**密级： 保密期限：**



**硕士学位论文**



**题目： 基于深度学习的排序模型的研究与实现**

**学 号： 2015111589**

**姓 名： 赵 轩**

**专 业： 计算机科学与技术**

**导 师： 闫丹凤**

**学 院： 网络技术研究院**

**2017年 11 月 30 日**

独创性（或创新性）声明

本人声明所呈交的论文是本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，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邮电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。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名： 日期：

关于论文使用授权的说明

学位论文作者完全了解北京邮电大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属北京邮电大学。学校有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许学位论文被查阅和借阅；学校可以公布学位论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存、汇编学位论文。（保密的学位论文在解密后遵守此规定）

保密论文注释：本学位论文属于保密在 年解密后适用本授权书。非保密论文注释：本学位论文不属于保密范围，适用本授权书。

本人签名： 日期：

导师签名： 日期：

基于深度学习的排序模型的研究与实现

摘 要

PDF（Portable Docuent Format）由Adobe公司开发并推广，是一种独特的跨平台的便携文件格式。跨平台特性使得PDF文件可以广泛的运用于Windows，Unix，Mac OS等当前主流的操作系统中，并使其成为Internet上电子文档发行和数字化信息传播的理想文档格式。如今的互联网上，越来越多的电子书籍、产品使用说明书、公司公告财报、网络资料、科学文献、电子邮件等都开始使用PDF格式作为电子文档的首选格式。

随着PDF格式的普及，大量有价值的信息都以PDF文档的形式呈现出来。因此从PDF中提取有价值的信息也成为了近年来的研究热点。然而由于PDF的结构相对复杂，从PDF中提取文本、图形、表格等信息的难度也相应增加，尤其是PDF中的表格信息。与Html等格式不同，PDF格式对于表格并没有单独的定义，PDF中的表格只是单纯的线条与文字的集合，因此PDF表格数据的识别和提取变成了不小的挑战。传统的针对网页中表格的识别和提取技术依赖于Html中表格的tag信息，因此很难直接应用于PDF中的表格提取，因此本课题提出了一种针对PDF的表格识别和提取的通用方法，为了验证方法的有效性和准确性，本课题将该方法应用在公司财务表格数据的提取上，经过测试，该方法具有较好的性能。

本课题首先阐述了论文的研究背景，介绍了PDF结构的主要特点，同时对本系统使用的PDF类库PDFBox进行了介绍。接着对比了几种常用的表格提取方法，通过比较和分析各个方法的优劣最终引出本系统采用的方法。论文的后半部分对本方法涉及的技术难点进行了详尽的介绍，包括基本的表格框线识别，基于框线的表格还原，复杂首行首列表格的处理，跨页表格的合并，表格数据格式化等内容。最后通过实现PDF财报中三大财务报表表格数据的识别和提取对本系统的解析的效果和性能进行了相应的测试和评估。

**关键词：**PDF结构 表格识别 表格栅格化 表格框线识别

**Design and Implementation of PDF Format Based Table Extraction Method**

ABSTRACT

PDF（Portable Document Format）is an unique portable cross-platform file Format developed by Adobe. The cross-platform feature makes PDF files widely used in Windows, Unix, Max OS and other current mainstream operating system and make it become the internet electronic document issued and ideal document format of digital information transmission. Now on the internet, more and more electronic books, product specification, company earnings announcement, network information, science, literature, E-mail, etc are firstly choosing PDF format as electronic document format.

With the popularity of PDF format, a large number of valuable information are presented in the form of a PDF document. So to extract valuable information from PDF files have become a research hotspot in recent years. However, due to the complex structure of PDF, it is not that easy to extract text, graphics, tables from PDF files, especially for PDF tables' extraction. PDF format is different from Html format. There is no definition for tables in a PDF file. A table in PDF files are collection of words and lines which make PDF table extraction a big challenge. Traditional methods for table recognition and extraction are highly relies on the tag of Html form information which is not suitable for PDF table extraction. In order to solve this problem, this paper presents a general method for PDF table recognition and extraction. In order to verify the validity and accuracy of the method, the paper then apply the method on the extraction of financial table data, the result shows that the method has good performance.

The paper firstly elaborated the research background, the main characteristics of the PDF is introduced, at the same time introduced the PDFBox library which is used by the system. Secondly, the paper compares several common table extraction method, through the comparison and analysis of the pros and cons of each method eventually lead to the method which is used by the paper. The paper then made detailed introduction of the PDF files' table extraction method including basic box line identification, table reduction, first raw and first column processing, across page table merging, tabular data format, etc. Finally, the paper tested and evaluated the method's performance by implementing the three financial statements form data recognition and extraction.

**KEY WORDS:** structure of PDF table recognition table rasterize box line identification

目 录

[第1章 绪论 1](#_Toc406841142)

[1.1 研究背景 1](#_Toc406841143)

[1.2 研究意义 2](#_Toc406841144)

[1.3 硕士研究生期间的工作总结 3](#_Toc406841145)

[1.4 论文组织结构 3](#_Toc406841146)

[第2章 背景理论及相关技术介绍 4](#_Toc406841147)

[2.1 PDF介绍 4](#_Toc406841148)

[2.1.1 PDF概述 4](#_Toc406841149)

[2.1.2 PDF组织结构 4](#_Toc406841150)

[2.1.3 PDF文档示例 4](#_Toc406841151)

[2.2 PDFBox介绍 4](#_Toc406841152)

[2.2.1 PDFBox概述 4](#_Toc406841153)

[2.2.2 PDFBox组织结构 4](#_Toc406841154)

[2.3 本章小结 4](#_Toc406841155)

[第3章 常见表格数据抽取方法的介绍与分析 4](#_Toc406841156)

[3.1 通用的表格识别方法介绍 4](#_Toc406841157)

[3.2 不同媒介下的表格特征识别 4](#_Toc406841158)

[3.2.1 基于Web的表格特征识别 4](#_Toc406841159)

[3.2.2 基于图像的表格特征提取 4](#_Toc406841160)

[3.2.3 基于PDF的表格特征提取 4](#_Toc406841161)

[3.3 表格数据抽取的难点分析 4](#_Toc406841162)

[3.3.1 表格结构识别 4](#_Toc406841163)

[3.3.2 复杂表头的处理 4](#_Toc406841164)

[3.3.3 无框线表格的还原 4](#_Toc406841165)

[3.4 本章小结 4](#_Toc406841166)

[第4章 PDF文件的表格数据抽取关键问题研究 4](#_Toc406841167)

[4.1 基于PDFBox的表格特征提取 4](#_Toc406841168)

[4.1.1 PDF字符流中表格的基本格式介绍[1] 4](#_Toc406841169)

[4.1.2 PDFBox中文字和线条的处理介绍 4](#_Toc406841170)

[4.1.3 基于PDFBox的文字信息提取与封装 4](#_Toc406841171)

[4.1.4 基于PDFBox的线条信息提取与封装 4](#_Toc406841172)

[4.2 表格还原 4](#_Toc406841173)

[4.2.1 基于线条信息还原基本二维表结构 4](#_Toc406841174)

[4.2.2 基于位置信息填充表格文字 4](#_Toc406841175)

[4.2.3 一表多页的复杂表格合并 4](#_Toc406841176)

[4.3 表格结构识别 4](#_Toc406841177)

[4.3.1 复杂表头的识别与转换 4](#_Toc406841178)

[4.3.2 表格结构类型 4](#_Toc406841179)

[4.3.3 表格展开方向识别 4](#_Toc406841180)

[4.4 本章小结 4](#_Toc406841181)

[第5章 PDF表格数据抽取系统设计与实现 4](#_Toc406841182)

[5.1 PDF表格数据抽取系统的整体设计 4](#_Toc406841183)

[5.1.1 系统整体流程设计 4](#_Toc406841184)

[5.1.2 系统层次描述 4](#_Toc406841185)

[5.2 基于PDFBox的文档解析层 4](#_Toc406841186)

[5.3 PDF表格还原层 4](#_Toc406841187)

[5.4 PDF表格数据封装层 4](#_Toc406841188)

[5.4.1 表格数据的HTML格式封装 4](#_Toc406841189)

[5.4.2 表格数据的JSON格式封装 4](#_Toc406841190)

[5.5 测试和分析 4](#_Toc406841191)

[5.5.1 表格识别技术的评估参数 4](#_Toc406841192)

[5.5.2 实验结果与分析 4](#_Toc406841193)

[5.5.3 表格结构识别测试与分析 4](#_Toc406841194)

[5.6 本章小结 4](#_Toc406841195)

[第6章 总结与展望 4](#_Toc406841196)

[6.1 论文总结 4](#_Toc406841197)

[6.2 工作展望 4](#_Toc406841198)

[参考文献 4](#_Toc406841199)

[附录 4](#_Toc406841200)

[致 谢 4](#_Toc406841201)

[攻读学位期间发表的学术论文目录 4](#_Toc406841202)

# 绪论

## 研究背景

随着互联网的高速发展，计算机相关技术已经渗透入人们学习生活的方方面面。现在，人们可以使用搜索引擎获取各类信息知识，可以在电商平台进行网上购物，可以使用社交软件与亲友保持联系。得益于信息检索和推荐系统领域相关技术的发展进步，各类互联网服务正趋于个性化和智能化：搜索引擎会根据用户的搜索点击历史，为其提供个性化的搜索结果；新闻资讯类应用软件可以主动给用户推送其可能感兴趣的信息内容；电商平台会根据消费者特点进行商品推荐；社交应用通过分析用户的人际关系，向其推荐可能认识的人，帮助用户结识新朋友、拓展人脉。在此背景下，如何提高检索和推荐结果的准确度，成为当前迫切需要解决的问题，针对这一问题的研究具有十分重要的现实意义。

排序问题是信息检索和推荐系统等领域的核心问题之一。例如，在搜索引擎的应用场景中，需要将网页搜索结果按照与用户的检索目的的符合程度进行排序；在推荐系统中，需要把候选物品按照用户可能感兴趣的程度进行排序。排序结果的准确性和合理性会直接影响检索和推荐的质量。因此，对排序模型和排序结果优化方法进行研究具有很高的价值。排序问题的传统解决方案大多依赖人工经验，通过组合一系列排序规则，或是由专家根据待排序项的特征确定排序公式。随着对排序问题研究的不断深入，目前比较常用的做法是利用机器学习相关技术解决排序问题。与传统解决方案相比，基于机器学习的排序模型具有更高的效率和准确度，得到的排序结果也具有更强的客观性和可解释性。

近年来，深度学习作为学术领域的重点研究方向之一，取得了一系列显著的进展。例如，卷积神经网络（CNN）在处理计算机视觉方向中的物体识别、图像分类等问题中，取得了明显领先于以支持向量机（SVM）为代表的传统算法模型的效果；循环神经网络（RNN）则被广泛用于处理时间序列信息数据，特别是其变体：长短时记忆网络（LSTM）和门控循环单元（GRU），通过特殊的结构设计解决了RNN中的长期依赖问题，在处理如机器翻译、语音识别、机器创作等任务中取得了令人惊喜的效果。众多研究成果表明，在训练数据充足的条件下，深度神经网络模型在处理如分类、回归等机器学习任务时，能够取得显著优于传统机器学习算法模型的效果。深度学习技术在排序问题中的应用方面，谷歌公司在2016年发表的论文中，提出其在YouTube的视频推荐系统中使用了深度模型实现的候选生成模块和排序模块。同时，谷歌还在其另一篇论文中提出了一种由广义线性模型和深度神经网络模型组合而成的混合模型结构，将该模型用于Google Play商店的应用推荐，并取得了良好的效果。

## 研究意义

如上所述，信息检索、推荐系统、计算广告等计算机技术应用方向已经广泛存在于人们生活的各类场景中，为用户提供各种便捷的服务。而排序问题作为上述领域中的一类通用问题，对其进行研究，对于优化应用效果、改进服务质量、提高用户满意度具有重要的意义。

