中国人民大学
RENMIN UNIVERSITY OF CHINA

论文编码：RUC-BK-080901-2013202490

中国人民大学本科学生毕业论文
（毕业设计）

面向 Helpal 移动群智服务的搜索技术研究及 web 前端工程实现

作　　者：陈波

学　　院：信息学院

专　　业：计算机科学与技术

年　　级：2013 级

学　　号：2013202490

指导教师：陈红

论文成绩：90

日　　期：2017.5.9

学位论文原创性声明

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包括任何其他个人或集体已经发表或撰写的成果作品。本人完全意识到本声明的法律后果由本人承担。

作者签名：陈波

2017 年 5 月 9 日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保障、使用学位论文的规定，同意学校保留并向有关学位论文管理部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权优秀学士论文评选机构将本学位论文的全部或部分内容编入有关数据进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本学位论文属于 1、保密□，在 10 年解密后适用本授权书

2、不保密□。

（请在以上相应方框内打“√”）

作者签名：陈波

2017 年 5 月 9 日

导师签名：陈红

2017 年 5 月 9 日

中国人民大学本科生毕业论文简表

项目	内容
选题背景和意义	现代社会，随着移动设备的不断普及，使得基于位置的服务(LBS)成为引人注目的范式。移动设备的便捷性使得移动用户在享受着这些服务的同时，也有意愿作为服务的提供者。同时，传统众包系统的不足，使得其只能满足长尾理论的主流需求。远远不能满足新时代移动用户的多样需求以及对便捷性与即时性的要求。在这样的背景下，我们提出了一个基于位置的服务平台-Helpal，使用户能够即时的享受多样的服务和需求，
主要挑战/实际需求	Helpal 是众包与基于位置服务相结合，通过智能搜索技术实现移动个体的专业化服务平台。该平台的核心还是一个搜索服务类型的搜索系统 ，对于用户输入的服务类型，系统如何能够最快，最准确地将用户周围满足其需求的服务者提供给他，这也是系统的关键点和难点。
从实际挑战中抽象出来的研究难题/实现新需求的关键之处	与传统的搜索引擎不同，Helpal 平台收集的不是网页信息而是闲置的可支配的用户劳动力资源。所以不同于一般的网页文本检索功能，Helpal 本质上说是超短类型文本匹配，所以更注重的是对服务类型关键字间的语义分析，匹配，评分机制。此外，由于 Helpal 是移动服务平台，搜索系统还必须考虑用户之间的位置评分以及用户的信誉评分等等。
解决方案主要思路或特点	对于 Helpal 的关键搜索算法，解决问题的核心在于搜索系统评分公式的提出。所以，本文主要分成文本相似度评分和空间相似度评分两个方面来阐明:文本相似度上，文章会介绍相关的模型并分析其优劣势，最终基于 Google-word2vec 提出文本相似度匹配算法；空间相似度上，分析 Helpal 平台系统

	的需求，提出空间距离模型评分机制。
实验所基于的环境、平台、系统，比较的对象，自己在实验中所做的工作，以及实验结果	Helpal web 平台的实验测试工程是基于 windows 的 Tomcat 服务器，代码运行环境为 java-1.8。我所做的主要工作就是平台关键搜索搜索技术的研究以及算法代码的实现。初步效果良好：对于文本语义相似度算法实现后，能够较为准确的识别两个关键字之间的语义联系。应用至平台上，对于用户输入的需求，系统也能够比较准确地返回满足需求的服务者 list。
主要相关工作有哪些，存在的主要问题，本文工作相对于相关工作的创新点	本文在正式提出 Helpal 平台的关键搜索算法之前，介绍了一些信息检索系统的模型及具体应用，发现传统信息检索系统更多的采用分词，计算词频等词性分析。所以，相对于一般文本信息检索系统本文最大的创新点就是采用了 word2vec 深度学习，更侧重了关键字的语义分析。
总结	本文主要研究了 Helpal 移动群智服务平台的关键搜索技术，提出了合理的相关模型以及评分方法并加以实现，最后运行对比试验初步评估验证了基于 word2vec 词向量的文本相似度算法的影响因素，并带入 web 平台系统取得初步成效。

指导教师签名　　陈红

摘要

近年来，随着移动互联网的不断发展，基于移动设备的开发应用呈井喷式的冒出。多彩的移动服务应用便利着人们生活的同时，也极大地加深了人们的联系，这潜移默化地改变了人们的生活方式。随着人们对移动设备的依赖不断加深，人们希望移动设备能够随时随地的为他们提供想要的服务和帮助。然而传统众包只是企业或个人通过互联网将任务发布在大众网络的单向服务行为，而且它只能满足长尾理论中的主流需求，对当代移动用户多样需求以及用户其便捷性及即时性的要求远远不能满足。

在这样的大背景下，我们提出了一个移动服务平台-Helpal。该平台在提供基于位置服务（LBS）的同时还拥有即时（instant）响应的特点。这充分弥补了传统众包的不足。在该平台，寻求帮助的人可以随时随地，即时的搜寻附近的服务；拥有某项技能的人可以无约束上架自己的技能去帮助他人并获取报酬。该平台的开发拥有着十分重要的意义和社会价值！

本文主要研究平台的关键搜索技术。平台搜索引擎如何能够准确而又快速的根据用户需求去搜寻与其相匹配的服务，这是一个极其重要而又不易解决的技术点。本文将严格遵循学术的思想，一步步的提出解决问题的模型。

首先，本文将介绍前人在信息检索领域的贡献，分析比较各个传统模型的优劣点；随后考察几个流行的信息检索算法应用-Google 的 simhash 算法，Google 的 word2vec 词向量工具法，百度 NLP-短文本相似度接口等。接着，基于上述的算法模型，本文将正式提出 Helpal 移动群智服务平台的关键搜索算法模型。并分成文本语义相似度与空间相似度两部分给出实现方法；最终达到根据键入的服务类型关键字，搜索系统能够准确地返回满足需求的 result-list 的工程效果。同时，本文的最后将给出 Helpal-web 前端工程框架简介。

Helpal 移动群智服务平台的开发及其关键搜索技术的研究具有十分重要的意义，在实际应用方面：为移动服务应用的开发提供了新的思想和方向；研究领域方面可以促进智能搜索匹配技术的发展！

关键词：基于位置服务，移动群智服务关键搜索技术，信息检索，实体匹配，语义分析

Abstract

In recent years, with the development of mobile Internet, mobile devices and their development appear rapidly. A variety of mobile service applications not only facilitate people's lives, but also greatly enhance the people's contacts, which changes in people's way of life in a big way. As people's dependence on mobile devices continues to deepen, people want mobile devices to provide them with the services and help them anytime, anywhere.

In this context, we put forward a project - Helpal. The Helpal system provides a mobile service platform for people. In this platform, people who seek help can search for nearby services anytime, anywhere; people who have a certain skill can help others and get paid by shairing their skills. The development of the platform has a very important significance and social value.

Searching for matching services reflects the user needs accurately and quickly by using the platform search engine, which is an important technical point. In this paper, we strictly follow the scientific thought and solve the problems in this model step by step.

First of all, we introduce the predecessors'contributions in the field of information retrieval. Then we analyze and compare the advantages and disadvantages of various traditional models; we examine several popular information retrieval algorithm application - Google's simhash algorithm, Google's word2vec word vector tool, Baidu NLP - short text similarity interface and so on. Based on the above algorithm models, we formally propose the key search algorithm model of Helpal mobile service platform. It is divided into two parts: the semantic similarity and the spatial similarity of the text. Finally, the search system can accurately return the result of the result-list which satisfies the demand. At the same time, we give a brief introduction to the Helpal-web front-end project framework.

The development of the Helpal Mobile Service Platform and the research of its key search technology are of great significance. In the practical application, it provides a new idea and direction for the development of mobile service applications. The research field can promote the development of the intelligent search matching technology.

Key words: based location service, mobile key search technology, information retrieval, entity -matching, semantic analysis

目录

图片目录

1 绪论

1.1 研究背景

2016 年 9 月，支付宝提出了一个实验项目，它是一个移动设备 app：Alipay-Everywhere，中文名称：到位。支付宝解释称这是一个计划让周围人都变成“ATM”的一个应用。用户可以将自己的需求发布在应用内，并设置一定的支付金额，系统经过复杂的运算和搜索匹配，会以最快的速度准确地使服务双方找到彼此，以促进交易的达成。这一项目的出现就充分体现了现代社会已掀起基于位置服务（LBS）应用的开发热潮！

现代社会，随着移动设备的不断普及，使得基于位置的服务(Location Based Service,LBS)成为引人注目的范式，人们逐渐深刻的体会到，通过移动设备，信息技术可以为我们提供更多的服务。人们在面临自己难以处理的问题时，往往想的是寻找专业人士进行求助，以求最大效率解决问题。这就体现出了众包(crowd of sourcing)技术的思想以及优越性。众包是一种帮助人们快速获得大量信息，解决大规模问题的有效方案。然而传统的众包服务只是企业或个人通过互联网将任务发布在大众网络的单向服务行为。它单单从互联网获取信息并且不能即时对用户发布的需求给予响应，服务效率较低，而且它只能满足长尾理论中的主流需求，对当代移动用户多样需求以及用户其便捷性及即时性的要求远远不能满足。但是，移动设备的普及使得现代众包服务的数据收集的过程不再仅仅依赖现有的互联网上的信息，移动设备端的信息传播与收集，使得众包能够即时的，更加高效的帮助人们分析解决问题，这种经济和高效的架构在研究和工业领域都得到了广泛的关注。

所以，在这样的背景趋势下，我们提出了这样的一个移动群智服务平台- Helpal，该平台是新时代众包(crowd of sourcing)技术与基于位置服务(Location Based Service,LBS)相结合，通过智能搜索技术实现移动个体的专业化服务平台。它拥有即时相应以及 lbs 的特点，充分弥补了传统众包系统的不足。

1.2 研究意义

当代社会，越来越多的用户愿意提供自身具有的某项技能来服务周围有需求的人。于此同时，用户的需求也越来越多，例如，“搜索附近会修电脑的用户、查找可以帮忙取快递的同学、推荐健身经验丰富的同事等等”。人们迫切需要着这样一个平台去分享自己的知识技能服务他人，同时为自己赚得报酬。本文提出的 Helpal 就是这样的一个服务平台。如图 1.1，它将众包(crowd of sourcing)技术和基于位置的服务（LBS）相结合，并通过智能搜索技术实现移动服务，从而达到设备计算能力与人类专业知识的相结合。不同于网页搜索引擎，Helpal 平台收集的不是网页信息而是闲置的可支配的用户劳动力资源。

