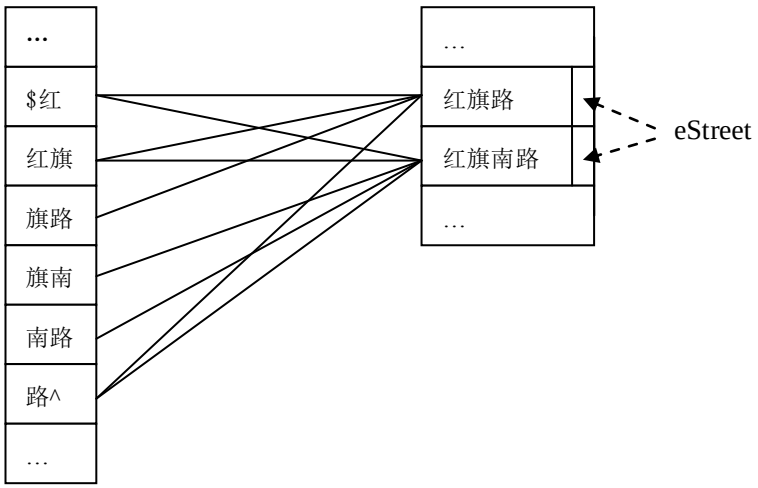

图 6- 3 Bigram 倒排索引文件结构

采用如上文件格式的好处是从用户输入的任意部分单词，能够快速搜索到对应的可能全称。并且简化了计算相似度的问题。

比如用户输入“红旗路”时，通过首先对该字符串分解成二元词元，然后查找主文件，能够迅速找与该单词有交集的地址库名称。其次求“红旗路”与“红旗南路”的Dice系数也不需要直接计算，只需找到两个的索引数组求交集然后与两个的个数和相除即可。

6.2.2 基于 n-gram 数据模型的地址匹配（含拼写校正）功能算法

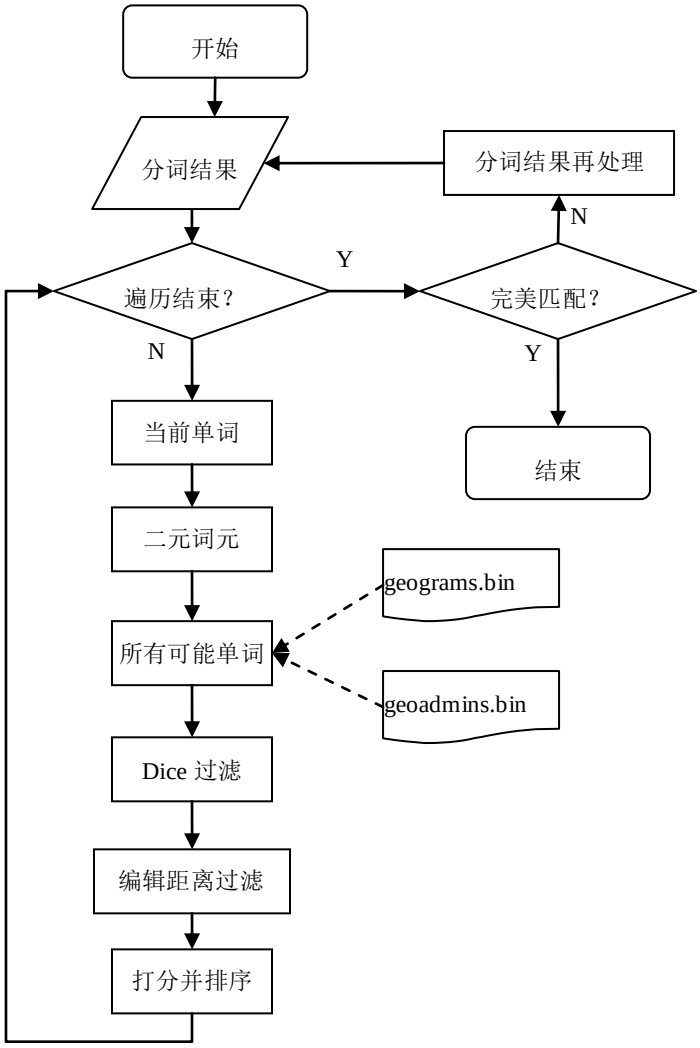

图 6- 4 地址匹配流程图

按照上图流程图，具体解释如下。比如用户输入“天津市南开区福秀堂药店”（地址库不存在“福秀堂药店”，而存在“福寿堂药店”）。首先采用上两章算法进行分词得到结果如下

天津市/南开区/福秀堂药店/

程序首先查找第一个单词“天津市”通过n-gram倒排索引文件找到该单词属性是eCity，置信度100%。转入第二个单词“南开区”也找到对应属性eCounty，置信度100%。转入第三个单词“福秀堂药店”，没有找到100%的，但根据打分排序，找到最大置信度的是“福寿堂药店”，属性ePoi，置信度是87.7%。这时最后一个，判断该句子的地址格式为“eCity/eCounty/ePoi”,在地址格式表（Address Format），定为完美匹配，退出并输出总置信度（100%+100%+87.7%）