针对排序问题的研究，可以主要分为排序系统的架构设计、主体算法模型的选取和结果评价标准的选择等方面。其中，排序系统的架构设计和主体算法模型的选择是研究的重点，也是难点问题。早期解决排序问题的方法主要是由专家根据业务场景特点，结合对历史数据的分析，制定一系列排序规则，或者是利用待排序项的各属性特征值确定出打分公式，使用该公式对各候选项进行打分，并根据得分高低确定最终的排序结果。这种方法的主要缺点在于排序完全依赖于专家的经验，缺少客观性和说服力。同时，人工分析历史数据的效率十分低下，在数据海量增长的当前环境下，已变得不可行。随着研究的深入，研究者们开始尝试使用机器学习的技术手段来解决排序问题，并逐步发展为一个热点主题：排序学习（Learning To Rank，LTR）。与依靠专家的传统方法相比，机器学习算法模型对数据的分析更加全面，能够挖掘出数据中隐含的、人工难以发现的规律和性质。同时，得益于计算机硬件的飞速发展和分布式计算等领域取得的一系列研究进展，如今计算机单机/集群的数据处理能力相比过去都得到了巨大的提升，这使得算法模型能够在大规模数据集上进行高效的训练。

LTR的主要流程包括数据采集、特征抽取、模型训练、场景应用等阶段。其中，特征抽取和模型训练是最为关键的两个阶段，很大程度影响最终的排序结果。模型训练阶段的重点在于算法模型的选择。目前LTR中广泛使用的机器学习算法模型，主要以统计学习模型为主，如逻辑回归（LR）、支持向量机（SVM）、梯度提升树（GBDT）、随机森林（RF）等。近几年来，深度学习成为广大学术研究者关注的热点问题，各种结构的深度神经网络模型不断涌现。然而深度学习的成功实践主要集中在计算机视觉和自然语言处理等领域，针对深度学习技术在排序问题中的应用的相关研究较少，发展空间很大。此外，在特征抽取方面，用户行为特征是一类广泛存在于各应用场景中的特征数据。例如，在搜索引擎中，用户具有历史查询记录和浏览行为；在推荐系统中，用户具有资讯阅读、商品浏览购买等行为记录。此类数据的共同特点是具有较强的时序特性，且能在一定程度上反应用户的偏好习惯。因此，研究如何处理用户行为数据十分重要，也是特征抽取阶段需要解决的重点问题之一。

本课题的目标是研究一种基于深度学习的排序模型。为了提高排序结果的准确度，本课题将深度神经网络模型融入排序模型，弥补了传统统计算法模型的不足。此外，经过对数据特征的充分理解和分析，本课题提出使用循环神经网络模型对用户历史行为数据进行处理，用训练得到的网络参数表示用户历史行为特征，并作为输入特征向量加入排序模型参与训练。为了验证所提出排序模型的可行性和有效性，本课题设计并实现了针对地铁交易数据的单边交易识别模型。地铁单边交易指进站出站信息存在缺失的交易记录，当前对于此类交易记录的处理方式通常是按照最长距离计算票价，这种方法缺乏合理性，容易造成用户损失而导致发生纠纷和投诉。本课题利用所提出的排序模型，根据用户的历史出行行为和地铁站点特征数据，对各候选站点按照其可能性进行排序，填补缺失信息，从而进行合理计费。该方法有助于提升交通行业的服务质量，提高用户满意度。同时，本课题所提出的模型也可以用于完成信息检索、推荐系统和计算广告等领域中的排序任务，具有较强的通用性和可扩展性。综上所述，本课题具有十分重要的研究意义和非常广阔的应用前景。

## 硕士研究生期间的工作总结

本人自2012年10月进入北京邮电大学网络与交换国家重点实验室交换与智能控制中心以来，参与了 “基于微博的金融资讯检索平台”和“基于云的金融数据服务中心”两个校企合作项目。在项目中，本人参与了多个模块的研究、设计和开发工作，负责完成的主要内容包括：

1. 利用微博开放API接口[2]对新浪，腾讯[3]和搜狐[4]三大微博平台的微博进行抓取和更新，协助完成系统设计，完成代码的编写和测试，并参与相关设计文档的撰写。
2. 基于金融微博的短文本分类系统的设计与实现，该系统是“基于微博的金融资讯平台”项目的的子系统，属于数据分析模块，为检索平台的类别检索提供基本支持。根据微博的特征，完成设计并实现了基于情感词典的短文本分类算法，并发表期刊论文一篇，EI检索。
3. 基于PDF文档的表格识别与提取算法的设计与实现。该算法应用在“基于云的金融数据服务中心”项目的数据采集系统。将非结构化的表格数据转换成结构化的Json格式数据，供平台的上层应用使用。由于PDF结构特殊，表格提取存在一定难度。主要难点包括表格基本特征提取，表格结构识别，表格还原，复杂表头处理等。

此外，本人还调研了云计算技术、搭建云平台环境，参与撰写多项技术解决方案、设计文档、专利等。本论文的工作将在第3点的基础上进行扩展，提出一种通用的基于PDF格式文档的表格提取方法。

## 论文组织结构

本论文的组织结构如下表所示：

|  |  |
| --- | --- |
| 第一章 | 绪论：研究背景，研究意义，工作总结，论文结构概述 |
| 第二章 | 论文背景理论及相关技术介绍 |
| 第三章 | 常见表格数据抽取方法的介绍与分析 |
| 第四章 | PDF文件的表格数据抽取关键问题的研究 |
| 第五章 | PDF表格数据抽取系统的设计和实现 |
| 第六章 | 总结和展望 |
| 附录 | 包含参考文献、致谢、攻读学位期间发表的学术论文等 |

表1-1 论文组织结构表

第一章：主要介绍了本课题提出的背景和主要的研究意义。同时对本人在研究生期间的工作进行了总结。最后对论文的整体组织结构进行了梳理和总结。

第二章：主要介绍了PDF格式的基本结构，从物理组织结构和逻辑组织结构两个方面加以介绍。同时用一个例子对PDF的文档格式加以说明。第二部分介绍了本课题使用的类库PDFBox。PDFBox是本课题提取PDF表格基本特征所使用的开源类库，由于涉及到修改源码，因此对其组织结构也进行了介绍。

第三章：主要介绍了基本的表格数据抽取方法，特征识别上介绍了基于Web，图像和PDF的三种典型方法。同时分析了表格抽取方法的主要难点，具体包括表格结构识别，复杂表头处理以及无框线表格的还原等。

第四章：针对第三章提出的问题，第四章给出了本课题的处理方案。本章是论文的重要章节，包含了本课题表格识别和抽取的关键问题。涉及基于PDFBox的表格特征提取，表格还原，表格结构识别等内容。

第五章：第五章是本方法的一个具体应用。即针对PDF财务报表数据的抽取系统的设计和实现。给出了系统的整体设计架构图，流程图，层次结构等。同时展示了抽取数据的HTML和JSON格式，并对PDF表格识别方法的准确性和有效性进行了测试和评估。

第六章：对论文的总结，介绍了本论文的创新点。同时展望未来工作，提出了可能的改进点及可能的改进方案。

# 背景理论及相关技术介绍

本章对深度学习相关技术进行了介绍，主要包括深度学习概述，典型神经网络结构和目前比较流行的深度学习框架三个部分。第一部分介绍了深度学习概念的提出及发展历程，同时将深度学习与传统统计学习方法进行了对比分析；网络结构介绍部分重点介绍了目前应用最为广泛的三种神经网络结构：全连接深度神经网络、卷积神经网络和循环神经网络；第三部分介绍了TensorFlow、Caffe等当前主流的深度学习框架，同时分析说明了每种框架的优缺点。

## 深度学习简介

### 深度学习概述

深度学习（Deep Learning， DL）的概念，由机器学习顶级专家、加拿大多伦多大学教授Geoffrey Hinton等人于2006年提出，指基于样本数据并通过一定的训练方法得到包含多个层级的深度网络结构的机器学习过程。

神经网络技术起源于20世纪五、六十年代，当时称为感知机（Perceptron）。感知机的模型结构包括输入层、输出层和一个隐含层，特征向量由输入层输入，经过隐含层的变换到达输出层，并由输出层得到分类结果。早期感知机模型存在一个严重的问题，即无法模拟复杂的函数，甚至无法表示异或计算。随着技术的发展，Hinton、LeCun等人于20世纪80年代发明了多层感知机（Multilayer Perceptron，MLP），克服了这一问题。多层感知机是含有多个隐含层的感知机，通过增加非线性操作的次数实现对复杂函数的拟合。多层感知机解决了此前无法表示异或逻辑的缺陷，提高了模型的表达能力，但同时也带来了网络参数求解易陷入局部最优、梯度消失等严重的问题。另外，随着网络层数的加深，权值参数的数量成倍增长，受限于当时的计算能力水平，训练神经网络模型十分困难，这导致神经网络的相关研究一度陷入低谷。直到2006年，Hinton及其学生Ruslan在世界顶级学术期刊《科学》上发表论文，提出了两个主要观点：第一，含有多个隐含层的人工神经网络模型有很强的表征学习能力，深度学习模型学习获得的特征数据能够更加本质的代表原数据；第二，针对深度神经网络训练难以达到全局最优的问题，可以使用逐层训练的方法解决。这一研究成果引发了深度学习在研究和应用领域的发展热潮，由此，深度学习逐步发展为机器学习的一个新领域。近年来，Google旗下DeepMind公司开发的围棋人工智能程序AlphaGo战胜多位世界冠军；百度公司开发的智能机器人小度，在最强大脑节目的人脸识别比赛中战胜了此前的人类冠军，这些成绩充分证明了深度学习的强大能力，同时也让普通民众直观感受到了深度学习和人工智能技术，促进了深度学习的进一步发展。

深度学习模型和支持向量机（Support Vector Machine， SVM）、提升方法（Boosting）、最大熵方法等 “浅层学习”方法相比，主要区别在于深度学习模型结构包含更多的层次，并且明确强调表征学习的重要性。深度学习模型通常包含5层以上的隐含层，多个隐含层增加了非线性操作的次数，能够使用更少的参数拟合高度非线性函数，这也是深度学习的“深度”所在。在特征提取方面，浅层学习主要依靠人工经验提取样本的特征，如使用核函数或手工编码等，不仅需要具备专业知识、耗费大量计算资源，而且模型学习得到的是没有层次结构的单层特征；而深度学习则是通过对原始信号进行逐层特征变换，实现样本在原空间的特征表示到新的特征空间的变换映射，自动学习得到层次化的特征表示。深度学习模型在语音和图像识别等领域取得的效果，已经远远超过此前的相关技术。例如，谷歌公司和微软研究院的语音识别方向专家先后采用深度神经网络技术，将语音识别的错误率降低了20%-30%；在计算机视觉领域图像识别应用的著名项目ImageNet中，深度神经网络模型将原来的错误率降低了9%。此外，Google、Facebook等公司也开始将深度神经网络模型应用在推荐系统、广告点击率预估等业务场景中，并取得了一定的成效。

### 神经网络结构介绍

如上所述，在神经网络技术的发展历程中，为了针对性的解决其应用在各类问题中时所面临的问题，研究人员设计发明了多种网络结构。本节对其中三种最基础，同时也是应用最广泛的网络结构进行介绍，具体包括深度神经网络（DNN）、卷积神经网络（CNN）和循环神经网络（RNN）。

* 深度神经网络

在当前的实际应用中所使用的深度神经网络往往是多种神经网络结构的融合，包括卷积层和LSTM单元等。本节所介绍的深度神经网络特指具有全连接的神经元结构的网络，也可以称为多层感知机。全连接深度神经网络结构如图所示。



多层感知机网络结构

传统神经网络模型通常使用Sigmoid或Tanh等连续函数作为激活函数模拟神经元对激励的响应，并使用Werbos发明的反向传播（Backpropagation，BP）算法进行训练。神经网络的层数直接决定了模型对复杂函数的拟合能力，然而随着网络层数的增加，优化函数越来越容易陷入局部最优解，同时局部最优解也会越来越偏离真正的全局最优解。另外一个不可忽视的重要问题是当神经网络层数加深时，“梯度消失”现象变得更加严重。具体来说，当使用Sigmoid函数作为神经元的激活函数时，对于幅度为1的信号，在使用反向传播算法计算梯度时，每传递一层，信号衰减为原来的0.25。在这种情况下，当神经网络模型包含多个隐含层时，就会导致低层由于梯度的指数衰减而接受不到有效的训练信号。