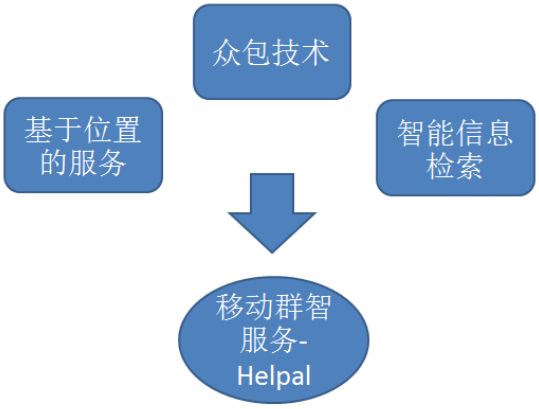

图 1.1 Helpal 平台

Helpal 平台计划包括用户信息聚集(Profile Aggregation)、信息索引建立(Information Indexing)、查询匹配(Query Match)、任务推荐(Task Recommendation)、结果评估(Result Evaluation)和用户交互界面(User Interface)六个部分。平台的开发也是有着很多的挑战：拿查询匹配技术来说，不同于传统的信息检索技术，Helpal 的移动群智服务中的查询对象往往是具有移动性的用户实体，通过用户键入的服务类型对平台内登入的用户服务类型进行超短文本的匹配查询，需要考虑到空间位置因素的同时还要考虑到服务类型跟用户需求是否一致或相似。而想要达到这样的效果，就必须考虑到语义分析，这就涉及到了人工智能，机器学习以及自然语言处理的知识。机器学习(Machine Learning,ML)[1]指的是计算机通过模仿人类的学习行为来获得新的知识或技能。简而言之，就是通过经验来改变计算机的性能。它是一门极其复杂的交叉学科，也是人工智能的核心。自然语言处理（natural language processing，NLP）[2]也是计算机科学人工智能领

域的一个火热的研究方向，主要研究人与计算机通过自然语言通信的各种算法和模型，包含大量的语言学和数学模型。

本文需要解决的开发挑战就是查询匹配技术的研究和实施模型。计划基于文本相似度与空间相似度的提出自己的匹配算法模型。实际应用方面可以完善 Helpal 移动群智搜索系统；研究领域方面可以促进智能搜索匹配技术的发展。

1.3 研究内容

现在的信息检索技术已经进入了人工智能的时代，传统的匹配算法已经完全不能满足人们的需求。随着机器学习以及自然语言处理（natural language processing，NLP）在研究领域的逐渐热门，信息检索的语义分析成为了所有信息检索算法研究者的需要攻克的难题，同时在该领域也取得了不小的进展。

总的来说，本文的关键搜索算法研究工作主要分为四个方面：

1. 阐述前人在该领域的贡献与成果，主要是介绍信息检索的几种最传统，最原始的模型：布尔模型（The Boolean model），区域模型（Region models），空间向量模型（The vector space model）等，以论述信息检索的基本理论框架与思想；

2. 考察现在比较流行的几个的信息检索系统与算法：google simhash 算法，google word2vec 算法，百度 NLP-短文本相似度接口等。阐述其原理并比较优劣点。

3. 深度分析 Helpal 平台的匹配模式与需求，根据应用的实际情况同时综合前人的研究提出基于 google word2vec 词向量工具的文本相似度的语义匹配模型以及基于空间距离模型的空间相似度匹配模型，结合文本相似度与空间相似度匹配技术提出 Helpal 移动群智服务平台的关键搜索技术算法模型，并给出模型的评分和结果返回机制。

4. 用代码初步实现算法后，给出评测指标。随后给定测试集以测试移动服务平台搜索算法的正确率及可靠性，给出测试结果及其分析，得出初步结论。

最后初步讨论并实现关键搜索技术后，本文将简要介绍 Helpal-web 前端框架的组织架构，功能及页面效果。提出和解决前端开发需要面临的技术难点。随后给出全文总结及展望。

1.4 论文的组织

本文计划用六个章节来阐述论文的研究内容：

第一章 绪论：该部分主要介绍了 Helpal 移动群智服务平台的开发背景与开发意义，简要说明了平台的预计架构及可能面临的挑战，并以此引出本文的讨论内容。

第二章 相关算法模型综述与分析：该部分讨论论述了相关信息检索的模型算法，并分析了其优缺点。综合各个模型的特征提出适用于 Helpal 移动群智服务平台系统的预想匹配算法。

第三章 Helpal 移动群智服务平台需求分析：该部分，本文通过一个典型的例子深度剖析了 Helpal 平台的需求，为下文关键搜索技术模型的提出奠定了基础。

第四章 Helpal 移动群智服务平台搜索算法模型提出：在该部分，正式提出了关键搜索技术的算法模型。随后在文本相似度评分与空间相似度评分两部分详尽的介绍了算法模型的实现原理和方法，并给出了初步的实现效果。

第五章 实验评估：在本章节中，本文提出了文本匹配度算法的评估指标，分析了可能影响算法效率的几个因素，并围绕分析结果和评估指标设计了几组简单的验证实验。随后分析实验结果，给出初步结论。

第六章 Helpal web 前端工程概述：在本章节，本文介绍了 Helpal-web 前端工程的组织架构设计，实际页面效果及功能演示。随后介绍了开发过程中遇到的几个技术难点，并给出解决方案

第七章 总结与展望：最后，总结了全文的主要内容，同时提出了不足的地方，并展望未来移动服务应用的发展。

2 相关算法模型综述与分析

2.1 传统信息检索模型[3]

传统信息检索模型包括布尔模型（The Boolean model），区域模型（Region models），空间向量模型（The vector space model）等。这些模型都有着自己的特征。

2.1.1 信息检索系统的概念及流程

信息检索系统是一个软件程序，它存储管理着文档上的信息，其中大多数是文本文档，但有可能也会是多媒体。系统支持用户检索他们需要的信息，但系统并不会明确的返回信息或答案，而是会返回某些或许包含用户所需信息的文档，称之为相关文档(relevant document).完美的信息检索系统返回的全是相关文档，没有不相关的文档，但这种系统是不存在的。因为检索语句的不完整，而且最重要的是，是否相关完全取决于用户的主观意识—即使两个人输入同样的检索语句，但两人最终采纳的结果列表却极有可能不同。

信息检索系统一般来说都包含三个过程：用户检索信息的表示，文档信息的表示以及两者的比较匹配。如图 2.1：

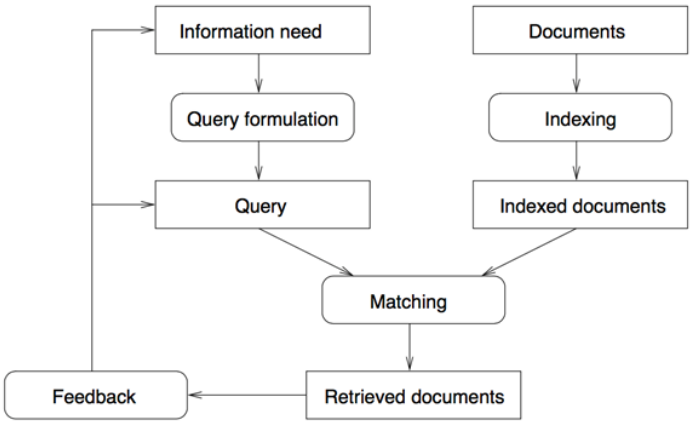

图 2.1 信息检索系统示例

文档信息的表示过程也可以叫做索引的建立过程(index process)，索引建立的算法是系统

决定的，终端用户参与不了的过程。索引建立有可能影响文档的实际存储。但通常情况下，索引只会存储文档的部分内容，例如文档的标题与摘要，以及文档的位置信息等，利用适当的索引建立算法可以极大地提升检索效率。

用户信息的表示过程也称为查询定制过程(query formulation process).总的来看，查询定制是一种系统和用户的交互过程，通过规范化用户的查询语句，目的是更好的理解用户的信息需要以及更合适的查询条件。

两者之间的比较称为匹配过程(matching process)，匹配的结果是文档的排名(ranked list)。排名检索奖最相关的文档排在前列，最小化用户阅读文档的时间。简单而又高效的排名算法不仅考虑术语在文档中出现的频率，也有其它信息的统计。排名算法背后的理论是信息检索的重要部分，这也是本文研究的重点内容。

2.1.2 布尔模型（The Boolean model）

布尔模型基于 boolean 逻辑词以及传统几何理论，它规定文档索引和用户的查询语句都用一系列的关键词术语来表示，所以布尔模型的检索是否成功完全取决于文档索引是否包含查询关键字。记 query 为 Q,若 Q 形似 A AND B(A，B 为查询关键词)，则根据布尔模型的要求，系统只会检索所有包含关键词 A 和 B 的文档集，然后返回给用户。

布尔模型的显著优点是检索系统能清楚的知道哪些文档能被索引出来并且该模型很容易实施，很符合直观概念；

但是，它最主要的缺陷在于不能形成一个 ranked list，若用户查询为 Q =A AND B AND C;文档 1 满足：A AND B AND D;文档 2 满足：A AND E AND F，很明显的，这两个文档都不会被检索到，但是文档 1 比文档 2 要有用或者说更能满足用户的需要，若没有完全满足要求的文档的话文档 1 应该先于文档 2 返回。但是这个 model 并不能意识到这种差别。

2.1.3 区域模型（The Region model）

区域模型可以说是布尔模型的拓展，与布尔模型不同：区域模型可以用文本文档的任意部分来表示该文档，也就是用文档内连续的文本字符串来表示文档；与布尔模型的用关键词来表示

索引文档不同，区域模型往往用一段有开始位置与结束位置标志的区域来表示文档。与布尔模型的用 and or not 三个布尔逻辑操作符类似，区域模型至少用两个操作符 containing / contained by 来表示逻辑操作