/3=95.9%。

以下算法为探讨拼写校正方法：

1）计算二元词元。

2）遍历所有词元，找到所有可能单词。

3）遍历所有可能单词，计算Dice系数，选择该系数在某一阈值的单词。

3）所选择的单词按Dice系数排序。

4）遍历二次选择的单词，计算编辑距离。选出该距离在某一阈值之内的单词。

5）所选择的单词按编辑距离排序。

6）结束，最大可能单词出现在最前面。

6.2.3 基于 n-gram 数据模型的联想功能算法

因为当用户在部分输入时，流程图同上节。但由于当前字符串是全称的子集，不能按照Dice系数和编辑距离过滤，而采用最长公共子串（LCS）过滤。

6.3 解决方案效果评价

本算法通过对天津的141799条地址记录，模拟用户输入部分错误情况得到地址匹配成功率达到75%左右。但是由于本系统没有更多用户真实输入的数据样本，不能得到一个有效的判断。但通过本文的方法，是可以根据用户的各种不同的输入模式，逐步调节各个步骤参数的。在用户输入比较规范的情况下，是能够比较准确的反应用户的意图的。

6.4 实验设计及结果分析

6.4.1 拼写矫正结果分析

地址名称识别最主要的部分就是分词。即便我们采用了两种分词算法来提高分词的准确性，也会考虑到在某种情况下分词失败。所以我们采用了在原来分词的基础上，采用n-gram进行二次矫正，生成二次分词结果。