为了解决神经网络模型训练容易陷入局部最优解的问题，Hinton于2006年提出了无监督预训练优化网络权值初值的方法，成功将网络的隐含层数量扩展到了7层。具体而言，预训练过程使用贪心思想，逐层训练模型，将上层训练好的结果作为下层的初始化参数，最终再进行权值微调。这一研究成果使神经网络真正意义上有了“深度”，并由此开始了深度学习的热潮。同时，为了克服梯度消失问题，研究人员提出了使用ReLU、Softplus、Maxout等函数替代Sigmoid函数作为神经元的激活函数，并逐步形成了现在深度神经网络的基本结构。相比于Sigmoid和Tanh函数，ReLU等函数具有单侧抑制、相对宽阔的兴奋边界和稀疏激活性等特点，能够有效解决神经网络模型存在的容易过拟合等问题。



各种激活函数对比

* 卷积神经网络

在全连接神经网络结构中，相邻层的神经元之间都能够形成连接，由此产生的一个严重问题是参数数量的爆炸增长。举例来说，假设以一幅像素为1K\*1K的图像作为输入，当使用全连接结构的神经网络模型时，隐含层含有1M个节点，对应10^12个权值参数需要训练。训练包含如此巨大数量的参数的神经网络模型，不仅需要耗费大量的时间，同时还会导致过拟合、容易陷入局部最优等严重问题。

20世纪60年代，Hubel和Wiesel在研究猫脑皮层中用于局部敏感和方向选择的神经元时发现其独特的网络结构可以有效降低前馈神经网络的复杂性。受此成果的启发，研究人员继而提出了卷积神经网络的概念。卷积神经网络有两个重要的特点，局部感受野和权值共享。区别于普通神经网络结构，卷积神经网络包含了一个由卷积层和子采样层构成的特征抽取器。在卷积层中，一个神经元只与邻层的部分神经元相连，即局部感受野。卷积神经网络的一个卷积层通常包含多个特征平面，其中每个特征平面由一系列矩形排列的神经元构成，同一特征平面中所包含的神经元具有相同的权值参数，共享的权值即卷积核。子采样，又称为池化（Pooling），是在特征向量上进行的一种聚合操作，通常包括均值子采样和最大值子采样两种形式。子采样可以视为一种特殊的卷积操作。局部连接和权值共享大大减少了神经网络中的参数个数，降低了模型的复杂度，同时也降低了过拟合的风险。

卷积神经网络模型由于其自身结构特点，非常适用于处理图像特征，因此被广泛应用在计算机视觉领域中的图片分类、图像分割等任务中，并且取得了巨大的成功。随着相关研究的深入，卷积神经网络衍生出了众多网络结构，其中比较著名的包括Lenet（1986年）、Alexnet（2012年）、GoogleNet（2014年）、VGG（2014年）、ResNet（2015年）和DenseNet（2017年）等。

* 循环神经网络

全连接结构的深度神经网络还存在另一个问题，即无法对时间序列上的变化进行建模。然而在机器翻译、语音识别、趋势预测等应用中，样本出现的时间顺序及上下文关联性是十分重要的。为了适应上述需求，研究人员提出了另一种神经网络结构：循环神经网络。在循环神经网络中，一个序列当前的输出不仅与当前的输入有关，还取决于之前的输出。具体表现为网络会对当前时刻之前的信息进行记忆，并参与到对当前输出的计算中。区别于普通神经网络中同层节点之间无连接的结构，在循环神经网络中，同层神经元之间存在连接，同时将输入层的输出和隐含层上一时刻的输出共同作为当前时刻隐含层的输入。



循环神经网络典型结构及在时间维度上的展开

与传统神经网络算法相同，对循环神经网络的训练同样使用误差反向传播算法，区别在于如果对循环神经网络进行时间维度上的展开，其参数是共享的，同时梯度误差不仅依赖当前网络的状态，还依赖于此前网络的状态。针对循环神经网络的学习算法称为Backpropagation Through Time（BPTT）。然而BPTT算法存在的一个重要问题是其无法解决长时依赖问题，即在当前的输出与之前较长的一段序列相关时，使用BPTT算法无法对网络进行有效的训练。这是因为当循环神经网络在时间维度上展开的层数过多时，BPTT同样会产生普通神经网络使用BP算法训练时存在的梯度消失或梯度爆炸问题。为了解决梯度在时间维度上无法进行有效传播的问题，研究者们提出了多种改进的循环神经网络结构，如长短时记忆单元LSTM和门控循环单元GRU。

LSTM由Hochreiter和Schmidhuber于1997年提出，并由Alex Graves进行了改良和推广。LSTM使用被称为Cell的结构来保存当前输入之前的状态，并使用精心设计的称为门（Gate）的结构来增加或去除Cell中的信息。LSTM单元中共包括三个门结构，分别是输入门（Input Gate）、输出门（Output Gate）和忘记门（Forget Gate）。多项研究成果已经证明LSTM网络结构对于解决长序列依赖问题十分有效，在自然语言处理中的词向量表达、词性标注、机器翻译和语音识别及图像描述生成等应用中都取得了显著的效果。GRU是LSTM的一种变体，由Cho等人于2014年提出。GRU将LSTM中的忘记门和输入门合并成了单一的更新门，简化了模型结构，同样得到了广泛的应用。除GRU之外，还存在多种LSTM变体，如Koutnik等人提出的Clockwork RNN（2014年），Yao等人提出的Depth Gated RNN（2015年）等。近年来，注意力（Attention）机制成为了循环神经网络研究的新热点，也取得了丰富的研究成果。





LSTM及GRU结构示意图

### 深度学习框架介绍

深度学习的火热发展及其取得的众多瞩目成绩，吸引了Google、Facebook、Microsoft等科技巨头围绕深度学习重点投资了一系列新兴项目，并因此推动了一批开源深度学习框架的诞生和发展。目前研究人员使用的主流深度学习框架主要包括TensorFlow、Torch、Caffe、Theano、Deeplearning4j、Keras等。这些深度学习框架各具特点，被广泛应用于计算机视觉、语音识别和自然语言处理等领域，并取得了十分优秀的效果。下面对这些深度学习框架进行简要介绍。

* TensorFlow

TensorFlow最初是由Google Brain团队基于DistBelief，针对机器学习和深度神经网络进行研究所研发的第二代人工智能学习系统，目前已经开源。

作为一款数学计算软件，TensorFlow使用数据流图的形式进行计算。数据流图中的节点表示数学运算，线条则表示多维数据数组，即张量之间的交互。TensorFlow的命名来源于其运行原理：张量（tensor）从数据流图的一端流动（flow）到另一端的计算过程。



Tensorflow流程示意图

TensorFlow的主要特点包括适应性强、可扩展性强、自动差分、计算高效等。具体表现在TensorFlow不只提供神经网络库，还支持用户自主构建流图、编写内层循环代码驱动计算；可以应用在各种类型的设备上，如GPU、云平台、移动设备等；提供包括Python和C++在内的多种编程语言的接口支持；TensorFlow可以将不同计算单元分配给不同的设备执行，能够充分利用硬件资源。

* Torch

Torch是一个已有十余年历史的科学计算框架，采用Lua语言编写，支持大量机器学习算法。Facebook开源了大量建立在Torch上的深度学习模块和扩展功能，促进了Torch的进一步发展。

Torch框架的优势在于使用其构建模型非常简单，易于使用。框架高度模块化，同时提供快速高效的GPU支持。Torch框架提供跨平台支持，可以嵌入到iOS、Android和FPGA等后端的接口。

* Caffe

Caffe全称Convolutional Architecture for Fast Feature Embedding，是一个高效清晰的开源深度学习框架，由加州大学伯克利分校的贾扬清博士开发，目前由伯克利视觉学中心进行维护。

Caffe中的计算是以层的形式表示的，其中每一层的基本流程为接受输入数据，进行计算，输出计算结果。其中计算包括由输入计算输出的前向传播和梯度的反向传播。定义好每一层的计算函数后，就可以把多个层连接成网络进行训练。

Caffe的优点包括其模型与对应的优化都是以文本形式而非代码形式给出，十分容易上手；计算速度快，可用于海量数据的训练，特别适用于计算机视觉领域；开放性好，提供公开的代码和参考模型，同时提供良好的社区支持。不足之处则包括文档匮乏，需要大量依赖包、安装过于复杂等。

* Theano

Theano于2008年诞生在蒙特利尔理工学院，其核心是一个数学表达式编译器，专门为深度学习中处理大型神经网络算法所需要的计算而设计，被认为是深度学习研究和开发的行业标准。Theano在其发展过程中派生出了大量深度学习软件包，如Blocks和Keras等。

Theano的优势体现在其集成NumPy并使用其中的ndarray数据类型、使用GPU加速计算、提供速度和稳定性优化、可以动态生成C代码、具备较强的灵活性等。而缺点则包括偏向底层，模型实现比较复杂，不支持多GPU和水平扩展等。

* Keras

Keras是一个完全使用Python编写而成的高层神经网络API库，支持的后端包括TensorFlow、Theano和CNTK。Keras的设计目标是支持快速实验，方便进行简易快速的原型设计，十分适合用于先进的研究工作。

Keras提供一致而简洁的API，可以大大减少工作量，用户体验优秀；模块性强，网络层、损失函数、优化器、激活函数、正则化方法等都作为独立模块，便于构建模型；支持以编写类或函数的方式创建新模块，具备很强的扩展性。

* Deeplearning4j

Deeplearning4j由创业公司Skymind于2014年6月发布，是一个面向实际生产环境和商业应用的具备高成熟度的深度学习开源库，也是首个商用级别的深度学习开源库。顾名思义，Deeplearning4j面向Java语言，可与Hadoop和Spark集成，同时支持在移动设备的应用中快速集成深度学习功能。

经过对上述各框架各自优缺点的分析对比，并充分考虑易用性、可扩展性及计算效率等因素，本课题最终选择使用TensorFlow实现课题研究中所涉及到的深度神经网络模型。

## 本章小结

本章主要介绍了深度学习的相关概念技术，包括深度学习的发展历程，经典神经网络模型的结构等，并对当前主流的深度学习框架进行了简要介绍和对比分析，为后续章节中涉及的算法设计、模型选取、系统实现等内容做了铺垫。

# 常用排序方法的介绍与分析

本章首先对现有的各种排序方法进行了介绍，并分析了不同方法各自的特点及适用场景。之后总结了排序模型中常用的算法模型及各自的优缺点，并介绍了几种常用的评价指标。最后概括分析了排序问题中的几个重点难点问题，包括特征数据的提取，时序类特征数据的处理，排序模型的优化等内容。在后续章节中将具体介绍本课题解决上述问题所采用的技术方案，并针对地铁单边交易识别的应用场景进行具体排序模型的设计与实现。

## 常用排序方法介绍

排序问题最初是作为信息检索的核心研究问题而被人们所关注。在搜索引擎等信息检索系统中，问题可以被抽象定义为给定查询Q和文档集合S={D1, D2, …, Dn}，按照某种规则返回S中文档的一个序列。传统的排序模型主要可以分为两类：相关度排序模型和重要性排序模型。其中，相关度排序模型根据文档D与查询Q之间的相似度对文档进行排序，通常做法是利用词频、逆文档频率、文档长度等因子，由人工确定排序公式，此类模型包括TF-IDF、隐语义分析（Latent Semantic Analysis）、BM25、基于语言建模的信息检索（LMIR）模型等。重要性模型则不考虑文档和查询之间的相似性，而是将包含查询信息的文档按照其重要程度进行排序。文档的重要程度通常根据文档之间的图结构来计算，典型算法如PageRank、HITS、TrustRank等。

