**电 子 科 技 大 学**

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

**学士学位论文**

**开题报告**

**BACHELOR THESIS**



论文题目  **基于机器学习的问答推荐算法设计**

专 业 **计算机科学与技术**

学 号 **2014060108020**

作者姓名 **伍峰**

指导教师 **俸志刚**

# 第一章 选题依据及意义

当今互联网发展如此迅速,日新月异的各种技术层出不穷,什么大数据,云计算,人工智能等正潜移默化的进入了人们的生活,这里面我们都能看到一个技术的身影,那就是机器学习,面对互联网信息爆炸的今天,信息量爆炸,用传统的信息处理方式已经不足以满足现在的数据量,也就有了机器学习的用武之地.由于对机器学习算法很感兴趣,再加上机器学习现在的火爆程度,以及今后的发展前景,就选择这个课题作为我的毕业设计.

# 第二章 研究现状及发展态势

按照人工智能大师西蒙的说法,学习就是系统在不断重复的工作中对本身能力的增强或者改进,使得系统在下一次的执行相同任务或者类似的任务的时候,会比之前做的更好更出色或者效率更高.

顾名思义，机器学习是研究如何使用机器来模拟人类学习活动的一门学科。稍为严格的提法是：机器学习是一门研究机器获取新知识和新技能，并识别现有知识的学问。这里所说的“机器”，指的就是计算机；现在是电子计算机，以后还可能是中子计算机、光子计算机或神经计算机等等。

机器学习分为三大类:监督式学习,非监督式学习,半监督式学习

监督式学习,基本上实现的就是分类的功能, 它从有标签的训练数据中学习，然后给定某个新数据，预测它的标签。这里的标签，其实就是某个事物的分类。

非监督学习,与监督学习恰恰相反,非监督学习面对的训练数据都是没有标签的,给定数据,从数据中学习,能学到什么,这取决于数据本身具有什么样的特征.

半监督式学习,这种学习方式借鉴了监督学习和非监督学习,去其糟粕取其精华,半监督学习通过对部分标签数据的学习,从而对之后的数据进行分类,但不像监督学习一样,一旦学习完毕,就不再更新自身,而半监督学习会根据之前的经验继续对数据进行学习

# 第三章 课题研究内容

课题研究的内容主要是搜索引擎和机器学习的交叉应用.

主要任务有:

1） 学习机器学习的基本理论、方法和工具

2） 编写网络爬虫工具从知乎网站抓取问答数据，构建训练和测试数据集

3） 设计“问题-回答”匹配算法模型及编程实现，并利用前述数据集训练该模型

4） 与其他相关算法模型比较模型的性能参数

5） 实现一个问题推荐演示系统

总体来说先编写爬虫在知乎上爬取问答数据,并且构建数据集,并且预处理数据,设计索引结构,类似搜索引擎的结构,接着设计匹配算法,用数据集训练搜索结果排序模型,然后实现一个问题推荐的演示系统即可.

# 第四章 主要理论

LTR(learn to rank):即机器学习排序算法,利用机器学习来对搜索结果进行排序, ,原来只有少量因素的时候,可以通过人工进行公式拟合,是行得通的,但随着搜索引擎所考虑的因素越来越多,指望人工拟合已经不太现实,这个时候就需要使用我们的机器学习算法来对搜索结果进行排序.LTR算法包括了三个大类,PointWise(单文档方法),PairWise(文档对方法),ListWise(文档列表方法).

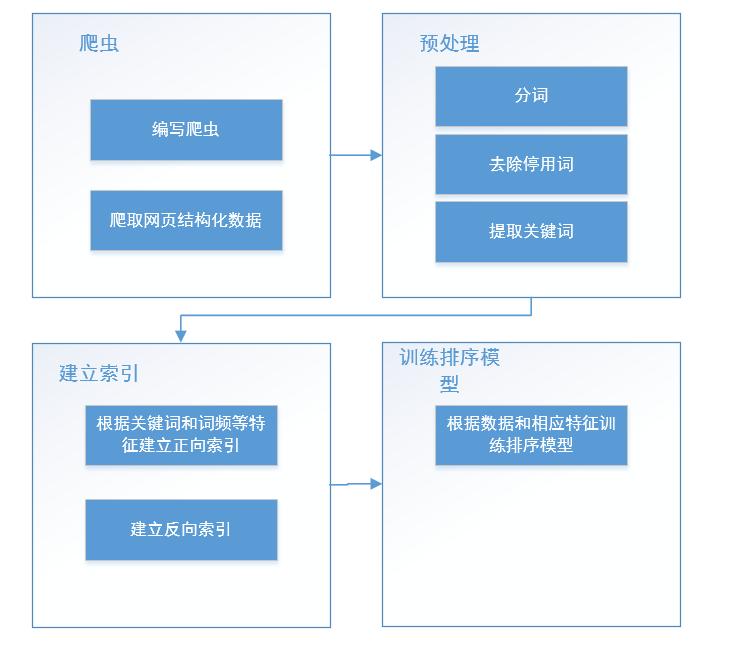
搜索引擎:搜索引擎的基本原理包括3个步骤,第一步,先通过网络爬虫爬取网站信息,第二步,预处理数据,包括提取关键字,建立索引等.第三步,最后提供检索服务.最重要的部分是数据的处理,建立合适的索引.