与布尔模型类似，它拥有一个致命的缺点就是不能形成 ranked list，ranked list 对于一个检索系统来说是至关重要的。

2.1.4 空间向量模型（The vector space model）

空间向量模型是到目前为止在信息检索领域最被人承认，同时也是应用最广泛的一个信息检索模型。本文的搜索算法模型也是建立在空间向量模型的基础上提出的。

Peter Luhn 率先提出了利用统计学的方法来搜索信息。他提出为了查询文档，要先准备一个与你需要查询的文档相似的规范文档作为代表，该文档与库存文档的相似程度可作为 ranked list 的标准。随后，他又提出了相似标准：给定的元素在两个代表文档中出现的频率越高，这俩个文档就越相似。又如上所说，信息检索的准备活动一般要先将用户查询语句定制化，库存文档索引化，定制化的查询语句就是用户需要文档的表示，库存文档索引就是库存文档的代表。将用户查询表示为$\vec{q}=(q_1, q_2, \cdots, q_m)$，$\vec{q}$的分量$q_k$(1<=k<=m)代表一个查询关键词，同理文档索引向量$\vec{d}=(d_1, d_2, \cdots d_m)$ 其分量代表同一类关键词，分量的取值只取 0 与 1，表示有或无此关键字。那么它们的相似度就可以用向量的内积来表示：$score(\vec{d}, \vec{q})=\sum_{k=1}^{m} d_k * q_k$

进一步拓展，Salton and McGill 建议将空间向量分布到高维欧几里得空间，其中每一个分量项也就是每一个关键字单独分配一个维度。图 2.2 是查询向量与索引向量被分配在被 social economic political 三个关键字维度分配的空间中，

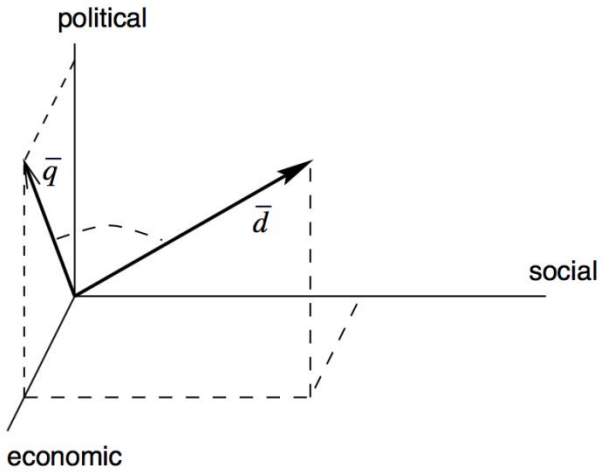

图 2.2 查询索引向量在高维空间中的表示

这样就可以用两个向量的夹角余弦作为关键字相似度得分，余弦公式计算相似度评分如公式(2-1)：

$$score(\vec{d}, \vec{q}) = \sum_{k=1}^{m} d_k * q_k \Big/ \sqrt{\sum_{k=1}^{m} d_k^2} * \sqrt{\sum_{k=1}^{m} q_k^2} \qquad (2\text{-}1)$$

由评分公式可知，score 越接近于 1，表明查询相似度越高。

空间向量模型的最大优点就是它的评分机制，该模型可以根据用户查询与文档索引的相似度，按照向量的内积（或欧氏距离）给出一个相似度评分。信息检索系统就可以按照评分从高到低返回给用户一个 top-k 的 ranked-list，满足信息检索系统基本要求。本文的关键搜索技术算法模型的匹配机制也是在空间向量模型的基础上进行开发的。

2.2 Google simhash 算法模型[4]

简要介绍完最基础的信息检索模型后，我们来考察几个流行的信息检索算法模型。

simhash 是 Google 公司开发出的一种用于网页文本去重的算法。随着互联网产业的蓬勃发展，信息爆炸的时代已经来临。网上充斥着大量的重复的信息。而过滤掉这些重复的信息，是实现高效，准确的信息检索系统的前提条件，在这样的大前提下，Google 公司开发出了 simhash 文本相似度比较算法。

传统的 hash[5]算法能够把一段文本随机地映射成一个签名值（hash 算法原理不再赘述），类似于伪随机数算法。如果在同一个 hash 算法下，两段文本的签名值相同，那么只能保证这两段文本有一定的概率相同；若签名值不相同，那么这两端文本一定不相同。那么如果想用签名值相似度代表文本相似度，就必须设计出一个算法，满足两段相似的文本有着相似的签名值，simhash 算法就实现了这一点。

Simhash 算法的输入值是一个向量，该向量是文档的特征值（feature）集合，特征值的选取看实际情况，能唯一的代表该文档即可，每个特征值都被赋予一个权重（weight），一般为特征值在文档中出现的频率；输出值就是用于比较的签名值，一般为二进制位。简单的算法流程如下：

1. 初始化 n 维向量 S，使其 n 位分量全为 0；
2. 对于文档向量的每一个特征值，用 hash 算法将其转换为 n 位二进制位的签名值 s。对 s 的每一位做这种处理：若该位为 1，则该位变成该特征值的权重；若该位为 0，则该位变为改特征值的权重*（-1）；
3. 将全部特征值的 n 位签名值相应的位置相加，对于累加后的每一位，如果结果>0,则 S 的相应位变为 1；若结果<=0，则 S 的相应位变为 0；
4. 输出 S 做为该文档的签名值。

如图 2.3，是 simhash 算法的图示流程：

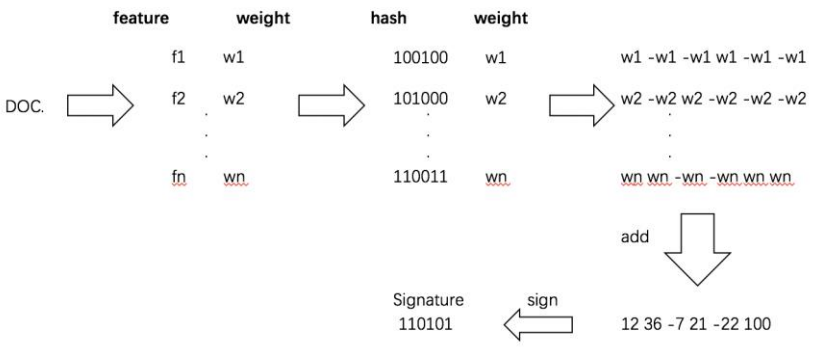

图 2.3 simhash 流程图

得到每个文档的签名值后，如何判断两文档的距离呢？假设若文档 a 的特征值为a_1:110001，文档 b 的特征值为b_1:100111，则a_1与b_1的海明距离（Hamming distance）即为文档 a 与文档 b 的距离。如上例所示，文档 a 与文档 b 的距离为 3。那么距离为多少可以判断两文档是否相似呢？这就要看实际情况与实验，根据你的实际情况，设定一个相似底线距离 n，也是就所谓的 baseline。当两文档签名值的距离小于等于 baseline 时，可以认定两文档相似或重复；反之，两文档不相似。

simhash 算法的主要优点就是将庞大，复杂以及不好进行相似度比较的文档以及网页信息转换成为 n 位二进制签名值。对于检索系统来说，方便存储大规模的文档特征索引信息；对于文档本身来说，方便相互之间的相似度的比较。所以被 Google 广泛用于网页查重。

相对于本工程的查询需求来说，simhash 算法的缺点也是明显的：正如上面所说，simhash 被广泛用于网页查重，输入的是某个网页若干个关键词组成的特征值向量。但实际运用到我们的工程来说，工程需要的是搜索用户服务类型的搜索引擎，本质上说是超短类型文本匹配，用户输入的只是服务类型的关键字。因为 Simhash 算法没有上下文的语义分析，所以运用到短文本相似度匹配时效果不好，这也成为 simhash 最大的制约点。

2.3 Google word2vector 词向量模型[6]

Word2vec 是 Google 公司基于神经网络语言模型技术在文本挖掘领域的一个开发项目。它是深度学习(Deep Learning)的一种实现；它是一种将词表转换为实数向量的一种有效的应用工具。Word2vec 采用 CBOW(Continuous Bag-Of-Words，连续词袋模型)以及传统的 Skip-Gram 模型计算出词汇的向量表示，并用于后续深入的自然语言处理研究。

Word2vec 工具将分词完毕的文本语料库作为输入，经过模型训练后，把产生的词向量文本作为输出。它首先通过从训练文本数据构建出一个个词汇，然后学习这些词汇的向量表示。通过这样的方式进行大量的训练，word2vec 可以将文本内容转为 n 维的词向量，并通过词向量的运算空间上的相似度（计算向量的欧氏距离或者向量夹角的余弦值）代表文本内容语义上的相似度。Word2vec 输出的基于语料库训练的词向量文件还可以做多种的 NLP 分析应用，这充分体现出了 word2vec 的研究意义与深远价值！

Word2vec 有着两种最主要的模型：CBOW(Continuous Bag-of-Words Model)和 Skip-gram(Continuous Skip-gram Model)[7][8][9]，这两个模型都是神经网络语言模型根据实际情况的改进与应用。对于每种模型有两种实现框架：HS(Hierarchical Softmax)[10]和 NS(Negative Sampling)[9]。简要介绍下 word2vec 的模型及实现的方法和原理。

2.3.1 词向量的表示

自然语言处理的问题归根究底就是机器学习的问题，所以，我们要做的第一步就是把这些文字符号数字化，词向量就应运而生。

word2vec 运用的是 Distributed Representation,原理见文献[13]这种低维实数向量的表示方法，一般为 100 维左右，词向量一般是这个样子[0.22 0.79 0.56 -0.12 0.89 …]；但向量的表示不是唯一的，这是因为 word2vec 在不同的语料库和不同的训练方法下，一个词会有不同的 Distributed Representation 向量表示。这种向量表示法的主要作用就是让语义相近的词，在距离上更为接近了。

2.3.2 传统的统计学语言模型

word2vec 模型是基于神经网络语言模型的，为了更有助于理解模型的原理与公式，首先介绍下传统的语言模型。

通常来说，统计语言模型是用来估计一个句子出现概率的概率模型。该概率模型是基于你提供的语料库所构建的，语料库不同，所构建出的概率模型也就不同。记一个句子为 W，

$W=w^T=(w1,w2,w3,\cdots,wn)$，表示句子 W 是由词 w1,w2,…wn 顺序排列组成的。则联合概率

$P(W)=P(w^T)=P(w1,w2,w3,\cdots,wn)$即为句子 W 的概率。由概率分布的 Bayes 公式分解可得公式(2-2)。

$$P(w^T)=P(w1)*P(w2|w1)*P(w3|w^2)*\ldots*P(wn|w^{n-1})=\prod_{i=1}^{i=n}P(wi|context(xi)) \quad (2\text{-}2)$$