通过实际测试，输入的拼写校正率能够达到80%左右。

6.4.2 联想功能结果分析

以天津市范围数据作为测试用例。

1) 道路查询分析（“天津市南开区红旗路”）

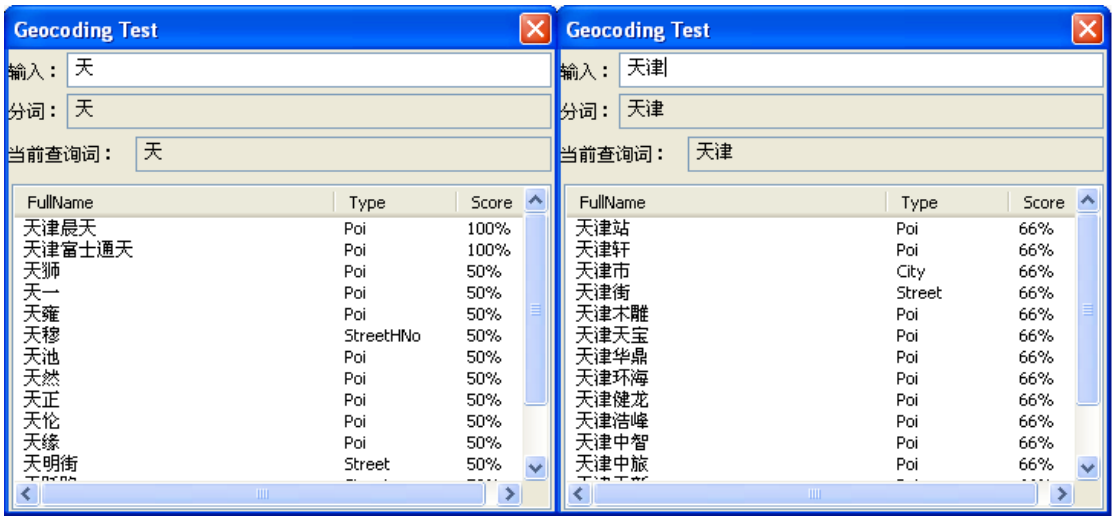

图 6- 5“道路查询”联想结果展示图

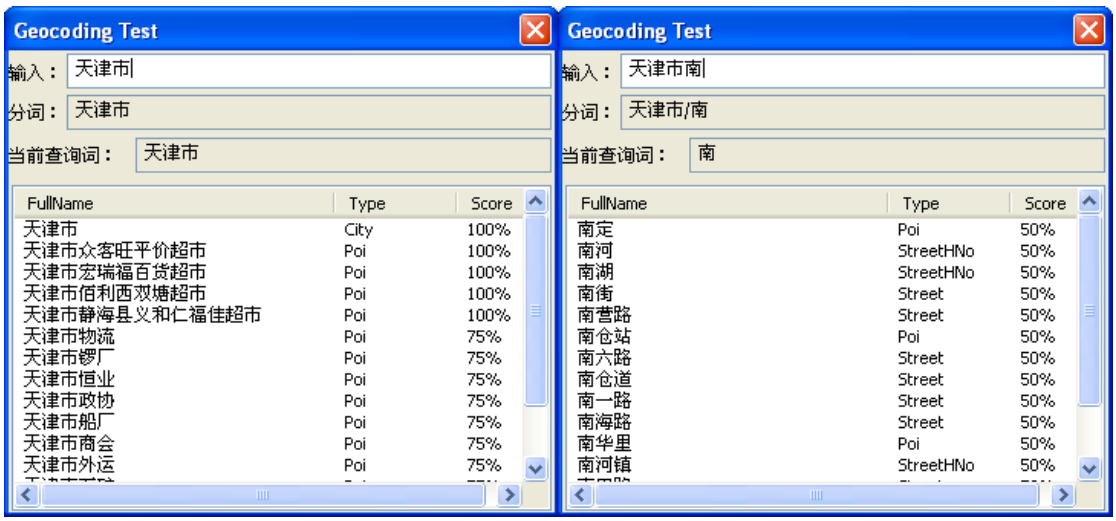

图 6- 6“道路查询”联想结果展示图