然而，无论是相关度排序模型还是重要性排序模型，都存在一个严重的问题，即只考虑了某一方面的因素（相关度或重要性），而没有对各种因素进行综合考量。随着机器学习技术的发展，研究者们开始使用统计学习方法解决排序问题，即LTR。LTR将查询关键词、文档信息、甚至是传统模型的结果（如查询与文档的相似度、PageRank值等）等数据作为输入特征向量，将由人工标注或从日志数据中挖掘出的文档排序列表作为输出，通过监督学习训练机器学习模型进行排序学习。与传统排序模型相比，LTR训练得到的排序模型使用更多的特征拟合排序公式，充分考虑了各方面因素，同时由算法根据历史数据学习得到的模型参数，也比依赖人工的方法更加客观和高效。随着相关研究的深入，LTR已经不仅仅局限于信息检索领域，在推荐系统和计算广告等领域中，排序学习同样得到了广泛的应用，排序对象也从单一的网页等文档对象扩展为包括商品、新闻资讯、广告内容等在内的各类实体。现在，排序学习已经演化为一类独立问题，并且发展成为研究人员所关注的热点研究方向。

## LTR模型训练方法

LTR的主要流程包括收集训练数据、特征抽取转换、训练排序模型和模型应用等步骤。其中，排序模型的设计和训练是LTR的核心，也是技术难点所在。按照原理的不同，LTR的模型训练方法可以分为三类：PointWise方法、PairWise方法和ListWise方法。下面分别对三种方法的思想原理及特点进行分析说明。

### PointWise方法

### PairWise方法

### ListWise方法

## 本章小结

本章主要对表格提取的几个关键步骤所涉及的方法以及技术难点进行了总结和分析。在表格特征识别方面，对三种不同的媒介（Web，图像，PDF文档）下的表格特征识别进行了介绍；在表格数据抽取部分，对几个关键的技术难点问题进行了介绍。包括了表格结构的识别、复杂表头的处理以及无框线表格的还原等问题。为第四章基于PDF的表格数据抽取做好铺垫。

# PDF文件的表格数据抽取关键问题研究

本章主要对基于PDFBox的PDF表格抽取的关键问题进行介绍。具体包括表格特征提取，表格还原，表格结构识别等。其中表格特征提取使用了开源的项目PDFBox，为了完成表格框线的识别，本系统修改了PDFBox的源码。表格还原主要包括线条过滤，表格定位，复杂表头处理等问题。表格的结构识别则提出了基于表格内容和表格结构的识别方法。

## 基于PDFBox的表格特征提取

### PDF字符流中表格的基本格式介绍[5]

由于PDF的字符流格式比较复杂，为了方便理解，截取PDF中某表格的部分字符流进行说明。本系统就是通过PDFBox从下述的字符流中提取相应的文字和线条信息的。

图4-1-1 PDF中文字流示意 图4-1-2 PDF中线条流示意

PDF使用特定的字符表示文字流，线条的流的开始和结束。对于文字以BT开始，ET结束。图中的红色框线部分是表格某单元格内的文字，其中Tj内的信息是文字的字符信息。Tm左侧信息表示的是位置信息。

右图的绿色区域则表示了框线的信息。‘re’表示矩形路径构造的开始。[89.96 597.58,229.46 0.72]分表表示了矩形的位置和大小。其中89.24表示矩形左下角x轴坐标，597.58表示矩形左下角y轴坐标，这些位置信息都是绝对位置信息，由此可以确定该矩形的位置。229.46表示长度，0.72表示高度。因此该信息表示了一条长为229.46的线条，且该线条呈水平方向。

### PDFBox中文字和线条的处理介绍

在PDFBox中，对于PDF内的文字和线条都经过了重新的处理和封装。文字和线条在PDFBox中都存储在CosStream对象中。该对象继承自CosDictionary类。这些对象均处于PDFBox的CosModel层，即PDFBox的最低层，该层负责完成对PDF基本结构的封装。PDFBox通过PDFStreamParser解析CosStream对象，针对不同的类型进行分别处理。下面是PDFBox低层关键部分的代码：

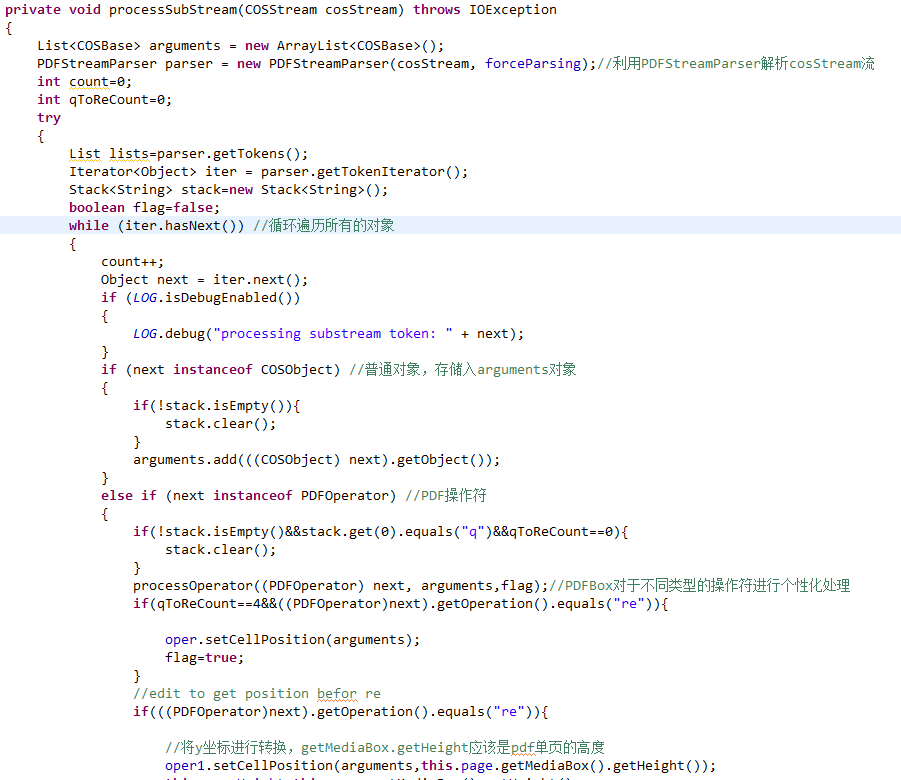


图4-2 PDFBox低层对象遍历处理截图

如图，PDFBox利用processSubStream函数完成底层各个对象的个性处理。在函数内，PDFStreamParser负责解析CosStream下对应的各个对象，并利用processOperator函数完成不同类型操作符的个性化处理。由于PDFBox并未对线条类对象进行处理，因此本系统修改了该部分的代码。如下图：



图4-3 修改PDFBox低层代码实现线条信息提取部分截图

依据PDF中线条的特殊格式，通过识别操作符“q”,“Q”,“re”等对线条信息进行提取。argument对象用于存储线条的位置信息[xpos,ypos,width,height],当碰到标识符“re”则将argument中的线条信息封装存储。

### 基于PDFBox的文字信息提取与封装

由于PDFBox已经对文字的信息进行了较好的封装，本系统针对表格还原的需求对该部分的文字信息进行了进一步的过滤和再封装。只保留对表格还原有效的特征量，即文字的字符信息和文字的位置信息。下图文字流在PDFBox中的表示情况以及本系统对文字流的封装情况。



图4-4 基于PDFBox的文字信息提取与封装

原始的文字信息包括了字符(Character)，位置(xpos,ypos)，字体(font)，大小(size)，颜色(color)，方向(PageRotation)，间隔(PageWidthValue,PageHeightValue)等。由于本系统只需要文字的字符和位置信息，因此只保留<character,xpos,ypos>信息表示单个文字的全部特征。

### 基于PDFBox的线条信息提取与封装

在4.2.1节已经对线条的格式进行了大致的介绍，由于PDFBox不支持线条的提取，因此需要修改PDFBox源码完成线条信息的提取和封装。前面已经介绍了线条的构造标识符为‘re’（矩形线条绘制路径），下图是对线条信息的提取和封装流程说明。在处理过程中通过识别‘re’标识符，提取线条信息并封装。



图4-5 基于PDFBox的线条信息提取与封装

上图是对线条提取与封装流程的简要说明。其中Cosstream是PDFBox对PDF的字符流的封装格式，在处理的时候将其转换成Iterator格式进行遍历。当碰到cosObject结构表明是位置信息应该保存到argument的可变数组中。当碰到PDFOperator结构时检查是否是re，如果是则提取前一个argument信息，即为该线条的位置信息，反之则舍弃不处理。最终线条信息成Line Information格式，包含横轴信息(xpos)，纵轴信息(ypos)，线条长度(width)，线条高度(height)。因此每条线段的特征向量为<xpos,ypos,width,height>。

## 表格还原

借助PDFBox，我们已经获得了表格的基本特征信息，即位置信息和文字信息。下面介绍如何利用基本信息还原真实的二维表格，为了方便快速提取表格特定行号和列号的数据，本系统采用二维数组来存储表格信息。整个还原过程分为三个部分，首先通过线条信息确定表格的基本维度，即二维数组的行数和列数，同时确定各行各列的位置坐标。然后将对应的文字信息根据位置坐标填充入对应的二维数组中。最后由于可能存在一表多页的情况，需要对各页的二维数组进行进一步的合并。下面分别给出各部分的流程图。

### 基于线条信息还原基本二维表结构

常见的PDF中表格形态如下图所示，即单页PDF可能包含多张表格。要利用横纵线条还原基本的表格形态需要特别注意线条间的交错情况，避免将不同表格的线条错判为同一表格。因此在设计表格还原算法时，优先处理纵向的横线信息。

如图4-6所示，PDF Page中有两个表格，观察虚线条1,2,3.其中1,3属于表格1而2属于表格2.为了防止线条错位，本算法优先处理纵向横线。如图所示，在表格1中y0和y1表示横向线条的垂直方向坐标。h0是纵向线条的高度。在判断y0和y1是否属于同一表格的充分条件是。其中是可容忍的偏差阈值。对于线条y3和y4，由于不存在纵向线条h’使得成立，因此y3和y4应该属于不同的表格，即y3和y4是两个表格的分离线条。



图4-6 表格栅格化示意图

基于线条的表格还原算法伪代码如下：

|  |
| --- |
| 算法作用：根据线条LineInformation信息确定单页表格坐标刻度。  输入量：ArrayList<LineInformation> 单页横向和纵向全部线条信息  输出量：ArrayList<ArrayList<Float>> PageTableXpos 单页全部表格纵线刻度  ArrayList<ArrayList<Float>> PageTableYpos 单页全部表格横线刻度 |
| 1 Begin  2 Filter LineInformation //过滤无关线条  3 Sort LineInformation which(width>>height) //排序纵向横线条，按ypos排序  4 foreach line (width>>height)  5 if(|pre\_ypos+pre\_height-ypos|<tolerance) //两条线条属于同一个表格  6 YposArray🡨ypos  7 else //两条线条不属于同一表格  8 PageTableYpos🡨YposArray;  9 New YposArray;  10 Sort LineInformation which(height>>width)//排序横向竖线条，按xpos排序  11 foreach line (height>>width)  12 foreach PageTableYpos  13 if(ypos∈PageTableYpos[i])  14 PageTableXpos[i]🡨xpos  15 End |

本算法的最终目的是确定单页PDF中的表格数量，并将各个线条的刻度记录在ArrayList<<ArrayList<Float>>PageTableXpos和ArrayList<<ArrayList<Float>>PageTableYpos中。两者为可变长数据，长度与单页PDF中表格数量一致。每个表格对应了横向和纵向的全部刻度坐标。依据分析，为了避免表格线条的错判问题，本算法优先处理纵向的横线。首先针对横线进行排序，然后根据公式|pre\_ypos+pre\_height-ypos|<tolerance确定表格数目以及各个表格内的纵向刻度。接着分表格依次处理横向的竖线。进而确定各个表格的横向刻度从而得到PageTableXpos和PageTableYpos的所有值。下面分别给出纵向横线和横向纵向刻度分析过程的流程图：



图4-7 基于线条还原二维表纵向横线信息流程

上图流程可以完成单页PDF表格个数的确定，同时统计好每个表格的垂直方向（ypos）刻度情况。通过ypos即可确定每个表格含有的行数以及每行具体的位置信息。所有的表格信息通过PageTableYpos进行存储，包含该页表格的纵轴位置信息。