其中 context(xi)代表词 xi 的上下文即词 xi 的语义。根据概率公式,可以将看作 P(wn|w^{n-1})为语言模型在当前语料库的参数。也就是说如果求得这些参数则任意给定一个句子 W，它的概率也很快就能算出来。但算出这些参数是及其困难的！假设语料库的词典大小为 N，给定一个句子的长度为 T，则参数的数量级就达到了 TN^T。所以，具体在应用中，有很多优化的算法，word2vec 的模型就是神经网络语言模型的一个具体的应用优化例子。

2.3.3 模型介绍-CBOW 和 Skip-Gram

CBOW(Continuous Bag-of-Words Model):

CBOW 模型如图 2.4

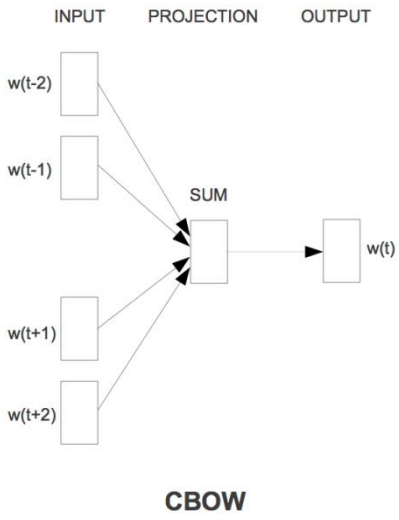

图 2.4 CBOW 模型示意图

该训练模型是 Bengio 在 2003 年提出的神经网络语言模型的优化应用版[10]，由图可以轻易看出，CBOW 模型是根据词的上下文向量来预测词向量，word2vec 规定对于 sample：(context(w),w)来说，context(w)是取词 w 的前后 n 个词（图中是取前后 2 个词，共四个词）作为它的上下文，来预测词 w 出现的概率。它包含三个层：输入层(input)，投影层(projection)以及输出层(output)。

输入层输入语料库中的词对应的词向量：即图中的 w(t-2)…w(t+2)；

投影层把上下文向量做了一次求和运算；

输出层为相应的词 w。

该模型的优化目标函数形如公式(2-3)

$$\mathcal{F}=\sum_{w\in D}\log p(w|context(w))=\mathcal{F}(w,context(w),\theta) \qquad (2\text{-}3)$$

（其中 D 表示语料库）即可以看成$^{w,context(w),\theta}$的函数，其中$^{\theta}$是待定参数的集合。该模型通过不断对 sample 进行训练，计算和反馈，从而动态更新$^{\theta}$里的参数值使目标函数达到最优化，语料库训练完毕后输出的二进制文件就是我们所需要的词向量文件。所以文件中词向量的代表性与准确率完全取决于语料库，语料库越庞大，越具有代表性，那么词向量运算得出的相似度就越能代表现实中两个关键词的语义相似度。

Skip-Gram(Continuous Skip-gram Model)

Skip-Gram 模型的如图 2.5

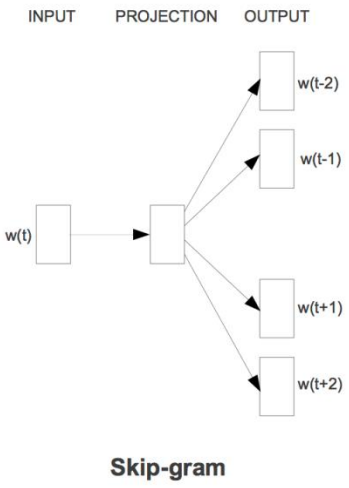

图 2. 5 Skip-Gram 模型示意图

它也有三个层：

不同的点是：它是通过当前词 w 去预测以当前词为中心的上下文。该训练模型的优化目标函数形如公式(4-4)

$$\mu=\sum_{w\in D}\log p(context(w)|w)=\mu(w,context(w),\theta) \qquad (2\text{-}4)$$

训练方法与 CBOW 训练模型类似，不再赘述。

HS(Hierarchical Softmax)和 NS(Negative Sampling)：是实现 CBOW 与 Skip-Gram 训练模型的两种训练框架，也可以理解为 word2vec 中提高性能的关键方法。HS 用的是 Huffman 树，树的叶子节点数目即为语料库中词的数目，利用树的结构不断二分类而提高模型性能；而 NS 采用的是随机负采样的方式来提高模型性能，具体细节参考相关文献。

word2vec 词向量工具的优点就是就比较能够满足 Helpal 平台搜索匹配系统的要求，因为移动群智平台 Helpal 搜索系统是一种搜索用户服务类型的搜索引擎，本质上说是超短类型文本匹配。我们要达到的效果只是用户输入少量关键字（通常为 1 个），去查询到相关的服务。所以，搜索引擎不采用传统的靠文档分词匹配的搜索模式，而是更注重关键词之间语义的相似度，根据这个得出评分。所以本文将主要以 word2vec 词向量模型为工具，研究，挖掘词向量的语义匹配模式。

2.4 百度自然语言处理 api[11]

Google 公司一样，百度搜索引擎作为国内最流行的搜索引擎之一，百度公司自然也投入了大量的研究工作在当下最热门的自然语言处理（natural language processing，NLP）技术以及机器学习(Machine Learning,ML)技术，百度 NLP 部门也由此诞生。

百度自然语言处理部门开展包括机器学习,数据挖掘以及自然语言处理等人工智能领域的产品与技术的研究工作，为百度其他部门的运作提供技术支持。在本文所关心的自然语言处理方面，查询了官方相关文档后，发现百度官方提供了包括：分词，词性标注，词向量表示，中文 DNN 语言模型，短文本相似度接口，评论观点抽取等自然语言处理的 api 接口。因为 Helpal 的搜索引擎系统需要的是超短文本的相似度匹配，所以本文重点研究了百度 NLP-短文本相似度匹配 api[12]。

查询了百度 NLP-短文本相似度匹配 api 的相关技术文档，发现请求参数中最重要的就是两个 terms_sequences 参数，也就是需要比较的俩段短文本字符串。如果请求成功，那么返回字符串中的参数 score，就是俩段短文本的相似度评分。一个成功的请求示例如图 2.6 所示。

请求

```
{
  "input":
    {
       "qslots":[{"terms_sequence":"你好百度"
", "type":0, "items":[]}],
       "tslots":[{"terms_sequence":"你好世界"
", "type":0, "items":[]}],
       "type":0
    }
}
```

返回：

```
{
  "output":
    {
      "score":0.758419,
      "error":0,
      "type":0,
      "error_note":"",
      "items":[]
    }
}
```

图 2.6 百度 NLP-短文本相似度 api 示例

优点：首先从查询需求来看，百度 NLP-短文本相似度 api 正好满足 Helpal 平台的超短文本匹配的查询需求；从技术层面上看，根据相关技术文档：该 api 应用主要有了以下 3 点技术优点：

1.深度学习技术：在神经网络语言模型技术基础上，从单一词语义到短文本语义进行组合建模，效果更好更突出。在这点上与 Google 的 word2vec 模型类似，都是利用了最先进的神经网络语言模型并进行训练建模；

2.实现了语义匹配：与 word2vec 一样，该接口也是利用了词向量技术去进行语义匹配，但关键词匹配不到时会自动进行同义词，近义词的匹配。十分高效；

3.用户反馈机制：该接口运用的百度用户反馈数据进行再次训练，用户的反馈结果指导系统进行更优质的算法改进，返回更加准确的，令人满意的结果，在我看来，这是它由于现有主流算法的最重要的一点。因为现有的主流算法都没有使用用户的反馈数据去提升算法性能。

但是，百度 NLP 不足的地方也是有的。因为该 api 是百度 NLP 官方提供给开发者使用的，使用它需要申请开发权限。并且最重要的一点是，该工程并没有公开源码，相关论文支撑等。而且该 api 每天有使用次数上限，一天最多请求 1000 次。所以综上所述，并不是很适合真正实际应用到 Helpal 平台上。不过百度 NLP 官方的训练语料库十分庞大，还有用户反馈数据处理去优化算法效果和性能，所以匹配效果非常好，故平台实验期间为了更好的比较和实验效果，可以做测试用。在下文的算法评估测试环节，会应用到此 api 接口。

2.5 本章小结

信息检索技术发展至今已经取得了不少的成果，有着各种各样的算法模型。在本章节，首先介绍了信息检索系统的基本流程概念以及几个具有代表性的检索模型。随后考察了几个近期比较流行的算法模型框架-Google 的 simhash 算法,Google 的 word2vec 词向量工具法,百度 NLP-

短文本相似度接口等，分析比较了这些模型的优劣点，为下文 Helpal 移动群智服务关键搜索算法模型的提出奠定了基础。

3 Helpal 移动群智服务平台需求分析

Helpal 是一个移动群智感知的服务平台，它利用众包和移动群智服务的思想，为社会闲置的可支配的用户劳动力资源提供了一个体现了其价值的平台：如果一个服务者（姑且我们把上架服务的人称为服务者）拥有维修电脑的技能，并且他想利用闲余时间靠这个技能去服务大众，顺便为自己赚点外快，那么他就可以在 Helpal 平台上上架他的服务类型-“电脑维修”，系统会利用移动端的定位功能在后台存储他的实时位置并不断更新。上架服务这一方面就体现出来了 Helpal 移动群智平台人性化，大众化的特点：服务者不需要有任何的专业技能证书，只要他具有足够的专业能力并且有空余时间，都可以在该平台上上架自己的服务；如果此时有一个用户的电脑坏了需要维修，他就可以在 Helpal 平台的搜索引擎上进行搜索，键入关键字“电脑维修”。这时候，问题的重点来了，Helpal 平台的移动搜索算法如何返回用户想要的结果呢？首先，搜索算法的第一步肯定是先根据用户输入的查询关键字去跟平台数据库的服务类型字段进行匹配，这就牵涉到文本相似度的匹配评分；文本匹配完毕后，若有 2 个服务者提供了相同或相似的排名，此时文本相似度得分相似，那么此时搜索算法该怎么进行排名（rank）呢？因为 Helpal 平台是一个移动群智服务平台，所以必须考虑位置的因素，这就涉及到空间相似度的匹配评分；到此，是不是只要考虑文本相似度和空间相似度就可以进行排名了呢？正如上面提到的那样，Helpal 平台人性化与大众化的特点从某种程度上来说也是缺点，因为上架服务的门槛过低，所以服务质量不能够保证，所以此时 Helpal 平台添加了以用户反馈为基础的“信誉评分机制”-每次用户享受完服务后，系统都会提示用户对该服务者进行打分做为其“信誉评分”。