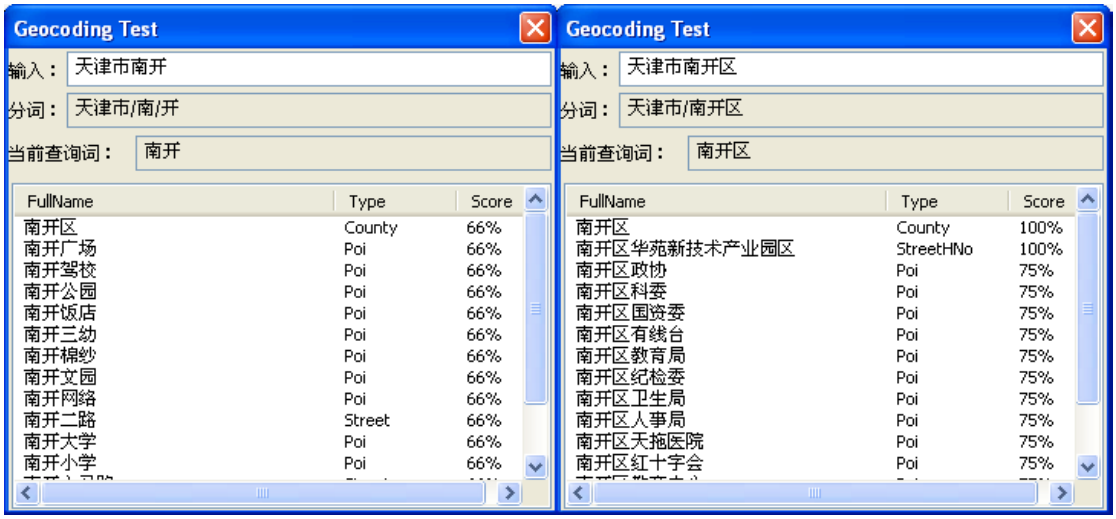

图 6- 7“道路查询”联想结果展示图

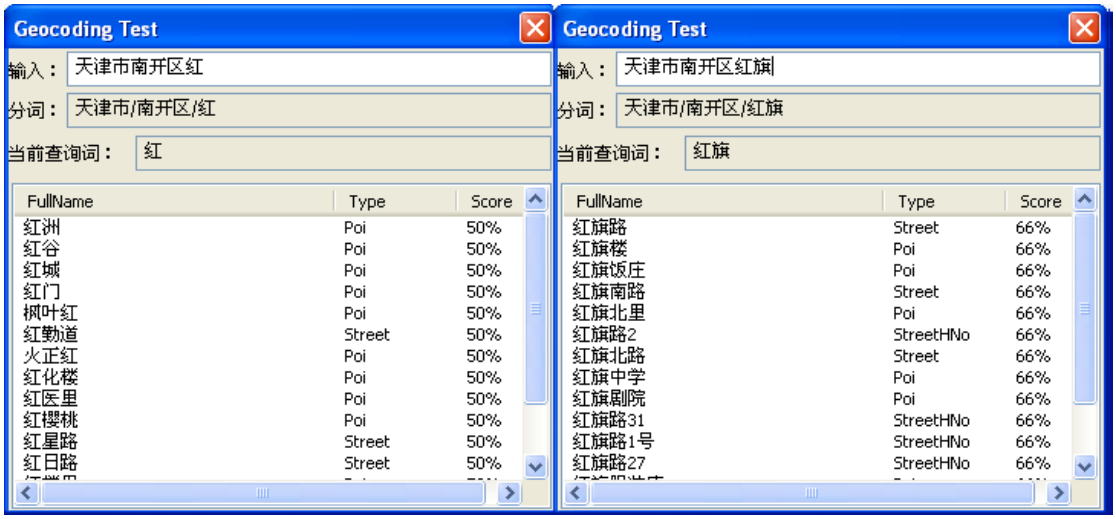

图 6- 8“道路查询”联想结果展示图

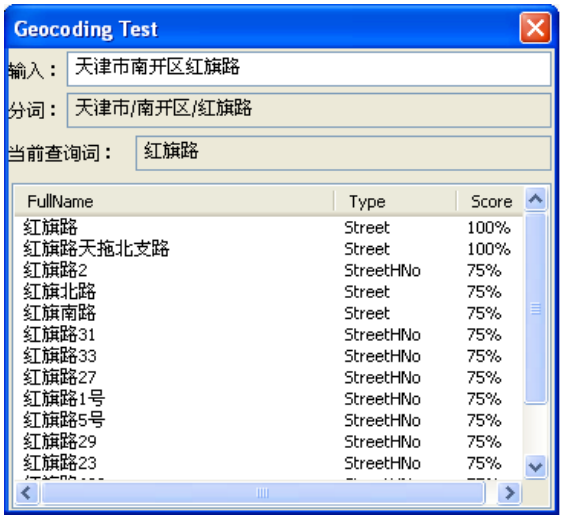

图 6- 9“道路查询”联想结果展示图

2) 交叉路口分析(“天津市南开区红旗路与鞍山西道”)