线条信息过滤主要是滤除重复线条和垃圾线条。例如底层线条信息可能包括如下形式[319.42 597.52 0.72 0.059998]，此类线条的长和高都接近于0，因此不是一条具体点线条，而是点信息，此类线条对于确定表格的轮廓没有帮助，属于垃圾信息，需要滤除。重复线条是指位置差异小于给定阈值的线条，此类线条实际表示的位置基本相同，可以滤除。

按照ypos进行排序主要用于确定表格垂直方向的刻度信息。利用pre\_ypos+preheight表示期待的线条的ypos信息，如果和当前的ypos相差量大于容忍误差(tolerance)，说明前面的线条和当前线条不属于同一个表格，当前线条从属于新表格。因此以前的表格信息可以存入PageTableYpos，同时构造一个新数组存储当前ypos信息。因此，对于|pre\_ypos+preheight-ypos|<tolerance的情况，表明当前线条仍然属于当前表格，可以保持ypos信息。反之则开始新的表格，将当前表格存入PageTableYpos集合中，同时构造一个新的数组存储新开始的表格的ypos信息。

下面接着确定水平方向的坐标刻度情况。对于水平刻度的确定需要结合垂直刻度来确定当前刻度从属的表格数。从而避免不同表格的刻度划分到相同表格的情况。水平方向的刻度信息存储于ArrayList<ArrayList<xpos>>PageTableXpos中，下面是具体的流程图：



图4-8 基于线条还原二维表横向竖线信息流程

首先对xpos进行排序。由于流程1已经确定了每个表格的ypos信息，因此比较ypos可以确定线条属于哪个表格，将不重复的xpos信息加入到对应的xpos数组下，最终可以得到整个页面的PageTableXpos信息。之所有要先确定ypos再确定xpos是因为不同表格的xpos很可能是交错的，如果不确定ypos，很容易将不同表格的xpos错误的估计到一张表格里。

通过横纵向的排序和比较，可以初步获取单页PDF中的表格数量以及每个表格的行列信息。PageTableYpos.size(),PageTableXpos.size()相等并为单页PDF中表格的数量，而每个xpos数组则反映了列数及列位置信息，每个ypos数组则反映了行数及行位置信息。示意图如图4-4的表格栅格化示意图所示。图中PageTableXpos和PageTableYpos存储了单页PDF中表格的全部物理位置信息。

### 基于位置信息填充表格文字



图4-9 表格文字填充示意图

单个文字特征量用向量< character,xpos,ypos>表示。通过比较文字的位置信息<xpos,ypos>以及表格线条刻度信息PageTableXpos和PageTableYpos判定单个文字所处单元格。对于同一单元格中的文字，仍然需要根据<xpos,ypos>信息加以排序。为了便于在单元格内排序，此处定义二维数组的类型为TreeSet类型，由于TreeSet类型元素是有序的，因此只需要将文字填充到对应单元格内，单元格内的文字将根据TreeSet制定的排序规则自动实现排序。TreeSet类型需要实现Comparator接口，下面是实现的Comparator接口的关键代码：

|  |
| --- |
| 类名称：CompareWord  作用：通过比较文字xpos,ypos两个维度的位置信息对同一单元格内的文字排序 |
| 1 class CompareWord implements Comparator<CharacterInformation>{  2 public int compare(CharacterInformation a, CharacterInformation b){  3 if(a.getY()<b.getY()){//首先判断ypos位置  4 return -1;  5 }else if(a.getY()==b.getY()){//当处于同一水平位置，再判断xpos位置  6 if(a.getX()<b.getX())  7 return -1;  8 else if(a.getX()==b.getX())  9 return 0;  10 else  11 return 1;  12 }else{  13 return 1;  14 }  15 }  16 } |

单元格内文字的排序主要依据两个维度xpos和ypos。如图4-9示意，文字W1，W2，W3。对于W1和W2，两者的ypos相同，因此考虑xpos的情况。显然W1的xpos小于W2因此W1被排在W2的前面。而对于文字W2和W3。优先比较ypos，由于W2的ypos小于W3的ypos，因此W2排在W3的前面。可见在相同单元格的文字排序规则中，优先考虑的是ypos的关系，当ypos关系相同时再考虑xpos的位置关系。由于二维数组采用了TreeSet结构，因此相同单元格内的排序是默认进行的。在处理的时候，只需要根据文字的<xpos,ypos>坐标判断到特定的单元格内即可。对于单元格内的排序是自动完成的。下面给出表格字符填充的流程图



图4-10 文字填充基本流程

遍历单页PDF中所有的文字信息，对于单个文字信息，已经封装在CharacterInformation类中。对比xpos，ypos以及垂直和水平刻度信息可以确定文字的基本位置。如果该文字不属于任何的表则不做任何处理，对于表内的文字，确定其基本位置将文字填充入二维数组。由于该类已经实现了文字信息的排序，因此在单元格内部填充的文字信息已经是有序的了。

### 一表多页的复杂表格合并

由于PDF单页大小是固定的，因此存在单个表格分布在不同PDF页面的复杂情况，因此需要对各页的PDF进行判断，对于可能的跨页表格进行合并。基本的算法伪代码如下：

|  |
| --- |
| 函数名称：MergeForms  作用：将制定页数的表格输出，最后一个表格如果有跨页情况需合并后再输出  输入量：PDF页码  输出量：Form格式的表格 |
| 1 Public ArrayList<Form> MergeForms(int page){  2 ArrayList<Form> curForm=GetFormsByPage(page);  3 If(curForm==null||curForm.size()==0)  4 return curForm;  5 int i=1;  6 //合并可能的表格  7 while(true){  8 ArrayList<Form> tem=GetFormsByPage(page+i);  9 i++;  10 if(tem==null||tem.size()==0)  11 break;  12 else{  13 //该函数用于判断两张表是否可以合并  14 if(isOneForm(curForm.get(curForm.size()-1)),tem.get(0)){  15 //合并两张表，合并后的表存入curForm  16 MergeTwoForm(curForm.get(curForm.size()-1),tem.get(0));  17 if(tem.size()>1)  18 break;//如果该页不止一张表，则不可能继续合并直接退出  19 }else{  20 break;  21 }  22 }  23 }  24 //返回合并之后当前页的全部表格  25 return curForm;  26 }  27 //判断两张表格是否可以合并  28 Private Boolean isOneForm(Form f1,Form f2)  29 {  30 //判断两张表的列数是否一致  31 if(f1.colsize!=f2.colsize)  32 return false; //列数不一致显然不是同一个表格，直接返回false  33 else  34 {  35 if(|f1.downpos-PageDownPos|>tolerance)  36 return false;//第一张表的最后一条线不处于pdf尾部，返回false  37 else  38 {  39 if(|f2.uppos-PageUpPos|>tolerance)  40 return false;// 第二张表格第一条线不处于pdf头，返回false  41 return true;  42 }  43 }  44} |

整个合并的思路如下：

1. 获取指定页面的所有表格。表格数目为0直接返回。
2. 获取Page+i(i=1,2,3…)页的全部表格，如果表格数目为0，直接返回。
3. 利用isOneForm判断两张表是否是一张，false则结束循环。True则合并两张表格，同时继续（2）步骤直到循环结束。
4. isOneForm用于判定两张表是否是同一张表，判断依据包括两张表的列数、第一张表的最后线条的绝对位置、第二张表的第一条线条的绝对位置。

## 表格结构识别

### 复杂表头的识别与转换

表格的表头由于需要表示每个单元格的具体含义有时候会比较复杂，在本文的第三章已经对复杂表头的可能情况进行了总结。由于图形结构的表在实际应用中并不常见，因此本系统只针对树形结构的表进行了处理。处理思路是先根据行列坐标刻度将表格转换为一个树形结构，然后从树的根节点开始深度遍历树直到叶子节点，所生成的路径即为简单表格的一个单元格。下图是一个简单的流程说明图。



图4-11 复杂表头处理示意图

从复杂表头转为简单的表头，首先需要根据表头情况建立一颗树。对于单个树节点给出如下定义：

|  |
| --- |
| TreeNode{ //树节点  CellNode cellinfo; //单元格节点  ArrayList<TreeNode>children; //孩子节点  } |

对于表格中每个单元格的定义如下：

|  |
| --- |
| CellNode{ //单元格节点  String value; //具体文字内容  float leftpos; //左边框线坐标  float rightpos; //右边框线坐标  float uppos; //上边框线坐标  float downpos; //下边框线坐标  } |

需要注意的是，单元格是按照uppost进行排序的，因此表格单页格式分层的，图中用不同颜色表明了表格的层次关系。同时每层的单元格又是按照leftpos排序的。使用两个队列分别存储前一层和当前层的节点信息。以图中树为例，建树流程描述如下：

1. 初始化root节点，该节点不存放有效信息。将root存入queue1.
2. 接着分层遍历表格单元格，如图遍历1,2.此时将node1和node2存入queue2.并将root.children.add(node1),root.children.add(node2).
3. 此时当前层遍历完。将queue2中的元素导入queue1.清空queue2.
4. 继续遍历下一层的3,4,5.通过比较发现node3.leftpos>node1.rightpos,因此que0.poll().接着比较node2，发现node3.rightpos<node2.rightpos&&node3.leftpos>=node2.leftpos，因此node2.children.add(node3).同理将4,5加入node2.更新queue1和queue2.
5. 继续遍历node6完成数的建立。

建树流程图如下：

图4-12 复杂表格建树流程图

队列queue1存放父亲节点，即当前层的上一层。队列queue2存放当前层的节点。当通过uppos的值判断节点的层数。每次到达当前层的最后一个节点则更新queue1的内容，同时清空queue2的值。在同一层，通过判断fanodde.leftpos,fanode.rightpos,node.leftpos,node.rightpos的位置关系判断当前节点的父亲节点，并将其加入到相应父亲节点的子节点中。本算法默认只处理树形结构的表格，如果出现图形结构，则会自动结束并抛出错误异常。

从树结构转换成简单表结构则通过树的深度遍历实现，即从根节点开始向下搜索，直到叶子节点。采用递归的方法不断的遍历孩子节点，当到达叶子节点后，再回退选择新的未遍历的节点进行遍历。深搜算法图形描述如下图所示，图中包含四条路径，分别用不同的颜色标识。

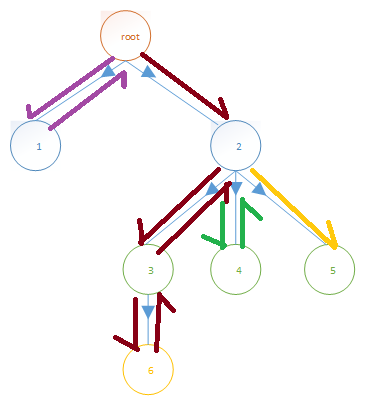


图4-13 复杂表格深度搜索示意图

|  |
| --- |
| 函数名称：DFS  作用：深度搜索表格树  输入量：  node：保存树的根节点  res：保存最终所有路径的全部文字内容结果  sub：保存当前路径的全部节点文字内容 |
| 1 Public void DFS(TreeNode node,ArrayList<ArrayList<String>> res,ArrayList<String> sub){  2 if(node.children==null)  3 return;  4 if(node.children.isEmpty()){  5 ArrayList<String> copy=(ArrayList<String>)sub.clone();  6 res.add(copy);  7 return;  8 }  9 for(int i=0;i<node.children.size();i++){  10 sub.add(node.children.get(i).cellinfo.value);  11 DFS(node.children.get(i),res,sub);  12 sub.remove(sub.size()-1);  13 }  14} |

深度遍历TreeNode的所有节点。深度遍历过程中只存储树节点中的文字信息，即cellinfo.value，将位置信息等忽略。最后res.size()则是简化后表格的列数，同时每个单元格中存储了不同行的文字信息。

### 表格结构类型

在本文的第三章已经介绍了表格的九种结构，其中涉及到较为复杂的嵌套结构。考虑到大部分表格都属于非嵌套结构，同时为了简化识别模型，本系统着重处理以下三种类型的表格结构。