进行了上述的平台需求分析后，Helpal 移动群智服务平台的关键搜索算法 rank 评分的主要影响因素已经确定：1.文本相似度评分；2.空间相似度评分；3.用户信誉评分。

4 Helpal 移动群智服务平台搜索算法模型提出

经过上面的系统需求深度剖析后，现在我们可以正式的提出关键搜索算法的模型了。设用户查询的关键字与平台系统数据库里的服务类型的文本匹配相似度得分为α，文本相似度评分占整体评分的权重为w_1；设用户地理位置与系统服务者地理的空间相似度评分为β，空间相似度评分占总体评分的权重为w_2；服务者的信誉评分为γ，服务者信誉评分所占总体评分的比例为w_3。则移动搜索算法的评分$score_{Helpal}$表示为公式(4-1)

$$score_{Helpal} = \alpha \times w_1 + \beta \times w_2 + \gamma \times w_3 \quad (4\text{-}1)$$

其中：α，β，$\gamma \in (0,1)$;

$w_1 + w_2 + w_3 = 1$；

$score_{Helpal} \in (0,1)$

由公式(4-1)可知，对于某个服务者来说，在考虑了其服务类型的评分，位置的评分以及信誉评分后的最终得分：$score_{Helpal}$越接近于 1，表示该服务者的与用户的查询需求越契合，则该服务者的 result-list 中的排名就越发靠前。事实上，搜索系统返回给用户的 top-k 排名列表（ranked-list）就是按照$score_{Helpal}$的大小为标准进行排名。

对于评分公式(4-1)中的待定参数 γ：用户的信誉评分来说，该项评分是用户反馈的结果，不涉及算法等技术问题，系统数据库会记录用户的每次评价，并给相应的服务者进行综合评分。当搜索系统需要该项评分时，直接从系统数据库中提取计算即可，所以在此不做多讨论。公式中其余的待定参数 α，β，w_1，w_2，w_3的确定方法本文将一一详尽说明。

4.1 文本相似度匹配算法

比较文本的相似度中最常见，也是最基本的是关键词匹配，例如[西餐 牛排]和[牛排 烤冷面]之间，如果以关键词个数计算，其相似性就是 1(牛排为同时出现的关键词)。这么看来问题似乎很简单就解决了，但是这样的相似性度量是不是好的?如[西餐 牛排]和[牛排 烤冷面]这个例子，相似性也可以是 2，因为如果我认为相关的关键词都记为匹配，即西餐也包括牛排，那么[西餐][牛排]=1，[牛排][牛排]=1，结果为 2。事实上，后一种考虑到相关语义的匹配算法才是真正的比较完善的匹配算法。

传统信息检索文本匹配算法的基本流程是这样的：首先对用户输入文本进行分词，关键字提取，然后根据训练模型，实际情况等给相应的分词或关键字添加权重（weight），最后运用搜索算法去文档库中进行搜索，评分，加权计算，最后进行排序-类似于百度搜索，Google 搜索那样的大众化的搜索引擎。与传统信息检索系统不同，Helpal 移动群智平台的搜索系统是一种搜索用户服务类型的搜索引擎，本质上说是超短类型文本匹配，所以我们更注重具体服务类型关键字的搜索，根据用户输入的服务类型关键字进行语义匹配评分。故关键字间的语义相似度匹配就成为了算法研究的重点！

正如相关模型综述所提到的那样，Google 的 word2vector 词向量模型恰好满足了我们的需求，利用分词工具对语料库进行分词，随后利用 word2vec 进行语料库的带入，训练，训练完毕后就产生的我们需要的词向量文件。其中的词向量间的空间相似度就可以代表对应词的语义相似度。

4.1.1 算法初步实现与效果

本小节，将具体介绍借助通过 word2vec 的两种主流训练模型（CBOW,Skip-Gram）训练出的词向量文件的应用例子。

a. Google word2vector

上面较为详尽说明了 Google 的 word2vector 词向量工具的原理模型及优点，那么实际情况下的语义匹配效果好不好呢？下面我们就来简单实现下。

为了检验 word2vec 词向量工具的效果，我从网上下载了大约 1G 的中文 wiki 百科作为语料

库，在输入 word2vec 模型训练之前，需要对文本进行分词处理。具体分词的方法网上有很多资源方法，例如 jieba 分词，java 的 ANSJ 分词工具等，在这我运用了张华平博士的 NLPIR 汉语分词系统[14]，具体的操作流程不在细说。

具体分完词后的文档大约有 500MB，打开后的效果如图 4.1：

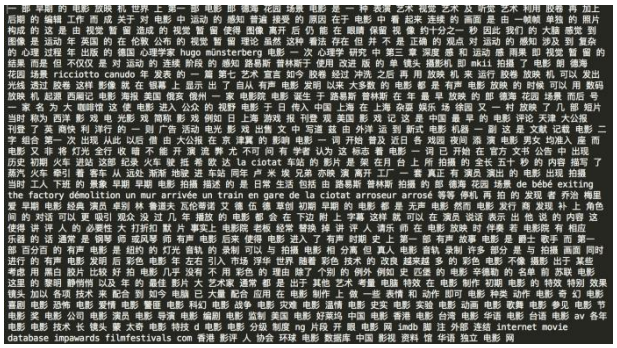

图 4.1 分词后的文档

接着上 Google word2vec 的官网下载 C 语言源码进行预编译：如图 4.2

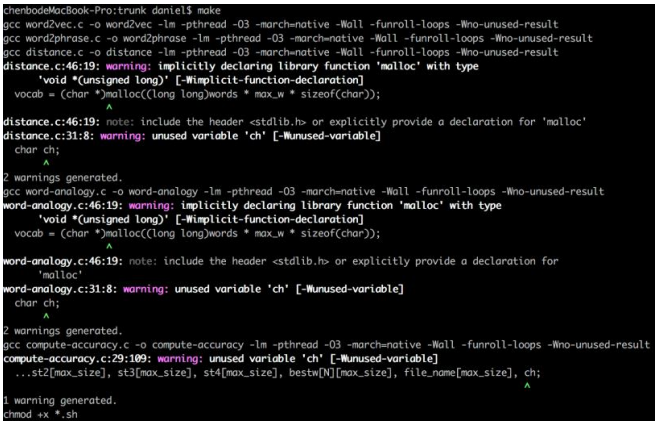

图 4.2 编译 word2vec 源码

将实现分好的词作为语料库输入进行 word2vec 的模型训练，命令如图 4.3

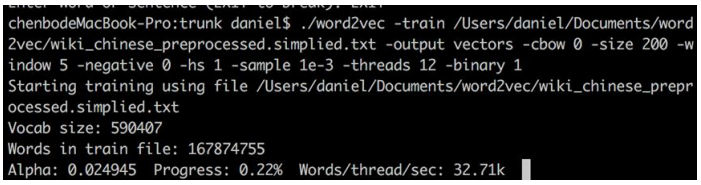

图 4.3 word2vec 模型训练

执行 word2vec 代码时需要注意的是，有几个重要的编译参数：

-train：输出词向量文件；

-output：指定输出的词向量文件的路径及名字；

-cbow：是否使用 CBOW 模型，0 表示不使用，默认使用 Skip-Gram 模型，Skip-Gram 模型训练会更慢一些，但是对低频词的效果好，CBOW 模型训练会更快一点；

-size：词向量的维度；

-window：表示上下文的范围，例如：-window 5 就代表取当前次的前五个与后 5 个词共 10 个词表示该词的上下文语义；

-negative，-hs：代表是否使用此方法，0 代表不使用；

-sample：代表采样的阈值，如果词的出现频率大于此值就会被输出至词向量文件；

-thread：并行的进程数；

-binary：是否采用二进制存储，默认采用；

在这里我使用了 Skip-Gram 模型，词向量的维度为 200 维，使用关键字的前后共 10 个词作为它的上下文语义，使用 HS(Hierarchical Softmax)训练框架，采用的阈值为 1e -3，程序运用的并行进程数为 12 个。

本文写了一个简单的 java 代码，加载训练好的词向量文件进行 NLP 处理的例子，找出与查询词语义相近（distance）的词组 list，输入查询关键字:“甜点“，输出效果如图 4.4

Word distance

图 4.4 word2vec-distance 演示

右边的 distance 就是训练后语料库里的词与查询词之间的语义匹配相似度，按从高到底排列。由图可知，关键词之间的语义匹配效果良好。

令人惊奇的是:运用 word2vec 工具训练出来的词向量还可以展现语言学的规律：

向量运算：vector(‘paris’)-vector(‘france’)+vector(‘italy’)得出的向量跟 vector(‘rome’)非常相似；

向量运算：vector(‘king’)-vector(‘man’)+vector(‘woman’)得出的向量跟 vector(‘queen’)非常相似；

这充分的体现了：语料训练后产生的词向量工具在向量空间上的关系能够较为精确的代表了关键字之间的语义关系。

具体应用到 Helpal 移动群智服务平台来说，平台需要的是服务类型搜索引擎即对用户查询服务类型的关键字与平台系统上架的服务类型关键字进行语义匹配评分，然后按评分高低进行排名并反馈给用户。这就需要根据基于 word2vec 训练后的词向量文件进行 NLP 开发。本工程开发了一个专门计算两个关键字之间的语义评分的 java-api 接口-similarity()，它用两个主要的函数：

1. loadmodel()：读取训练好的词向量文件读取，并按照 word-vector 键值对的格式将其存至 java 的 hashmap 数据结构中，以供查询用；
2. similartity(query1,query2):将俩个关键词 query1，query2 带入 hashmap 进行查找，找到后取出它们的 vector 并利用向量夹角的余弦公式计算 vector 之间的余弦值作为文本语义相似度评分返回，用向量之间空间上的相似度代表俩个关键字之间语义的相似度。

查询“汽车”与“自行车”的相似度，效果如图 4.5

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_111.jdk/Contents/Home/bin/java ...
汽车 - 自行车 0.41543463