以天津市范围数据作为测试用例。

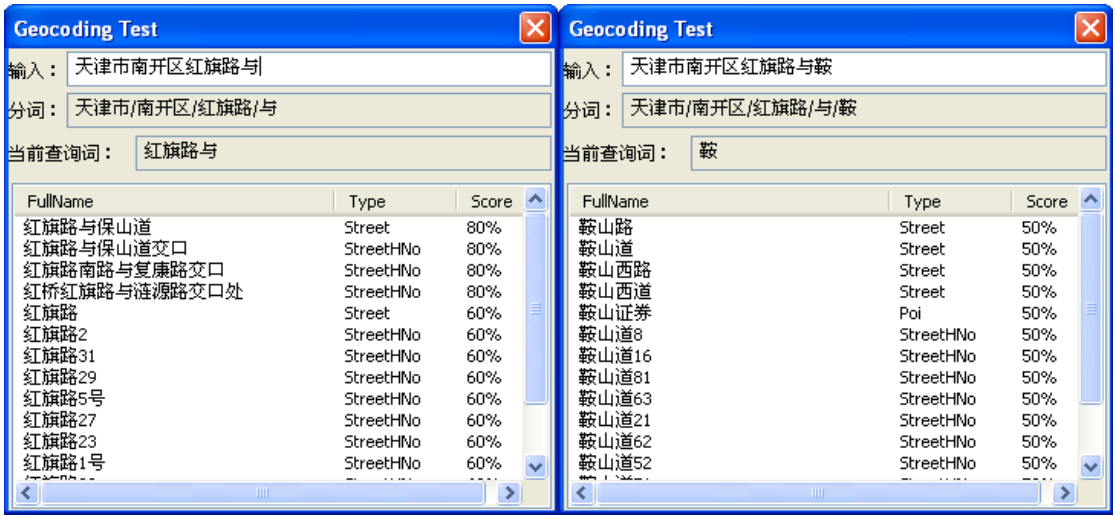

图 6- 10“交叉路口”联想结果展示图

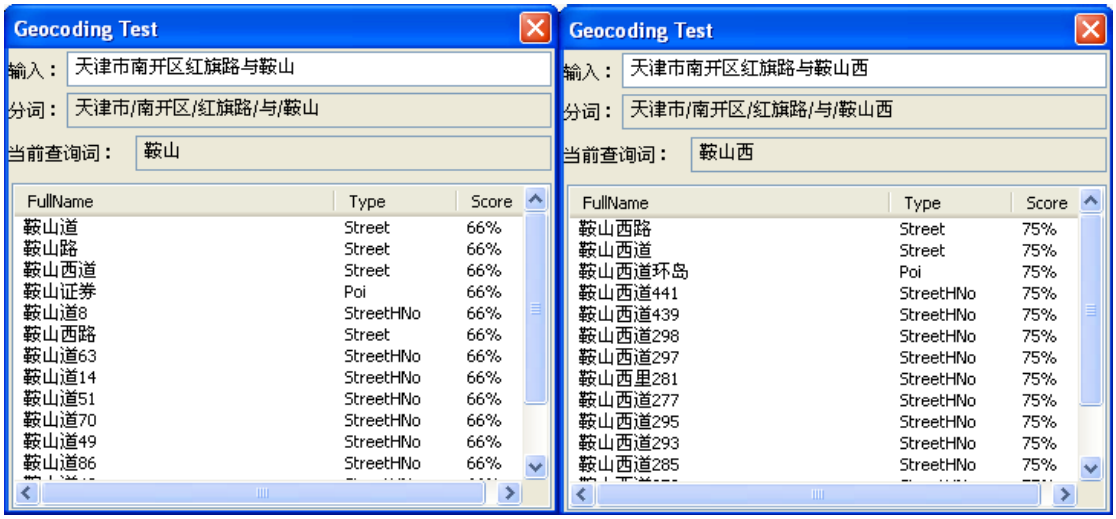

图 6- 11“交叉路口”联想结果展示图

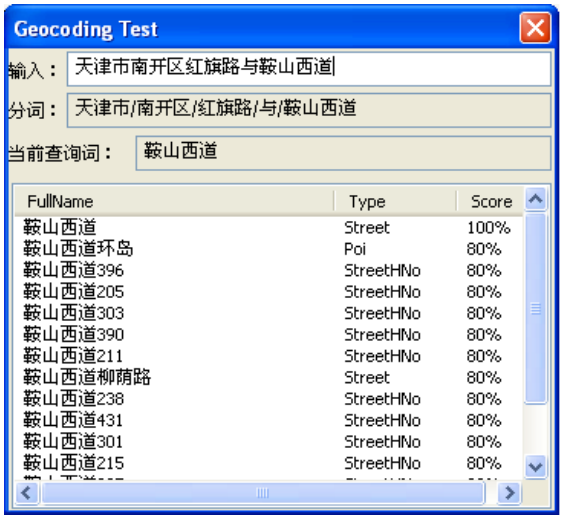

图 6- 12“交叉路口”联想结果展示图

3) POI输入分析(“天津市武清区华蕾农资超市”)