图4-13 三张典型的表格结构

对于a类横向展开型表格，所有的属性都几种在表格的第一行。而b类纵向展开型表格属性则几种在第一列。c类混合展开型表格最为常见，其第一行和第一列均表示属性，而(0,0)单元格内容往往为空。下面将着重讲解本系统对以上三类表格识别方法。

### 表格展开方向识别

为了识别表格的展开特征，本系统首先考虑表格的物理特征，如表所示，这也是表格最基本也是最重要的特征。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 情况 | 首行跨列 | 首列跨行 | 判定 |
| 物理结构特征 | 1 | 是 | 是 | 混合展开型 |
| 2 | 是 | 否 | 横向或混合 |
| 3 | 否 | 是 | 纵向或混合 |
| 4 | 否 | 否 | 待判定 |

表4-1 表格物理特征与表格结构关系表

通过判定跨行或跨列的情况可以对表格的基本结构进行初步的判定，对于情况1已经可以确定其类型，则无需进行后续的判定。对于情况2,3，已经可以确定其横向（纵向）的特征，需要进一步确定其纵向（横向）特征。情况4的表格属于简单表，则需要从内容上对其横向和纵向特征进行判定。本系统主要考虑的表格内容特征如下表所示：

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | 情况 | 说明 |
| 表格内容特征 | 内  容 | 1 | 首行特殊领域词语占单元格比率 |
| 2 | 首列特殊领域词语占单元格比率 |
| 行列  相似  度 | 3 | 首行外其他行的平均长度标准差 |
| 4 | 首列外其他列的平均长度标准差 |

表4-2 表格内容特征与表格结构关系表

采用打分判决思想，从内容和行列相似度上分别进行打分，最后总分的计算按如下公式：

其中A表示基于文字内容的得分，S表示行列相似度上的得分，T表示总得分情况。raw,col,raw\_col表示打分的两个维度，β是可调节的阈值，用于调节文字A和S对最后结果影响权值。当待分析的表格属于某个特定领域时，则提高A的比例有助于提高结果的准确性，反之则将表格当作普通情况处理，加大文字长度的影响权值。表格的最终类型与三维度中得分最高的维度保持一致。如果表格在raw和col上的打分差异不大，则归为混合展开型表格。下面分别介绍A和S的打分规则。

1. **基于单元格内容的打分规则**

此规则主要适用于表格所属领域已知的情况，对于该类表格，其属性值往往是固定。因此可以通过首行，首列中特定属性值的占比来判定表格的类型。例如财务三大报表中固有属性值往往包括（流动资产，非流动资产，资产总计，应付票据，流动负债，所有者权益）等关键属性。因此可以根据该类属性词建立属性词典。然后对表格首行和首列的内容进行分词，统计关键属性词占比最终确定表格的展开类型。基本流程如下图所示：



图4-14 基于内容的展开方式识别流程

基于内容的表格格式识别具有较高的准确率，但由于需要建立个性化的匹配属性词典，因此普遍适用性较差。处理流程的关键是分词以及属性词典的匹配。对于每个单元格内的文本内容进行分词，记录各个单元格中匹配的词语数量。最后使用P1和P2分别记录行列中匹配属性的概率。通过比较P1和P2大小来确定行列展开情况。如果P1和P2匹配概率基本相同，则判定为混合型展开。

1. **基于行列相似度的打分规则**

除了从单元格内容进行判定以外更普遍的方式是根据表格行列方向单元格内文字的数量进行判定。通常情况下横向展开型表格各个列具有较强的相似度。而纵向展开型表格各个行具有较强的相似度。根据本规律，可通过计算表格行列展开下表格均值和方差加以判定。行列均值和方差的计算公式如下：

其中，表示列i中的平均单元格字数。r表示除去首行的表格行数。表示第i列第j行的单元格内的文字数目。表示行i中的平均单元格字数。s表示除去首列的表格列数。表示第i行第j列的单元格内的文字数目。表示列i中的方差，表示行i中的方差，分别依据和得到。

下面使用一个横向展开表格为例加以说明。

|  |  |  |  |
| --- | --- | --- | --- |
| **姓名** | **性别** | **籍贯** | **年龄** |
| 小红 | 女 | 北京 | 20 |
| 王伟强 | 男 | 天津 | 33 |
| 李维维 | 男 | 黑龙江 | 23 |
| 唐煦日辉 | 男 | 安徽 | 20 |
| 李俊基 | 男 | 江西 | 20 |
| 王强 | 男 | 内蒙古 | 33 |
| 何平 | 男 | 新疆 | 23 |
| 艾文 | 女 | 河北 | 20 |

表4-3 横向展开表格示意

依据公式依次计算去除首行首列后各个单元格内的文字数量及各行各列的平均字数及标准差，计算结果如下图所示：



图4-15 去除收行首列的行列均值方差统计图

如图，蓝色框图中统计的是表格各单元格内的字数。橘黄色框图表示的是各行和各列的均值和方差统计结果。则=(0.25+0.25+0.82+0.25+0.25+0.82+0.25+0.25)/8=0.39，=(0+0.43+0)/3=0.14，，因此该表格为横展开的表格。

## 本章小结

本章主要介绍了PDF表格数据抽取系统的关键技术。对于PDF的表格特征提取，本课题选择了PDFBox开源项目辅助实现特征提取。对PDF中的文字信息进行了重新封装。为了获取表格线条信息，修改了PDFBox源码，实现了对PDF中线条信息的提取和封装。表格的还原由于使用了基于线条的还原方式，因此有效的保证了表格还原的准确性。对于表格结构的识别，主要涉及复杂表头识别，表格展开方式识别等常见问题，本章基于PDF表格的特点，分别给出了对应的解决方案。

# PDF表格数据抽取系统设计与实现

前文对PDF表格数据抽取的关键技术进行了详细的介绍，本章将该方法运用于系统中，用于提取PDF文件中的财务报表数据。本章阐述了表格数据抽取系统的设计和实现过程，依次介绍了PDF文档解析，PDF表格还原，PDF表格数据封装等内容，最后从准确性和覆盖率两方面对表格解析情况进行了测试。

## PDF表格数据抽取系统的整体设计

### 系统整体流程设计

本系统主要针对PDF格式文件的表格进行数据抽取。因此在表格特征识别方面主要依据PDF格式的特定来提取表格信息——文字内容，文字位置，线条位置，线条长度等。借助了PDFBox完成此类信息的提取。在表格信息还原上以框线的位置信息作为特征量，通过设计相应算法实现表格基本框架的还原。然后利用表格中的文字位置信息将相应的文字内容填入对应的单元格最终完成整个表格的还原。

当表格的基本形态确定后接下来需要完成的是表格数据的抽取。本系统设计了相应的算法分别完成了表格结构的识别和复杂表头的处理。为了完成非结构化数据的结构化，本系统对复杂的表格进行了拆分处理，实现了复杂表格和简单表格的统一表示。并最终将表格数据封装成JSON的格式真正实现了非结构化数据的结构化。整个处理过程如下图所示：

图5-1 基于PDF的表格数据抽取系统处理流程

### 系统层次描述

根据系统的功能将系统分为如下的三个部分。最底层是PDF文档解析层，主要完成PDF中文字和线条信息的提取和封装。中间是PDF表格还原层，该层是整个解析系统的核心层，实现了表格还原的核心算法。主要包括表格轮廓栅格化，复杂表头的识别与转换，表格内容归位等内容。顶层是对已处理的表格数据的进一步封装。



图5-2 基于PDF的表格数据抽取系统层次图

* + - * PDF文档解析层：主要借助PDF开源类库PDFBox完成基本的PDF文档解析。通过修改PDFBox底层代码完成对PDF字符流中文字流与线条流的识别，并将其封装成上层能够识别和使用的格式。
      * PDF表格还原层：主要的PDF表格还原算法都在该层实现。表格轮廓还原算法主要通过对线条位置信息的排序和归类完成。复杂表头的识别和转化通过将表头转化成树形结构完成。转化完成后整个表格被整理成二维数组的形式，再通过对比表格中的文字信息与表格的位置将文字归位。最终完成从文字线条信息到二维数组的转换。
      * PDF表格数据封装层：有了中间层的数据处理，在数据封装层可以将二维数组形式的表格数据封装成各种需要的形式。本系统目前支持HTML、JSON格式的封装。HTML形式的表格能够很好的支持网页浏览，而JSON格式则利于信息的交换和存储。

## 基于PDFBox的文档解析层

PDF文档解析层主要借助PDF的开源类库PDFBox完成对PDF中文字与框线信息的提取和封装。PDFBox作为一种优秀的PDF类库，在底层对PDF的文字信息进行了较好的封装。单个的文字流包含了文字的字体，字号，颜色，位置坐标等信息。对于文字信息，本系统在PDFBox封装的基础上进行了简化处理。而对于框线信息，由于PDF中对于表格的框线没有明确的定义，表格框线和普通的线条并无区分。因此本系统根据线条的特征，参照文字的封装格式对线条进行了封装，为中间层的表格还原做好准备。在第四章已经着重介绍了基于PDFBox提取基本表格特征的实现细节，下面给出整个文档解析层的总体实现框图。



图5-3 解析层总体实现框图

解析层最终目的是为上层的表格还原层提供接口，使其能按页获取单页PDF的文字和线条信息。整个流程横跨PDFBox的三层，主要的修改集中在Cos Model层，通过解析PDFOperator对象的Q，q，re标记完成线条信息的提取。通过解析PDFOperator的BT，ET标记完成文字信息的提取，并将文字和线条信息分别封装于LineInformation类和CharacterInformation类中。

## PDF表格还原层

表格还原层是整个表格提取的核心层，通过处理下层封装的基本文字和线条信息还原出实际的表格，并最终通过二维数组来表示所有的表格信息。以便于上层对表格数据进行进一步的封装和使用。主要算法包括表格轮廓栅格化，表格内容归位，分页表格合并，复杂表头处理，表格展开方式识别等内容。在第四章已经对各个算法进行了详细的介绍，下面给出整个表格还原层的总体实现框图。



图5-4 表格还原层总体实现框图

该层利用解析层的线条对象和文字对象，首先利用线条获取二维表格的行数和列数，同时利用行列刻度将对应的文字填充入单元格。为了覆盖PDF中的大部分表格，针对大表做了表格合并处理，针对复杂表头做了表格简化，同时基于表格内容和行列特征对表格的方向进行了判别，上层的表格进行做好铺垫。上层的表格解析则利用该层提供的二维字符串型数组完成。

## PDF表格数据封装层

本系统提供两种表格的封装形式，HTML和JSON格式。HTML能很好的展示表格的结构，便于浏览。而JSON格式的数据这更利于存储和传输，该格式已经被广泛应用于非结构化的数据存储中。本系统中解析的表格数据就最终存放在非关系型的mongodb数据库中。

### 表格数据的HTML格式封装

HTML是一种标记语言，该格式的数据便于在浏览器中浏览，具有较强的可读性。而HTML对于表格数据又具有较好的支持，事实上HTML页面中很多的数据都是以表格的形式存储的。基本的表格标记包括<table>、<th>、<tr>、<td>等。<table>标记用于描述一个表格，该标记中可以使用border等属性表示表格的框线等信息。<th>描述表格的标题，<tr>表示表格的行，<td>则表示表格的列。对于一个二维数组形式的表格form,通过HTML表示后的样子如下：