Process finished with exit code 0
```

图 4.5 word2vec-汽车与自行车的语义相似度评分

b. 百度 NLP-短文本相似度 api

正如相关模型综述所说的那样，百度 NLP-短文本相似度算法模型不仅利用了神经网络语言模型，深度学习以及词向量模型进行模型训练和语义匹配，而且最重要的一点是：百度自然语言处理部运行了大量用户的反馈数据进行了算法的再次优化以及模型的再次训练。这样经过“千锤百炼”的模型往往具有更好的效果。继续拿上面的“汽车”和“自行车”例子举例对比，效果如图 4.6

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_111.jdk/Contents/Home/bin/java .
汽车 - 自行车: 0.617797

Process finished with exit code 0
```

图 4.6 百度 NLP-汽车与自行车的语义相似度

由图可以看出，关键字“汽车”和“自行车”的文本相似度评分由基于中文 wiki 语料库的 0.415 变成了百度 NLP 的 0.618，语义相似度大大提高。这也说明了百度 NLP 接口的语义匹配效果的优越性。

4.2 空间相似度匹配算法

如何去衡量算法的空间相似度得分呢？因为 Helpal 提供的是一个移动群智服务平台，平台的目的是利用用户周围的闲置的社会劳动力，所以一般只会搜索周围较小半径内的服务。故在搜索范围比较小，比较容易达到的服务半径内，用人与人之间的欧式距离作为衡量空间的相似度得分就比较合适了。对于每个用户,Helpal 移动群智服务平台会时刻刷新着用户的位置信息，只要用户在移动端或者 web 端登录，系统就会根据 wifi 或者 GPS 定位功能将用户的最新位置的经纬度存入系统数据库，以备空间相似度的计算。

函数 getDistance(double lng1, double lat1, double lng2, double lat2)；本工程开发出的一个根据用户（lng1，lat1）和服务者（lng2，lat2）的经纬度，算出二者之间的欧式距离（以米为单位）的 java-api 接口。在得到用户之间的欧式距离后还需要进行的一个步骤就是数据的归一化处理。归一化处理数据的方式与很多：

z-score 标准化，Decimal scaling 小数定标标准化，对数 logoistic 模式等等。在这里我们采用最适合本系统开发的，最简单的归一化处理，该处理方式称为离差标准化，是对原始数据的线性变换，使其结果平滑地映射到区间[0 - 1]之间。转换函数公式如公式(4-2)

$$x^* = x - min/max - min \quad (4\text{-}2)$$

其中 max 为样本数据的最大值，min 为样本数据的最小值。这种归一化方法有个缺陷，就是每当有新数据加入时，可能导致 max 和 min 的变化，这样就要重新定义其值。但是在 Helpal 移动群智服务系统中，我们的 min 就为 0，max 即为用户在搜索时就设定好的搜索半径，不会改变，所以较为适合本系统的设定。举例来说，若俩人之间的距离为 800m，用户的搜索半径为 6km，则空间相似度得分$x^* = 1 - \{800 - 0/6000 - 0\} = 0.87$。至此空间相似度评分模型也初步形成。

4.3 $score_{Helpal}$公式权重参数w_1，w_2，w_3的确定

由$score_{Helpal}$的评分公式(4-1)可知，待定参数w_1，w_2，w_3分别代表了文本相似度得分，空间相似度得分以及用户信誉得分的占总体评分的权重（weight）。权重的确定方式是多种多样的，是根据实际情况来分析确定的。拿传统信息检索模型的关键字权重确定来说，一般采用的是 tf-idf（term frequency–inverse document frequency）方法确定，其中 tf 代表的词频，表示关键词在文档中出现的频率，idf 是逆文档频率，表示若包含某一关键词的文档越少，则 idf 的值越大，最后关键词的权重等于 tf 与 idf 的乘积，这样充分考虑的关键字在文档库中的重要性与代表性。文献[15][16]介绍了权重确定的一些主流方法。

在本系统平台中，利用层次分析法容易得到，文本相似度的得分应该是占总得分$score_{Helpal}$的很大一部分，是否能找到满足用户需求的服务类型往往都是最主要的。在搜索半径中找到相应的服务类型后，空间相似度的得分以及用户信誉得分所占的权重应该是差不多的，具体合理评分通过实验来确定。

经过如上的分析后，本工程运用模型优化的思想，首先构建待优化的目标函数公式(4-3)

$$\varepsilon = \alpha \times w_1 + \beta \times w_2 + \gamma \times w_3 \quad (4\text{-}3)$$

其中：$0.5 < w_1 < 1$；

$w_1 + w_2 + w_3 = 1$；

$\varepsilon \in (0,1)$

对于不同的测试数据α，β，γ得分来说，分别改变w_1，w_2，w_3的值以获取最合理的，最优的ε的值。随后再经过大量测试，不断优化目标函数(4-6)，就能得出w_1，w_2，w_3的估计最优取值比例（测试后的比例大约在 8：1：1 左右），然后以这取值组合带入新的查询验证，确定了其实际合理性。

5 实验评估

在实验评估这个环节，我们需要设置评测算法的指标，随后通过将对比数据集来带入算法运行并得出结果，最后分析结果并得出结论。首先要确定对算法的哪一部分进行评测。根据$score_{Helpal}$模型公式可以看出，决定模型的三个主要影响因素中：空间相似度得分基本就是获取用户与服务者的地理位置（经纬度），并根据此得出两人之间的欧式举例，根据归一化处理后带入公式即可，不太需要设计实验评估；用户信誉得分仅仅是用户反馈的数据，更是不涉及算法。所以最需要评估的就是文本相似度评分-语义匹配效果。

5.1 实验评估指标与算法影响因素

影响 word2vec 的语义相似度匹配的因素有哪些？根据相关论文的研究以及对 word2vec 源码的阅读后得知，在 word2vec 中影响词向量质量的主要有两个因素：训练语料库的质量以及训

练模型的优劣。所以在 Google 的训练模型算法足够先进的时候，训练语料库的质量就成为最关键的因素。语料库的质量包含语料库的大小以及语料库的相关性。

实验的评估指标来说，一般对于算法评估，大家都会在乎算法的效率（运算时间）以及效果（准确度）这两个指标。在本文的算法评估中，效率可以用代码记录下程序运行时间，但是两个词的语义相似度评分是否合理要怎么进行评估呢？

以个人的主观意见评估貌似不太合理，但是如果是一大群人的意见都是这样，那就比较有说服力了，在这方面，百度 NLP 短文本相似度 api 帮我们解决了这个问题。前面说过，百度 NLP 最亮眼的技术点在于，它使用了用户的反馈数据去调整了算法的性能与效果，接口运行至今必定被大量的用户调用过，所以用户的反馈数据代表了大部分人的观点，我们可以用百度 NLP 的文本相似度评分去衡量 word2vec 算法的文本相似度匹配效果！

5.2 实验及结果分析

5.2.1 实验环境

本次实验评估中，采用 Google 提供的 word2vec 的训练模型代码对语料库进行训练，训练模型环境为 c 语言运行环境；训练得到词向量文件后用 java 语言加载读取词向量文件并将其存在 java 数据类型 hashmap 中，最后运用自己编写的 similarity-api 接口在 hashmap 中进行关键字的查找，词向量的获得以及计算向量之间的余弦值，得出关键字的相似度评分。运行 java 语言的平台为 Intellij-idea。

5.2.2 实验设计及结果分析

实验 1

目的：验证语料库的大小影响语义匹配的的效果

数据集：中文 wiki 百科（1G）；搜狗提供的 2010 年体育板块的新闻语料（5G）；100 个查询关键字组。

实验流程：将两个语料库分别用分词工具进行分词，带入 word2vec 模型训练并获得目标词

向量文件，用 java 接口读取词向量文件后带入验证关键词并记录结果。

结果示例：取 5 组具有代表性的关键字组示例：西餐-牛排，自行车-汽车，果汁-饮料，餐厅-饭店，文具-铅笔。效果如图 5.1:

关键字组	语料库	中文wiki(1G)	搜狗新闻(5G)	百度NLP
西餐-牛排	相似度得分	0.6916326	0.7062346	0.731591
	运行时间(ms)	4	6	1301
自行车-汽车	相似度得分	0.41543463	0.5412536	0.617797
	运行时间(ms)	3	4	1062
果汁-饮料	相似度得分	0.8670166	0.8213452	0.811319
	运行时间(ms)	3	4	1020
餐厅-饭店	相似度得分	0.6195237	0.7612496	0.897259
	运行时间(ms)	3	6	1960
文具-铅笔	相似度得分	0.49137047	0.5329411	0.690909
	运行时间(ms)	3	5	1869

图 5.1 实验结果 1

从图 5.1 可以看出，对于挑选的 5 组关键字组，如果不进行语义分析，只进行关键字匹配的话，相似度得分会非常低，但是加上语义分析后，文本的相似度得分就比较符合人们的常识了；另外，以百度 NLP 的得分为标准，可以看出，搜狗的 5G 新闻语料库的训练出的词向量得分比起中文 wiki 百科 1G 的得分要更接近于百度 NLP 的标准得分。对于 100 组关键词组的结果分析大体上也符合上述趋势。所以我们可以得出初步结论：word2vec 词向量算法模型准确度与语料库的大小有关，语料库越大，越广泛，越具有代表性，则关键字匹配效果会更好。

实验 2

目的：验证语料库的相关性影响语义匹配的的效果

数据集：搜狗提供的 2010 年政治板块的新闻语料；搜狗提供的 2010 年体育板块的新闻语料；50 个政治相关的关键字+50 个体育相关的关键字。

结果示例：取 3 组政治相关的代表词汇：政府-政党，国家-中国，马克思-恩格斯；3 组体育相关的代表词汇：篮球-足球，竞技-比赛，科比-詹姆斯；结果如图 5.2

	关键字组	语料库	搜狗-政治	搜狗-体育
政治	政治-政党	相似度得分	0.49005306	——
		运行时间(ms)	4	
	国家-中国	相似度得分	0.50853885	0.42155351
		运行时间(ms)	3	4
	马克思-恩格斯	相似度得分	0.81254005	——
		运行时间(ms)	3	
体育	篮球-足球	相似度得分	0.6023571	0.6760753
		运行时间(ms)	3	3
	科比-詹姆斯	相似度得分	——	0.44004658
		运行时间(ms)		
	竞技-比赛	相似度得分	0.41530472	0.6012357
		运行时间(ms)		4