以天津市范围数据作为测试用例。

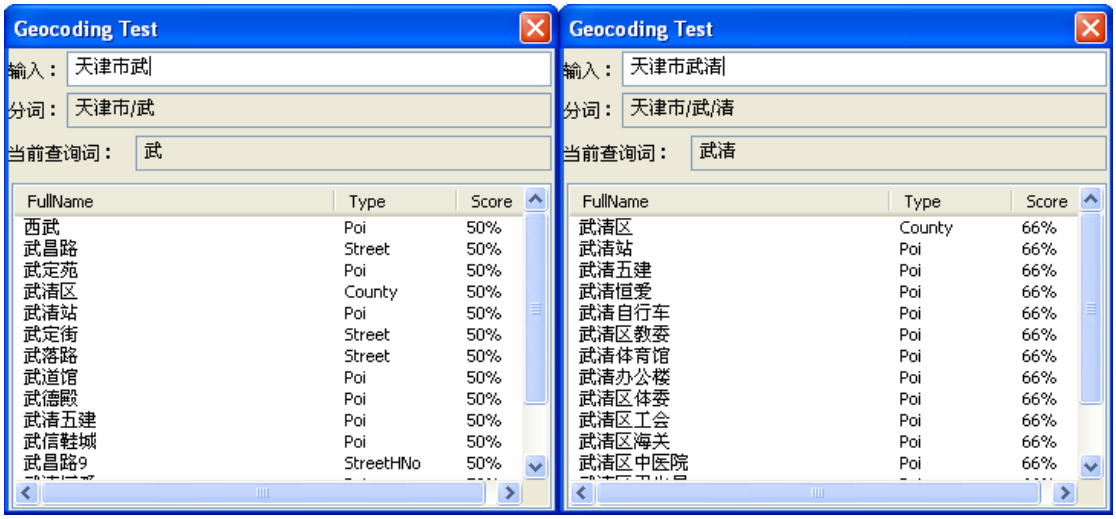

图 6- 13“POI”查询联想结果展示图

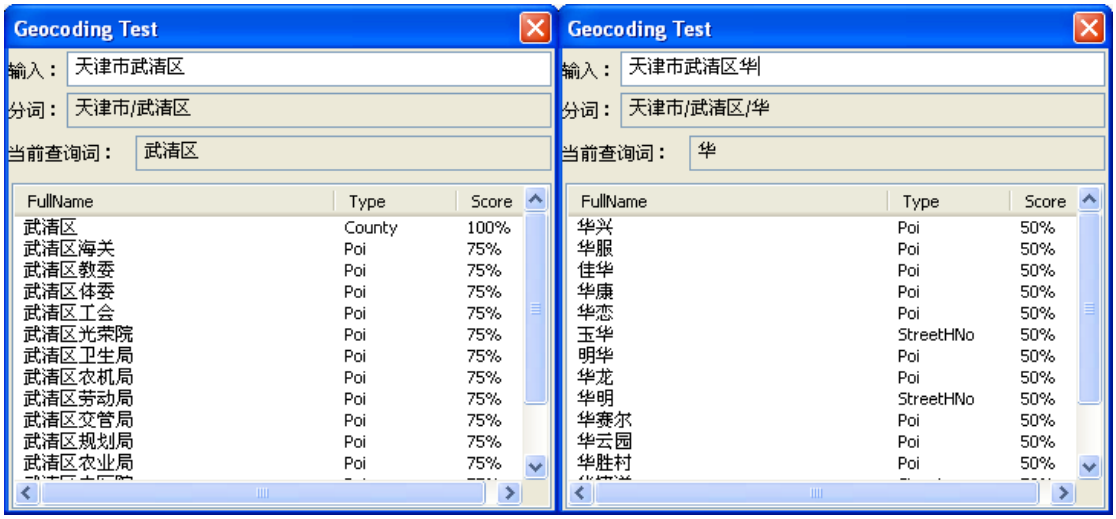

图 6- 14“POI”查询联想结果展示图

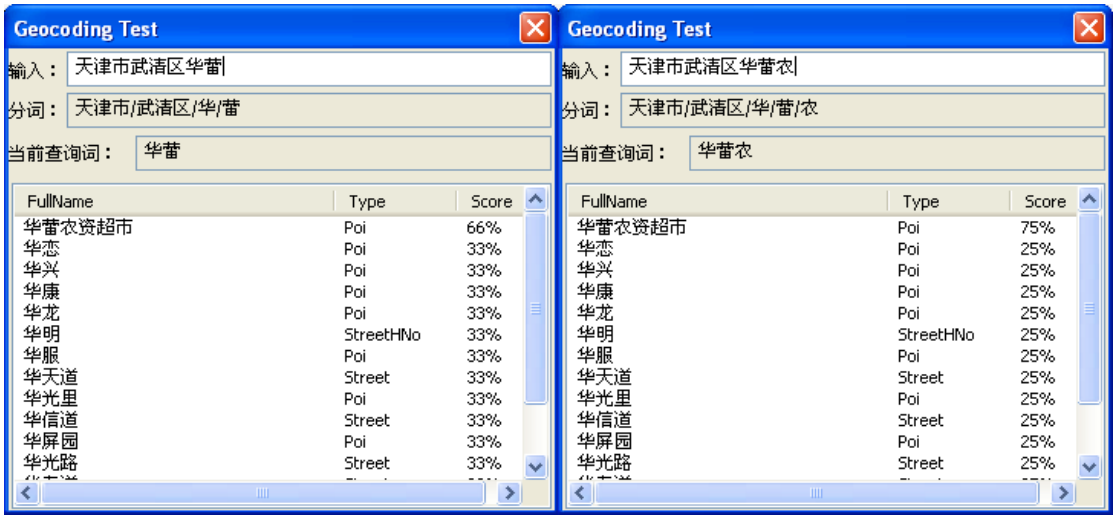

图 6- 15“POI”查询联想结果展示图

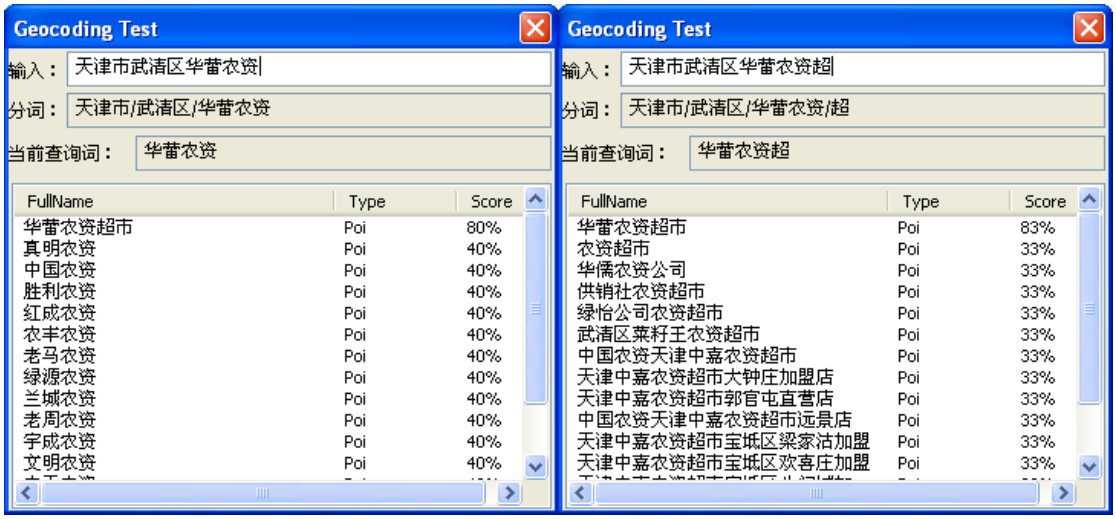

图 6- 16“POI”查询联想结果展示图

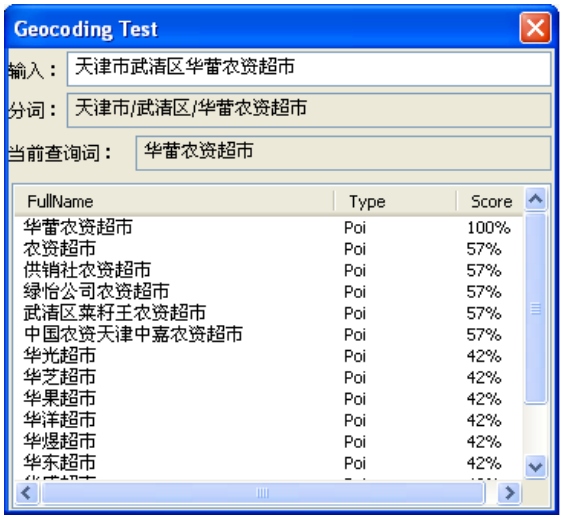

图 6- 17“POI”查询联想结果展示图

6.5 讨论及本章小结

对用户的输入进行分词后，我们还需要确定每一个单词的属性，这样才能在地图数据库里正确搜寻到结果。我们通过Dice系数、编辑距离和LCS等三个指标来确定每个单词的最大可能属性，并对可能属性列表进行打分并排序，交给用户选择或将最大可能作为用户输入的真正查询条件，从地图里搜索到结果，然后将位置等信息返回给客户。

即便我们使用了两种分词方法，考虑到用户输入的任意性，我们还需对用户输入进行模式匹配。当完全满足给定地址格式时我们称之为“完美匹配”，否则返回“最大匹配”。从测试结果来看，我们的算法在用户任意输入时都能否返回给一个理性结果，不存在百度或谷歌地图那样，返回结果为零的情况。

第七章　总结与展望

本文只是提供了利用现有技术，如何能够实现一个真正的智能地址匹配系统。实用性是本文考虑的重点。

7.1 总结

本文详细讨论了基于中文的地址名称识别技术的一整套解决方案，通过利用充分的测试数据证明该方案能够满足用户的需求，并通过与当前百度和谷歌地图的分析，证明了该解决方案在智能性和和鲁棒性方面明显优于它们的结果。

7.2 展望