图5-5 基于HTML封装的表格格式

下图是封装后的表格示例：

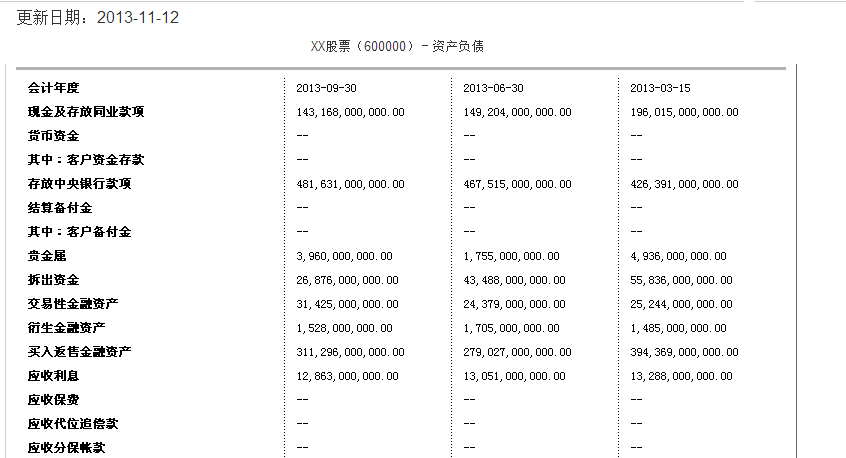


图5-6 基于HTML封装的表格示意

### 表格数据的JSON格式封装

JSON格式的表格数据展示。



图5-7-1 三张表的Json格式



图5-7-2 单张表的Json格式

单张表Json也是三张表中报表内容里每张表的具体格式。对于单张表的Json，内容的value值就是表格框线内的二维数组的内容。表格顺序是第一行各列的内容，由于Json的Object格式是没有顺序的，为了表示表格的基本顺序，使用了JsonArray格式记录表格的列顺序。下面的每行都是数组中的一个元素，key记录列值，value就是该列上第i行的内容了。通过该Json可以还原出原始的表格信息。

## 测试和分析

### 表格识别技术的评估参数

在信息抽取中，常见的性能评估方法包含召回率和准确率。其中召回率等于系统正确解析的结果占总样本的比例。具体计算公式如下：

而准确率则表示系统正确解析的结果占总的解析结果的比例，具体计算公式如下：

通常情况下召回率和准确率是个矛盾体，增加召回率可能对准确率会有一定的影响；反之，增加准确率同样对召回率会产生影响。因此为了综合反映一个抽取系统的性能，往往还会考虑召回率和准确率的加权几何平均值，计算公式如下：

其中，β是自定义阈值，用来控制召回率和准确率的权重。β等于1表示两者权重相同；β大于1则表示准确率更重要，小于1则召回率更重要。通常β值可以选取1,2,1/2。

本系统所做的实验的评估参数则主要参考上述的三个标准。在具体应用场景下，召回率R表示2000家上市公司财务报表中能用本系统解析方法处理的表格所占有的比例。准确率P表示所有能用本系统解析的表格中正确解析的表格所占的比例。F则是根据R值和P值综合算出的结果。

### 实验结果与分析

1. 沪深上市公司财报中财务报表的查全率分析

本次测试针对沪深两市的2000余家上市公司2012至2013年度的19938份财报进行测试分析。依照金融领域的相关规定，财报中必然包含财务报表，因此应该提取的表格数量与财报数量是一致的。进而可以通过统计解析成功的情况来计算相应的查全率。下表记录了各个季度财报对应的查全率。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 一季度财报 | 半年财报 | 三季度财报 | 全年财报 | 总计 |
| 财报数量 | 4937 | 4986 | 5040 | 4975 | 19938 |
| 解析数量 | 3028 | 3087 | 4468 | 3255 | 13838 |
| 查全率 | 61.33% | 61.91% | 88.65% | 65.42% | 69.40% |

表5-1 2012至2013年度财务报表查全率统计情况

各季度查全率柱状分布图如下：

图5-8 各季度财报解析查全率柱状分布示意图

查全率如图中折现所示，又上表知平均查全率为69.40%。各个季度的查全率分布较为平均。由于各公司财报形式各异，且相同公司不同季度财报的内容也有较大的差异。例如半年度和全年度财报往往需要与往年的数值进行对比，因此表格的形态也更加复杂，涉及复杂表格的处理。从查全率可以看出本系统对各类表格的处理具有良好的效果。对各个季度不同公司的不同相关表格具有较强的普适性。

对于未能解析的表格进行统计分析，结果如下图所示

图5-9 未解析表格统计结果示意图

由图可知，未解析的表格中65%属于无框线的表格。由于本系统需要根据表格框线信息来栅格化表格，因此对于无框线的表格是无能为力的。这也是本方法在未来的可改进的主要方向。对于图片类的表格，不是本课题的研究方向；不规范表格即除首行和首列外其他行列存在跨行跨列的表格，该类表格通常具有可视性但不具有结构性，也不是本课题的研究范围。

1. 沪深上市公司财报中财务报表的查准率分析

针对上述已解析的沪深两市的2000余家上市公司2012至2013年度的13838份财报进行测试分析。依照金融领域的相关规定，财务报表中必然包含某些特定的属性行，例如资产负债表中必然包含“流动资产”、“非流动资产”等项目；利润表中必然包含“营业收入”、“营业利润”等项目。因此可以通过检测每张表提取数据的完整性来衡量表格的正确情况。下表记录了各个季度财报对应的查准率。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 一季度财报 | 半年财报 | 三季度财报 | 全年财报 | 总计 |
| 解析数量 | 3028 | 3087 | 4468 | 3255 | 13838 |
| 准确数量 | 2663 | 2598 | 3971 | 2803 | 12035 |
| 准确率 | 87.94% | 84.16% | 88.87% | 86.11% | 86.97% |

表5-2 2012至2013年度财务报表查准率统计情况

各季度查全率柱状分布图如下：

图5-10 各季度财报解析查准率柱状分布示意图

查准率如图中折现所示，由上表知平均查准率为86.97%，且各个季度的查准率分布较为平均，均大于84%。可见本方法具有较高的准确率，原因在于采用了基于框线的表格还原算法，由于表格行列刻度的确定都是基于线条完成的，同时文字的填充也基于文字和线条的绝对位置关系，因此相对于其他的表格解析方式（如基于文字流的栅格化方式）具有更高的准确率。

对未能准确解析的表格进行分析发现，基本是表格跨页造成的。由于财务报表偏大，不能在一张PDF页面上显示，因此在PDF中的财报往往横跨了多页PDF，因此容易遗漏某些信息。该问题也是PDF中表格解析的难点问题。对于该部分的说明已在第四章的第二小节做了介绍。通过综合考虑列数，表格在页面中的相对位置等因素综合处理表格跨页问题，经过优化，准确率还是比较理想的。

1. 沪深上市公司财报中财务报表的F值分析

根据公式计算F指数，取β值为1，即查全率与查准率同等重要。计算出的各个季度的F值如下表所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 一季度财报 | 半年财报 | 三季度财报 | 全年财报 | 总计 |
| 查全率 | 61.33% | 61.91% | 88.65% | 65.42% | 69.40% |
| 查准率 | 87.94% | 84.16% | 88.87% | 86.11% | 86.97% |
| F值 | 72.48% | 71.23% | 88.76% | 74.17% | 77.56% |

表5-3 2012至2013年度财务报表F值统计情况

可见在查全率与查准率同等重要的情况下，本方法的F值平均为77.56%。而事实上本系统优先考虑的是查准率。由于采用了基于框线的表格还原算法，因此准确率得到了较好的效果。同时也舍弃了部分查全率。因此本系统更加适合比较正规的表格解析，例如财务报表等统计信息，由于相对规范，通常都是带有线条的，因此本方法能较好的完成该类表格的解析工作。

1. 财务报表解析成果展示



图5-11 武钢股份2013年一季度合并现金流量表部分表格PDF截图

图中为武汉钢铁股份有限公司2013年一季度合并现金流量表的PDF部分截图。由于财务三大报表的属性较多，通常一表跨页，因此无法截取整体图。该表格格式较为简单，是财报最常见的格式。下面分别给出解析后的HTML和JSON格式展示图。



图5-12 封装后的HTML表格部分截图



图5-13 封装后的JSON数据部分截图

### 表格结构识别测试与分析

由于财务报表类PDF中的表格基本都是横向展开的，因此很难测试表格结构识别的效果。为了更好的测试表格结构的识别效果，人工收集了公司财报、基金及普通类别的三类展开方式的表格各300个，分别采用基于单元格内容的识别、基于行列相似度的识别以及两者综合识别三种情况，得到的测试结果如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 表格展开类型 | 表格总数量 | 正确识别数量 | 正确率 |
| 财务类横向展开 | 100 | 95 | 95% |
| 财务类纵向展开 | 100 | 90 | 90% |
| 财务类综合展开 | 100 | 91 | 91% |
| 基金类横向展开 | 100 | 95 | 95% |
| 基金类纵向展开 | 100 | 99 | 99% |
| 基金类综合展开 | 100 | 89 | 89% |
| 普通横向展开 | 100 | 11 | 11% |
| 普通纵向展开 | 100 | 10 | 10% |
| 普通综合展开 | 100 | 20 | 20% |

表5-4 单独基于单元格内容的识别测试结果

|  |  |  |  |
| --- | --- | --- | --- |
| 表格展开类型 | 表格总数量 | 正确识别数量 | 正确率 |
| 财务类横向展开 | 100 | 70 | 70% |
| 财务类纵向展开 | 100 | 76 | 76% |
| 财务类综合展开 | 100 | 60 | 60% |
| 基金类横向展开 | 100 | 79 | 79% |
| 基金类纵向展开 | 100 | 55 | 55% |
| 基金类综合展开 | 100 | 65 | 65% |
| 普通横向展开 | 100 | 66 | 66% |
| 普通纵向展开 | 100 | 72 | 72% |
| 普通综合展开 | 100 | 60 | 60% |

表5-5 单独基于行列相似度的识别结果

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 表格展开类型 | 表格总数量 | 混合比例  （金融/普通） | 比例(内容/行列相似度) | 正确数量 | 正确率 |
| 混合横向展开 | 100 | 1:1 | 1:1 | 43 | 43% |
| 混合纵向展开 | 100 | 1:1 | 1:1 | 54 | 54% |
| 混合综合展开 | 100 | 1:1 | 1:1 | 60 | 60% |
| 混合横向展开 | 100 | 1:9 | 1:1 | 48 | 48% |
| 混合纵向展开 | 100 | 1:9 | 1:1 | 50 | 50% |
| 混合综合展开 | 100 | 1:9 | 1:1 | 34 | 34% |
| 混合横向展开 | 100 | 1:9 | 1:9 | 60 | 60% |
| 混合纵向展开 | 100 | 1:9 | 1:9 | 68 | 68% |
| 混合综合展开 | 100 | 1:9 | 1:9 | 70 | 70% |
| 混合横向展开 | 100 | 9:1 | 1:1 | 64 | 64% |
| 混合纵向展开 | 100 | 9:1 | 1:1 | 66 | 66% |
| 混合综合展开 | 100 | 9:1 | 1:1 | 72 | 72% |
| 混合横向展开 | 100 | 9:1 | 9:1 | 88 | 88% |
| 混合纵向展开 | 100 | 9:1 | 9:1 | 93 | 93% |
| 混合综合展开 | 100 | 9:1 | 9:1 | 96 | 96% |

表5-6 结合两者综合识别的结果

比较以上三组数据，不难发现：

1. 针对具体的类别，基于单元格内容的识别方式具有非常高的准确率。表5-4中，财务类和基金类的展开方式均有良好的测试效果。
2. 针对普通类别，基于单元格内容的识别方式几乎没有效果。表5-4中普通类表格的识别效果低于1/3，即低于随机猜测概率。
3. 基于行列相似度的识别方式对于不同类别表格的准确率基本一致。表5-5中，各类表格的识别率均在60%上。
4. 表5-6综合测试了不同比例下各类型表格的识别情况。当表格类型偏向某一特定类型时（如图中的金融类表格），加大内容识别比例能有效的提高表格特征识别率。而当表格类型不明确时（普通类型表格占比大），则采用行列相似度方式进行表格类别识别即可。
5. 综上，当表格类型已知时采用基于内容的表格结构识别方法具有明显的优势。当表格类型具有一定偏向时，采用内容与行列相似度结合的方式识别效果较好，同时可根据具体情况适当调整打分比例，即公式中的β值。当表格类型无明显倾向时采用单纯的行列相似度方式具有较好的识别效果。