图 5.2 实验结果 2

由实验示例结果图可以看出，政治相关的 3 个关键字组有两个在体育语料库中查询不到，体育相关的 3 个关键字组在政治语料库中有一个查询不到。并且在自己领域的词组得分较高。100 个关键字组（50 个政治相关，50 个体育相关）的结果显示趋势与例子类似。所以，我们可以得出初步结论：word2vec 词向量算法模型准确度与语料库的相关性有关，在语料库相关领域内的关键字匹配效果会更好。

综上所述：通过两个简单的实验，验证了我们对算法影响因素的分析：在 Google 的训练模型算法足够先进的时候，语料库的大小以及语料库的相关性就成为最关键的因素：语料库越大，越具有普适性，最终的文本相似度匹配效果就会越好！

6 Helpal web 前端工程概述

本章节将介绍 Helpal-web 前端工程的组织结构，主要功能和界面效果，分析讨论开发过程中的面临的难题，并给出解决方案。

6.1 web 前端概要设计

Helpal 的 web 前端有四个主要界面：

1. 登录注册界面：用户可以在此界面进行登录注册，见图 6.1 及图 6.2；

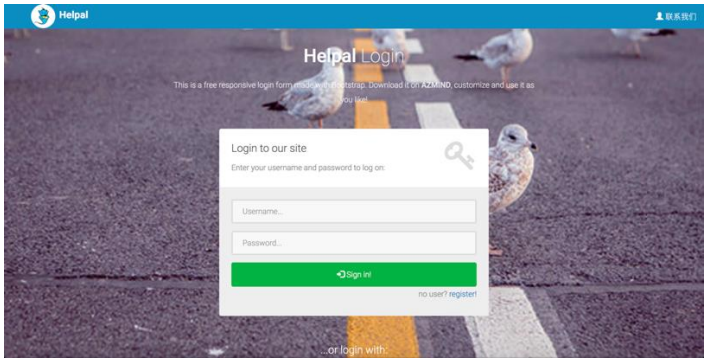

图 6. 1 Helpal 登录界面

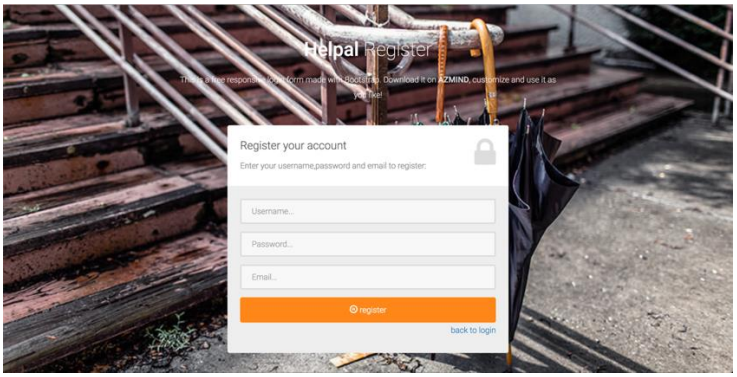

图 6. 2 Helpal 注册界面

2. 个人主页：用户可以在此界面修改自己的基本信息：修改用户昵称，头像等；此页面还有上架 / 删除服务以及添加 / 删除自己的兴趣功能，用户可以进行操作（见图 6. 3 至图 6. 6）；

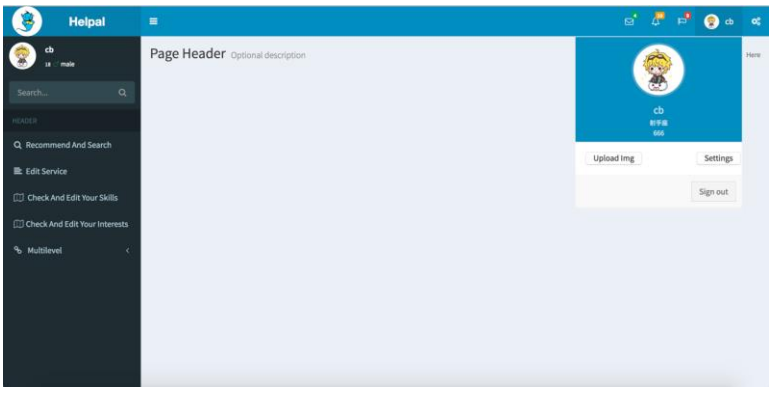

图 6. 3 Helpal 个人主页

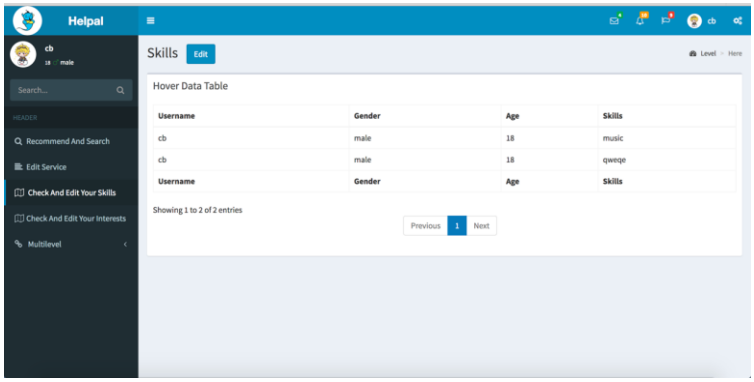

图 6. 4 Helpal 查看修改个人服务技能

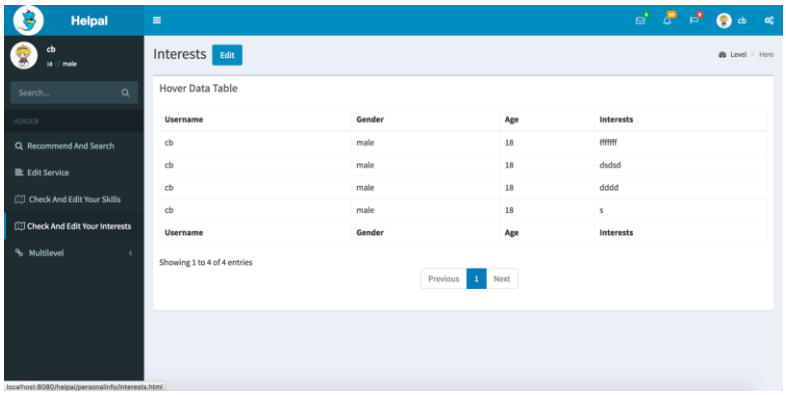

图 6. 5 Helpal 查看修改个人兴趣

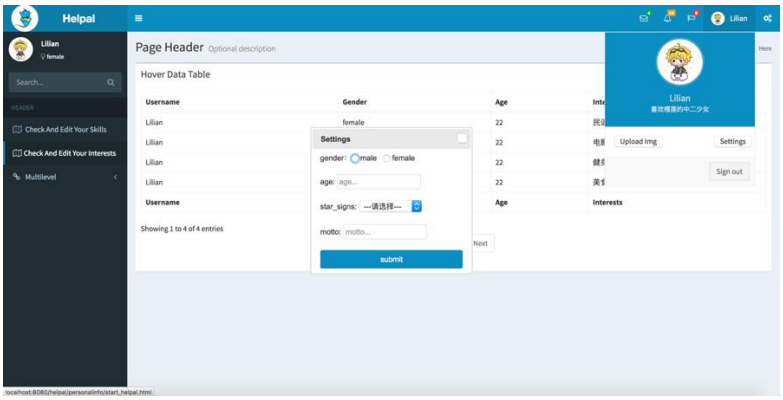

图 6. 6 Helpal 上传头像修改个人信息及登出功能

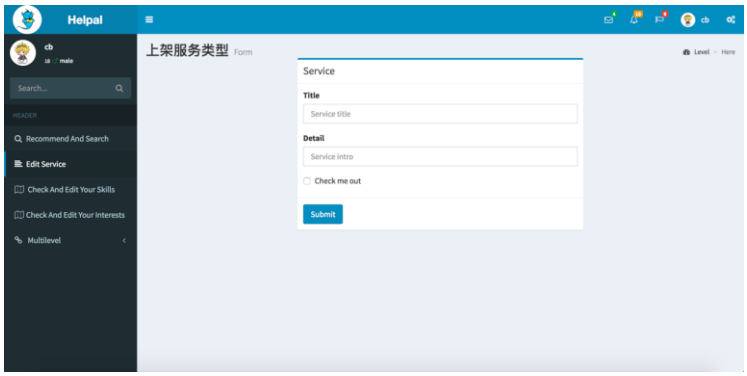

图 6.7 上架服务

3. 结合 baidu 地图的搜索及展示界面：见图 6.8 至图 6.11.，用户可以从个人主页进入此界面，该界面在地图上以地图图标的形式显示推荐用户的推荐内容；该界面上部拥有搜索框，用户可以在搜索框输入需求的服务类型关键字，例如“西餐”等，待结果返回后，在本界面以地图图标的形式显示给用户，鼠标停顿在图标上会有服务的简要介绍；

图 6.8 自动用户定位

图 6.9 推荐范围为 0.5km 时的地图效果

图 6. 10 推荐范围为 1.5 公里时的推荐效果

图 6. 11 搜索“牛排”时的地图效果

4. 及时通信界面：用户如果想要跟某一服务者联系，点击地图图标的联系功能，自动跳转至通信界面，在此界面用户可以跟服务者商量服务细节等。

Web 前端的流程图，如图 6.12 所示

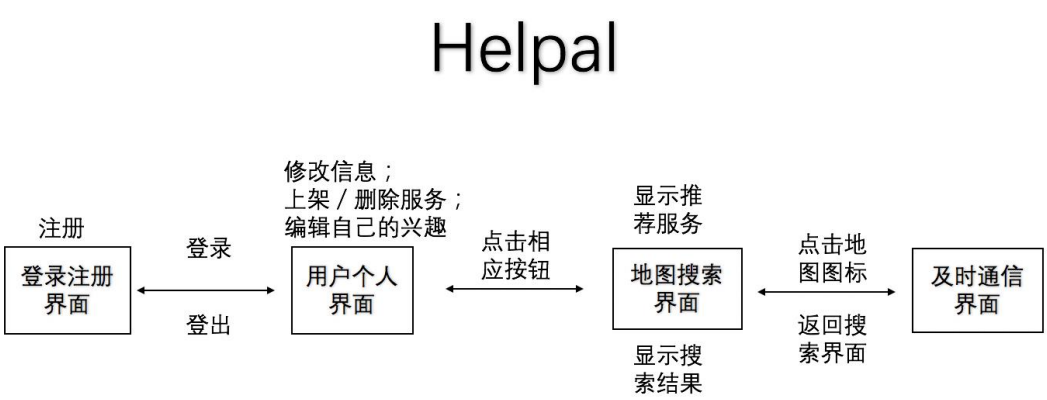

图 6.12 Helpal-web 前端系统流程图

6.2 web 前端工程技术点及解决方案

设计完 Helpal-web 的主体框架后，剩下的就是利用 javascript，jquery 等脚本语言进行前端工程开发了，在工程开发过程中，也遇到了不少问题和挑战，下面是一些典型的问题和解决方案。