本文的实现还是存在一些改善的方法。由于训练模型的地址库类型不同且规模有限，许多合理的地址搭配关系和习惯等在地址库中不一定存在，所以一定程度上会造成模型出现局部数据稀疏的现象。在统计自然语言处理中数据稀疏的一个表现即零概率问题。可以采用加法平滑或Good-Turing平滑算法实现，由于篇幅关系，本文没有进行这方面的探讨。

随着大数据、机器学习和人工智能的深入发展，地址名称识别技术一定会从传统的方式向更智能化的方向发展。

参考文献

[1]何建邦，李新通，对地理信息分类编码的认识与思考，地理学与国土研究，2002，18（3）：1~7

[2]江洲，李琦，地理编码(Geocoding)的应用研究，地理与地理信息科学，2003，19（3）：22~25

[3]李航，统计学习方法，北京：清华大学出版社，2012.171~190

[4]孙存群，周顺平，杨林，基于分级地名库的中文地理编码的研究，计算机应用，2010，30（7）：1953~1958

[5]马照亭，孙志刚，孙伟，印洁，一种基于地址分词的自动地理编码算法，测绘通报，2011，2：59~62

[6]孙亚夫，陈文斌，基于分词的地址匹配技术，测绘网

[7]马照亭，孙志刚，孙伟，印洁，基于规则的中文地址要素解析方法，地球信息科学学报，2010，12（1）：9~16

[8]罗刚，解密搜索引擎，北京：电子工业出版社，2014

[9]吴海涛，俞立，张贵军，基于模糊匹配策略的城市中文地址编码系统，人工智能及识别技术，2011，37（2）：194～196

[10]章意锋，贾凤海，毛其峰，侯武云，ARCGIS 中地理编码方法的改进，余姚市规划测绘设计院

[11]北京市质量技术监督局，地址数据库建设技术规范(报批稿)北京市地方标准

[12]M. J. Hutchinson, B. Veenendaal，AN AGENT-BASED FRAMEWORK TO ENABLE INTELLIGENT GEOCODING SERVICES，Dept. of Spatial Sciences, Curtin University of Technology, Perth, WA, Australia