中文分词: 分词就是将连续的字序列按照一定的规范重新组合成词序列的过程,但由于中文的特殊性,导致中文分词比英文分词要难很多,于是有了各种针对中文分词的算法,常用的有: 正向最大匹配法, 逆向最大匹配法, 最少切分等

提取关键词:即提取出文本中的关键词,常用的算法有TF-IDF, TextRank算法等.

# 第五章 技术路线

技术路线如下图,大致分为4个阶段:



# 第六章 实施方案

实施方案分为以下几步:

1. 用python语言,使用scrapy爬虫框架编写构建知乎网站爬虫,爬取问答结构化数据,包括问题,答案,主题.

具体结构设计如下:

问题:

class QAQuestionItem(scrapy.Item):

# 提问题用户ID

ask\_user\_id = scrapy.Field()

# 问题ID

question\_id = scrapy.Field()

# 提问时间

create\_time = scrapy.Field()

# 更新时间

update\_time = scrapy.Field()

# 答案数目

answer\_count = scrapy.Field()

# 关注者数量

followees\_count = scrapy.Field()

# 问题标题

title = scrapy.Field()

# 问题详细说明

content = scrapy.Field()

# 问题被浏览次数

view\_count = scrapy.Field()

# 问题所属话题ID列表

topics = scrapy.Field()

# 问题评论数目

comment\_count = scrapy.Field()

答案:

class QAAnswerItem(scrapy.Item):

# 回答用户ID

answer\_user\_id = scrapy.Field()

# 答案ID

answer\_id = scrapy.Field()

# 问题ID

question\_id = scrapy.Field()

# 答案创建时间

create\_time = scrapy.Field()

# 答案更新时间

update\_time = scrapy.Field()

# 答案点赞数

voteup\_count = scrapy.Field()

# 评论数

comment\_count = scrapy.Field()

# 回答内容

content = scrapy.Field()

# 简介

excerpt = scrapy.Field()

主题:

class QATopicItem(scrapy.Item):

# 话题名称

name = scrapy.Field()

# 话题描述

desc = scrapy.Field()

# 话题ID

id = scrapy.Field()

# 子话题ID列表

children = scrapy.Field()

# 父话题ID

father = scrapy.Field()

1. 预处理数据
   1. 分词
   2. 去除停用词
   3. 提取关键字并计算权重
   4. 提取其他特征值
2. 根据关键词建立正向和反向索引
3. 训练排序模型,采用listnet算法模型
4. 分析测试模型,写出分析报告
5. 使用模型完成一个输入问题显示推荐答案系统

# 第七章 关键问题

爬虫部分:这部分的关键问题包括:

1. 通过什么路径能够尽可能爬取到知乎网站的问答数据?
   1. 通过穷举所有问题ID,,然后补全URL(https://www.zhihu.com/question/XXXX/)进行一个个的获取问答数据
   2. github上有个非官方的API,可以用来获取问答数据
   3. 通过知乎独特的话题有向图设计,使用广度优先或者深度优先爬取各个话题下的所有问答数据

经过测试和分析,最后采取了第三种方案,并且使用广度优先爬取问答数据

1. 如何避免或者减轻网站对爬虫的检测
   1. 降低爬取速率
   2. 使用代理IP

预处理部分:最关键的问题是,如何设计特征值

训练模型部分:多少数据量才算合适

# 第八章 最终目标

1） 完成一个可为新问题智能匹配相关回答的算法模型

2） 针对特定测试数据集的测试结果分析报告

3） 输入问题，同步显示推荐回答的演示系统