6.2.1 浏览器的跨源访问

web 前端是通过 ajax，jquery 的 post，get 方式与后端 API 接口进行交互，在此遇到了浏览器的跨源访问问题。所有的浏览器都有同源策略(同协议，同域名，同端口);为了安全，浏览器都会拒绝跨源访问。之前都是以自己的电脑作为服务器，自然通过 ajax 跨源访问就不会出现该问题，但在开发 helpal web 端时，该问题就出现了，具体见图 6.13

图 6.13 web 跨源访问问题

在网上查找相关资料后，发现有两种解决方案：

(1).ajax 本身是支持跨域访问，在前端 dataType 要设定为“jsonp”。Jsonp(JSON with Padding)是 json 的一种“使用模式”，可以让网页从别的网域获取资料。

缺点：服务端需要配合前端在返回的结果前加上 jsonp 格式默认的 callback 函数头，由于

后端不只是为 web 端所做，故不采用该方案。

(2) 利用 W3C 规定的专门解决跨源访问的协议“跨域资源共享(CORS, Cross-Origin Resource Sharing)”。在服务器端设置返回的 header 头：response.setHeader("Access-Control-Allow-Origin", "*")，该方案的可行度较高，这样就不用后端在返回值方面做出改变，故采用该方案，实际效果见图 6.14，能够成功的获得服务器的返回字符串。

图 6.14 成功跨源访问示例

6.2.2 本地存储后端响应数据

因为 web 前端需要跟后端进行数据交互，所以难免需要在本地存储后端的响应数据，查阅相关资料后发现，前端存储数据主要有下面两种方式：

(1) localstorage 和 sessionstorage

HTML5 提供了两种在客户端存储数据的新方法：

localStorage - 没有时间限制的数据存储

sessionStorage - 针对一个 session 的数据存储,关闭浏览器后本地存储的数据自动消除

优点：可在本地存储大量数据；

缺点：不能设置过期时间，不符合网页标准；（不采用）

(2) 网页传统 cookie 存储

优点：可设置过期时间，能够实现网页的“记住”功能，可规定那些网页可使用 cookie；

缺点：不可存储大量的数据（但 helpal 项目并不需要在本地存储大量用户数据）；（采用）

我自己在 web 前端用 javascript 写了一个集成的存储 cookie 类-cookieStorage 类，解决了本地存储的问题，下面是简单说明类函数功能的伪代码；

```
function cookieStorage(max,age){        //maxage 为生效时间，path 为 cookie 生效的域
setItem(key,value);                     //设置名为 key，值为 value 的 cookie
```

```
getItem(key);                    //获取名为 key 的 cookie
key(n);                          //获取第 n 个 cookie 的名
removeItem(key);                 //移除名为 key 的 cookie 值
clear();                         //清除所有 cookie
}
```

6.2.3 用户的定位及地图调用

向之前所说，Helpal 是移动群智平台，充分利用用户周围的闲置劳动力，所以及时更新以及记录用户的实时位置是平台正常运行的基本条件。就 web 前端层面来说，也有相关的定位技术获取用户登录时的经纬度:HTML5 Geolocation API

HTML5 的 Geolocation API 用于获得用户的地理位置，支持包含 Internet Explorer、Firefox、Chrome、Safari 等流行的主流浏览器位置定位。该 API 主要是通过里面 getCurrentPosition() 方法来获得用户的位置，该方法将返回用户的经纬度。

优点：由于该 api 是通过浏览器定位，很符合 web 前端的定位标准，用户只要打开浏览器登录 Helpal,浏览器就会自动定位。

缺点：正式由于该 api 是通过浏览器定位的，所以误差比较大，大约会有几 km 左右的误差，不像移动端，拥有精确的 wifi 以及 GPS 定位系统。所以精确的定位以及更新用户的位置还是得依靠 Helpal 的移动端来实现。

地图调用上，为了与国内的网络兼容，考察几个地图应用后，最终还是采用了百度地图。因为百度地图[17]经过长时间的开发，基于地图的应用 api 也是比较全面，完善，易于理解的，地图上附加各种插件基本可以满足开发者的所有需求，而且 bug 很少，官方的各种开发文档也是极为全面的。

6.2.4 及时通信功能

因为在 Helpal 移动群智感知平台上，如果用户相中了满意的服务者，是需要联系服务者以商量具体细节的。所以，及时通信功能也是必须的。因为从头开始开发自己平台的及时通信功

能十分复杂，麻烦，效率不高，于此相比，网上的第三方通信软件比较健全，容易利用。所以，计划在 Helpal 工程中引用第三方通信框架进行简单的用户和服务着之间的通信交流。

7 总结与展望

本文主要研究了 Helpal 移动群智服务平台的关键搜索技术，提出了合理的相关模型以及评分方法并加以实现，最后运行对比试验初步评估验证了基于 word2vec 词向量的文本相似度算法的影响因素。此外，本文还介绍了 Helpal-web 的前端工程框架，工程开发中遇到的问题和解决方案。

到此，本文还有些不足之地：由于没有足够的环境支持以及时间紧迫，word2vec 的评估实验实验数据集还是过小，评估实验设计不够全面，只能看出大致趋势，不能排除极端数据对评估效果的影响；后续为了最终完善 Helpal 移动平台搜索系统，增加系统的的可拓展性，普适性以及人性化设置，还是需要对用户的输入进行关键字的提取，去除无关词汇对搜索结果的影响等。

未来社会，随着移动设备的大量普及（有相关研究结构报导，现在移动设备的数量已经超过了世界人口数），基于位置的服务（LBS）一定会成为最热门的开发领域，不断会有开发商涌进该领域。开发过程中，信息数据的获取往往是最重要的技术点和令人头痛的难点，所以本文的 Helpal 移动群智服务平台搜索技术研究是比较具有现实价值和意义的。本文在该技术的研究上也是提出了一点见解，希望在将来，基于位置的服务（LBS）类型会更加多样，人性化，让移动设备为人类社会创造更多的便利！

致谢

首先衷心的感谢我的毕业论文导师陈红教授，她给我提供了这么具有研究开发意义的毕业设计课题，给了我一个宝贵的锻炼自己，提高自己能力的机会。在她极其忙碌的时候还白忙之中抽出时间，每周给我们开一次组会，关心我的工程进展并热心的为我们解答疑问。

感谢实验室的博士生吴垚师兄，他带领我一步步完成了系统的开发，在我迷茫的时候，他

总是能够为我排除困难，给我指出正确的研究方向。他在我的研究工作以及论文的撰写上给予了我极大的支持。

感谢同一毕业设计课题小组的郭若杨，陈俊华，钟典同学，每当我有问题询问他们，他们总是不厌其烦地帮我一起研究问题，找到问题答案；每当我有工程要求时，他们都在尽力支持我的开发工作。还要感谢实验室的郭诚欣师兄， 吴天贞师姐，他们为我提供了实验室的服务器支持，让我能够顺利的跑出工程结果。

最后，感觉帮助过我的所有人。

作者签名：陈波

参考文献

[1] D R 威廉姆斯, J 希尔. 机器学习, Machine Learning:, CN 101010934 A[P]. 2007.

[2] Weikum G. Foundations of statistical natural language processing[J]. Acm Sigmod Record, 2002, 31(3):37-38.

[3] Hiemstra D. Information Retrieval Models[M]//Information Retrieval: Searching in the 21st Century. 2013:27-37.

［4］Google. simhash 官网 [EB/OL].https://simhash.codeplex.com.2017 年 3 月 10 日访问.

［5］Charikar M S. Similarity estimation techniques from rounding algorithms[C]// Thiry-Fourth ACM Symposium on Theory of Computing. ACM, 2002:380-388.

［6］Google.word2vector 官网[EB/OL].https://code.google.com/archive/p/word2vec/. 2017 年 3 月 15 日访问.

［7］Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space[J]. Computer Science, 2013.

［8］ Mikolov T, Yih S W, Zweig G. Linguistic Regularities in Continuous Space Word Representations[C]// 2013:296-301.

［9］ Mikolov T, Sutskever I, Chen K, et al. Distributed Representations of Words and Phrases and their Compositionality[J]. Advances in Neural Information Processing Systems, 2013, 26:3111-3119.

[10] Morin F, Bengio Y. Hierarchical probabilistic neural network language model[J]. Aistats, 2005.

[11]Baidu.NLP[EB/OL].https://cloud.baidu.com/product/nlp.html?track=cp:nsem|pf:pc|pp:nlp|pu:brand|ci:|kw:.62656.2017 年 3 月 30 日访问.

[12] Baidu-NLP. 短文本相似度 api 技术文档 [EB/OL].https://cloud.baidu.com/product/nlp/simnet.2017 年 3 月 30 日访问.

[13] Mikolov T, Le Q V, Sutskever I. Exploiting Similarities among Languages for Machine Translation[J]. Computer Science, 2013.

[14] 张华平博士.NLPIR 汉语分词系统[EB/OL].http://ictclas.nlpir.org.2017 年 4 月 2 日访问.

[15] Yoon K P, Hwang C L. Multiple Attribute Decision Making[M]. Springer Berlin Heidelberg, 1995.

[16] Chen S J J, Hwang C L / M J, Krelle W. Fuzzy Multiple Attribute Decision Making: Methods and Applications[J]. Lecture Notes in Economics & Mathematical Systems, 1994, 375(4):1-531.

[17] Baidu-map.api 开发文档[EB/OL].http://developer.baidu.com/map/jsdemo.htm#i8_1.2017 年 4 月 6 日访问.