[13]Hutchinson, Matthew John，Developing an agent-based framework for intelligent geocoding，Curtin University of Technology, Department of Spatial Sciences

[14] 龙树全，赵正文，唐华，中文分词算法概述，电脑知识与技术，2009，5（10）：2605～2607

[15] 孙铁利，刘延吉，中文分词技术的研究现状与困难，信息技术，2009，7（10）：187～189

[16] 王华栋，饶培伦，基于搜索引擎的中文分词评估方法，信息技术，2007，25（1）：108～112

[17] 何莘，王琬芜，自然语言检索中的中文分词技术研究进展及应用，信息技术，2008，26（5）：787～791

[18] 吴晶晶， 荆继武，聂晓峰，王平建，一种快速中文分词词典机制，中国科学院研究生院学报，2009，26（5）：703～710

[19] 付年钧， 彭昌水，王慰，中文分词技术及其实现，软件导刊，2011，10（1）：18～20

[20] 王思力，张华平，王斌，双数组 Trie 树算法优化及其应用研究，中文信息学报，2006，20（5）：24～30

[21] 崔尚森，冯博琴，最长前缀匹配查找的索引分离 trie 树结构及其算法，计算机工程与应用，2005，20：131～134

[22] 邢富坤，程东元，基于统计语言模型的英语易读性研究，解放军外国语学院学报，2010，33（6）：19～24

[23] 单煜翔， 陈谐，史永哲，刘加，基于扩展 N 元文法模型的快速语言模型预测算法，自动化学报，2012，38（10）：1618～1625

[24] 闫新娟，谭敏生，严亚周，吕明娥，基于隐马尔科夫模型和神经网络的入侵检测研究，计算机应用与软件，2012，29（2）：294～296

[25] 袁里驰，基于改进的隐马尔科夫模型的词性标注方法，中南大学学报（自然科学版），2012，43（8）：3053～3056

[26] 姜维，关毅，王晓龙，基于条件随机域的词性标注模型，计算机工程与应用，2006，21：13～16

[27] 苏祺，咎红英，胡景贺，项锟，词性标注对信息检索系统性能的影响，中文信息学报，2005，19（2）：58～65

[28] 胡春静，韩兆强，基于隐马尔可夫模型(HMM)的词性标注的应用研究，计算机工程与应用，2002，6：62～64

[29] 郑榕增，林世平，基于 Lucene 的中文倒排索引技术的研究，计算机技术与发展，2010，2（3）：391～393

[30] 邓攀，刘功申，一种高效的倒排索引存储结构，计算机工程与应用，2008，44（31）：149～152

[31] 王冬，左万利，赫枫龄，彭涛， 张长利，一种增量倒排索引结构的设计与实现，吉林大学学报（理学版），2007，45（6）：393～401

[32] 郭涛，曲宝胜，郭勇，自然语言处理中的模型，电脑学习，2011，2：113～117

[33] 姜华，韩安琪，王美佳，王峥，吴雲玲，基于改进编辑距离的字符串相似度求解算法，计算机工程，2014，40（1）：222～227

[34] 刁兴春，谭明超，曹建军，一种融合多种编辑距离的字符串相似度计算方法，计算机应用研究，2010，27（12）：4523～4525

[35] 姚建民， 周明，赵铁军，李生，基于句子相似度的机器翻译评价方法及其有效性分析，计算机研究与发展，2004，41（7）：1258～1264

[36] 王开云，孔思淇，付云生，潘泽友，马卫东，赵强，两种基于双向比较的最长公共子串算法，计算机研究与发展，2013，50（11）：2444～2454

[37] 张毅超，车玫，马骏，求最长公共子串问题的算法分析，计算机仿真，2007，24（12）：97～100

[38] 孔德廷，伍守豪，金涛，张义德，维特比均衡算法，通讯技术，2010，9（43）：27～29

[39] 周鑫，基于 EM 算法的 G0 分布参数最大似然估计，电子学报，2013，41（1）：178～184

[40] 张洁，朱莉娟，贪心算法与动态规划的比较，新乡师范高等专科学校，2005，19（5）：18～20

发表论文和参加科研情况

无

致 谢

感谢宫秀军老师的指导和帮助，在选题和论文编写上都给予了很大的支持和指导，并提出了修改意见。

同时感谢和我一起工作的同事给我的帮助，让我顺利地完成这个研究工作。

感谢天津大学的所有老师，在我三年的学习生活中得到了它们的辛勤培养。

求学路上，同学、家人的关心、理解、鼓励和帮助，奠定了我完成本文的基础和信心，本论文的完成同样凝聚着他（她）们的心血，在此表示由衷的谢意。

最后，我要向百忙之中抽时间对本文进行审阅，评议和参与本人论文答辩的各位老师表示感谢。