## 本章小结

本章首先给出了表格解析系统的整体流程图，将系统分成三层分别进行描述。接着分层对各个部分进行详细的介绍。在文档解析层着重介绍了基于PDFBox的文字和线条提取方法，并展示了文字和线条封装后的结构。在表格还原层，针对三个核心的技术进行详细的介绍，包括表格轮廓栅格化，复杂表头识别和转换以及表格内容还原。经过该层，表格已被处理成便于表示的二维数组结构。最后介绍了表格数据的封装格式，包括HTML和JSON两种形式。并展示了两种格式的封装效果图。最后利用本系统对2000余家上市公司的财务报表进行解析分析，从查全率，查准率，F值三个方面综合测试了本系统的性能。实验表明本系统采用算法对于全框线的表格具有较高的查准率。最后从内容和行列相似度两方面对表格结构识别进行了测试和总结。

# 总结与展望

## 论文总结

随着互联网行业的不断发展，电子文档优势日益明显，针对各类文档格式所进行的数据采集和数据挖掘工作逐渐成为近年来的研究热点。本文提出了一种基于PDF的表格提取方法，并通过将该方法应用于上市公司财务报表数据的提取证明了本方法的有效性和实用性。由于表格解析是基于表格框线的绝对位置信息完成的，因此在处理复杂表格以及表格识别准确率上较其他方法具有更高的准确性。下面归纳总结本课题的几个技术难点：

1. 由于PDF特殊的文档结构，对PDF进行表格解析具有一定的难度。市场上针对PDF的表格提取工具并不多，且都非开源。本课题独创性的修改了PDFBox的源码完成了针对PDF格式文档的表格特征提取。从表格的线条和文字位置信息入手设计相关算法完成表格的识别和提取，经测试解析效果良好。
2. 本方法的表格解析是基于表格框线的绝对位置信息完成的，因此在处理复杂表格和解析正确率上相对单纯基于文字流的识别算法都有了明显的改进。对于复杂表头处理，由于框线位置信息已知，能够较好的识别首行跨列、首列跨行等特殊情况。同时实现了复杂表头的简化并最终实现复杂表头和简单表头的归一处理。在解析正确率方面，由于所有的线条和文字信息都是已知的，因此能有效的避免文字错分，文字击穿等问题，因此识别准确率也相应增加。
3. 对于表格结构的识别，本课题提出了两种处理思路。综合考虑了表格的物理特征和表格的内容特征，经测试具有良好的识别效果。表格格式识别的多样性也确保了本方法表格识别的通用性，多样化了本方法的应用场景。
4. 本课题以解析财务报表为例展示了方法的实用性，通过测试，查全率能够达到70%，查准率可达到87%。已经可以较好了实现自动化的表格提取。事实上各类金融终端产品都涉及财务F10数据的采集，采用本方法能有效的减少人工介入，提高效率，具有较强的实际应用价值。

综上所述，本课题所研究的基于PDF的表格数据抽取具有较高的准确率和较好的通用性。同时通过对2000加公司2012到2013年的财务报表进行解析测试验证了本方法的有效性和实用性。随着PDF文档应用场景的不断丰富，相信本方法在表格解析方面将会有更多的实际应用价值。

## 工作展望

由于本方法依托表格框线进行表格解析，虽然在处理复杂表格以及解析查准率方面具有优势，但查全率方面仍然有待完善，具体总结如下：

1. 对于无框线表格的处理有待解决：无框线表格即肉眼可分辨行列但没有框线划分的表格。此类表格的处理本方法没有涉及，因此总体查全率不高。后续可以通过分析文字流位置信息对文字进行划分，栅格化处理等完成表格的还原。
2. 跨页表格的合并：本课题针对跨页表格的合并问题已经提出了一种解决方案。但是由于现有方案只考虑到行列数目以及绝对位置信息等物理信息，并没有从表格内容的方面进行考虑。因此在测试中发现依然存在表格信息不完整的问题。后续改进可以结合表格内容进行，针对不同类型的表格建立词典库，以避免表格解析信息不完整的问题。
3. 表格结构的识别：对于类型已知的表格（如金融类等），采用基于内容的属性识别具有较高的准确度。但对于普通表格，单纯利用行列单元格内的文字数目特征判定准确度依然不高。可考虑加入其他与类别无关的内容特征。如单元格中的数值占比，文字粗细，颜色等因素。

# 参考文献

<https://www.google.com/,google>搜索引擎

<http://open.weibo.com/>,新浪微博开放平台

<http://dev.t.qq.com/>,腾讯微博开放平台

<http://open.t.sohu.com/>,搜狐微博开放平台

PDF Reference sixth edition, Adobe PDF第六版官方文档

<https://pdfbox.apache.org/,PDFBox>开源项目

Liu Y, Wu G Q, Hu X G. A web table extraction algorithm based on tree edit distance[C], Conference Anthology, IEEE. IEEE, 2013: 1-6.

Cha S, Ma Z, Cheng J, et al. Learning of ontology from the web-table[C], Fuzzy Systems and Knowledge Discovery (FSKD), 2011 Eighth International Conference on. IEEE, 2011, 3: 1454-1458.

赵洪,肖洪,薛德军,等. Web表格信息抽取研究综述[J].现代图书情报技术, 2008, 3: 24-31.

林科锵.Web页中表格结构识别的研究与实现[J].成都:电子科技大学, 2006.

袁鸿雁.基于本体的Web表格信息抽取技术的研究[J].青岛大学学报:自然科学版, 2010, 23(2): 47-51.

B. Yildiz, K. Kaiser, and S. Miksch. pdf2table: A method to extract table information from pdf files. In IICAI, pages 1773–1785, 2005.

Hassan T, Baumgartner R. Table recognition and understanding from pdf files[C], Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on. IEEE, 2007, 2: 1143-1147.

Liu Y, Bai K, Mitra P, et al. Improving the table boundary detection in pdfs by fixing the sequence error of the sparse lines[C], Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on. IEEE, 2009: 1006-1010.

Deivalakshmi S, Chaitanya K, Palanisamy P. Detection of table structure and content extraction from scanned documents[C], Communications and Signal Processing (ICCSP), 2014 International Conference on. IEEE, 2014: 270-274.

Kasar T, Barlas P, Adam S, et al. Learning to Detect Tables in Scanned Document Images using Line Information[C], Document Analysis and Recognition (ICDAR), 2013 12th International Conference on. IEEE, 2013: 1185-1189.

李效东,顾毓清.基于DOM 的Web信息提取[J].计算机学报, 2002, 25(5): 526-533.

Hurst M. Classifying table elements in html[C], 11th International Conference on World Wide Web (WWW 2002), Poster Paper. 2002.

Y. Wang and J. Hu, “A Machine Learning Based Approach for Table Detection on the Web,” Proc. 11st Int’l World Wide Web Conf. (WWW 2002), pp. 7-11, 2002.

S. W. Jung, M. Y. Kang, H. C. Kwon, “Hybrid Approach to Extracting Information from Web-Tables,” LNAI 4285, pp. 109-119, 2006.

S. W. Jung, M.Y. Kang, H.C. Kwon, “Constructing Domain Ontology Using Structural and Semantic Characteristics of Web-Table Head,” LNAI 4570, pp. 665-674, 2007.

于伯峰. 印刷体中文文档中表格和汉字的识别研究[D]. 哈尔滨工程大学, 2011.

李海涛,张小惠.一种统计特征点网格分布的表格图像识别方法[J].华中科技大学学报: 自然科学版, 2002, 30(9): 60-63.

蒋东玉,田英鑫.图像中表格转换成 HTML 表格的研究与实现[J].黑龙江科技信息, 2014, 24: 184.

邵全. 基于 Hough 变换的表格倾角检测[J]. 科技信息 (学术研究), 2007, 16.

Zheng Y., Liu C. Ding x., Pan S.,"Form Frame Line Detection with Directional Single-Connected Chain", Proc. of the 6th Int. Conf.on Doc. Anal. & Recognition, pp. 699-703, 2001.

Thomas G. Kieninger, "Table Structure Recognition Based on Robust Block Segmentation", German research center for artificial Intelligence, 1998.

刘长松,潘世言.一种表格框线检测和字线分离算法[J].电子与信息学报, 2002, 24(9): 1190-1196.

郑冶枫, 刘长松,丁晓青,等.基于有向单连通链的表格框线检测算法[J].软件学报, 2002, 13(4): 790-796.

张伯.基于PDF文字流的表格识别技术的研究[D].北京工业大学, 2010.

Oro E, Ruffolo M. Pdf-trex: An approach for recognizing and extracting tables from pdf documents[C], Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on. IEEE, 2009: 906-910.

Yoshida M, Torisawa K, Tsujii J. A method to integrate tables of the world wide web[C], Proceedings of the International Workshop on Web Document Analysis (WDA 2001). 2001: 31-34.

Gu Y, Ji L, Jiang Z, et al. Endless and scalable knowledge table extraction from semi-structured websites[C], Data Mining Workshops (ICDMW), 2012 IEEE 12th International Conference on. IEEE, 2012: 835-842.

Ramel J Y, Crucianu M, Vincent N, et al. Detection, extraction and representation of tables[C], Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on. IEEE, 2003: 374-378.

Wang Y, Phillips I T, Haralick R. Automatic table ground truth generation and a background-analysis-based table structure extraction method[C], Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on. IEEE, 2001: 528-532.

Yuan F, Lu B. A new method of information extraction from PDF files[C], Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on. IEEE, 2005, 3: 1738-1742.

潘小燕, 孙承杰, 刘远超, 等. 半结构化文本中的表格识别技术研究[J]. 微计算机信息, 2008, 24(18): 198-199.

# 附录

|  |  |  |
| --- | --- | --- |
| 缩略语 | 全称 | 描述 |
| PDF | Portable Document Format | Adobe公司发布的跨平台文件格式 |
| HTML | Hyper Text Markup Language | 超文本标记语言 |
| JSON | JavaScript Object Notation | 一种轻量级数据交换格式 |
| DOM | Document Object Model | 文档对象模型 |

# 致 谢

光阴荏苒，研究生的两年半学习时光一转眼就过去了。在这两年半里，我不仅学到了很多计算机专业相关的基础知识，同时在实验室项目中锻炼了自己的动手能力。编程能力也得到了很大的提高。我本科并不是计算机专业的学生，转专业内心也有过挣扎，怕不能很好的适应计算机相关专业的学习和科研工作。不过到了今天，经过我自己的不断努力以及我的导师，师兄师姐，朋友的帮助，我真的觉得自己有了很大的进步。在这里对曾经给予我指导、帮助、鼓励的老师和同学们表示衷心的感谢。

首先，我要特别感谢我的导师闫丹凤闫老师。闫老师具有丰富的科研和教学经验，无论是在学术研究上，项目管理推进上，教书育人上都是我们学习的榜样。老师严谨治学的态度给我留下了深度的印象，我们提交的文档，大到学术论文，小到一次例会的周报，闫老师都会认真的阅读。同时给我们提出宝贵的意见，从内容到格式给予全方位的指导，甚至连小小的标点符号都不会放过，老师这种对学术一丝不苟态度深深的影响了我们每一个人，我们会将这种严谨扎实的工作态度带到工作中去，感染身边的每个人。

其次，我还要感谢项目组的师兄师姐、师弟师妹们，感谢田瑞师兄、张丽莹师姐、黄俊霖师兄、邹文涛师弟、徐佳师妹、华恩正师弟等等。项目组就像一个大家庭，大家互帮互助，共同学习进步。正是由于大家的团结踏实，项目才能顺利的推进。也正是由于他们的帮助和协作，我才能顺利完成我的硕士论文。

再次，我要感谢一直在我身后支持我的家人，朋友。一路走来，从小学到大学再到研究生，是他们在背后默默的鼓励和支持才使我变得越来越好。

最后，感谢我的母校北京邮电大学。七年的大学生活即将结束，真的很舍不得我的母校。虽然她不大，但是处处都是温暖。即将走入社会，我不会忘记母校对我的谆谆教诲，厚德博学、敬业乐群的“北邮精神”将长留我心中。

唐皓瑾

2014年12月30日

# 攻读学位期间发表的学术论文目录

1. TANG H, YAN D, TIAN Y. Semantic dictionary based method for short text classification[J]. The Journal of China Universities of Posts and Telecommunications, 2013, 20: 15-19.