文本复制检测报告单(全文标明引文)

№:ADBD2018R_20180530153127201805301548274401740454 68　　检测时间:2018-05-30 15:48:27

检测文献:　53140228_苑雨萌_计算机科学与技术_金融表格信息抽取及结构化存储应用

作者:　苑雨萌

检测范围:　中国学术期刊网络出版总库

中国博士学位论文全文数据库/中国优秀硕士学位论文全文数据库

中国重要会议论文全文数据库

中国重要报纸全文数据库

中国专利全文数据库

图书资源

优先出版文献库

大学生论文联合比对库

互联网资源(包含贴吧等论坛资源)

英文数据库(涵盖期刊、博硕、会议的英文数据以及德国Springer、英国Taylor&Francis 期刊数据库等)

港澳台学术文献库

互联网文档资源

CNKI大成编客-原创作品库

个人比对库

时间范围:　1900-01-01至2018-05-30

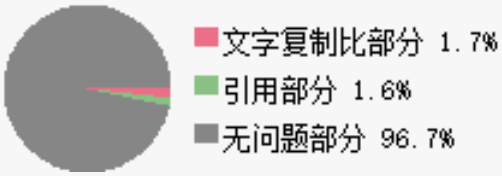
检测结果

总文字复制比：3.3%　　跨语言检测结果：0%

去除引用文献复制比：1.7%　　去除本人已发表文献复制比：3.3%

单篇最大文字复制比：1.6%（Web页中表格结构识别的研究与实现）

重复字数：	[805]	总段落数：	[6]
总字数：	[24214]	疑似段落数：	[5]
单篇最大重复字数：	[394]	前部重合字数：	[30]
疑似段落最大重合字数：	[297]	后部重合字数：	[775]
疑似段落最小重合字数：	[30]		



指　标：☐疑似剽窃观点　☑疑似剽窃文字表述　☐疑似自我剽窃　☐疑似整体剽窃　☐过度引用

表格：0　　公式：0　　疑似文字的图片：0　　脚注与尾注：0

0.7%（30）　中英文摘要等（总4581字）

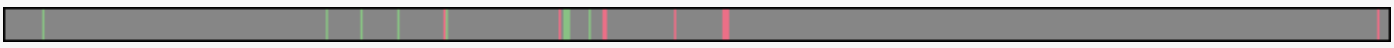
5.9%（214）　第一章绪论（总3638字）

9.4%（297）　第二章相关理论知识（总3156字）

2.4%（210）　第三章系统设计与实现（总8627字）

0%（0）　第四章实验结果及分析（总3692字）

10.4%（54）　第五章结论（总520字）



（注释：无问题部分　文字复制比部分　引用部分）

1. 中英文摘要等　　总字数：4581

相似文献列表　文字复制比：0.7%(30)　疑似剽窃观点：(0)

1	Web页中表格结构识别的研究与实现	0.7%（30）
	林科锵(导师：左志宏) -《电子科技大学硕士论文》- 2006-01-01	是否引证：是

原文内容　红色文字表示存在文字复制现象的内容; 绿色文字表示其中标明了引用的内容

摘要

金融表格信息抽取及结构化存储应用

随着互联网的高速发展，计算机已经渗透到我们日常生活的每一个角落。金融作为现代市场经济发展的核心，与计算机的结合越发紧密，信息技术与金融领域的结合，促进了金融领域的多元创新与发展，金融行业的信息资料也得以更完整地保存。表格是文档中呈现信息的一种有效方式，它以简单的结构展现出丰富的信息，是我们重要的知识来源，如何从表格中获取我们所需的信息是一个值得研究的课题。

表格抽取是解决这个问题的一种常见方法，也是本文研究的主要内容。本文主要研究金融类表格的信息抽取和结构化存储，对于一个待抽取的金融类表格，利用表格抽取的知识对表格进行分析，生成相应的属性-值对，存入结构化的数据库表内，从而完成对表格数据的获取。

本文调研了国内外现有的表格抽取方面的算法，根据表格的类型表格抽取可以分为文本类表格抽取和web表格抽取，其中有关web表格的抽取是当今研究的主流。对于文本类表格，我们可以利用知识工程技术（Knowledge Engineering）和机器学习技术（Machine Learning）对表格进行抽取。对于web表格，常见的抽取算法有：基于Wrapper学习的方法，通过归纳学习方法生成抽取规则，自动化或半自动化地构造抽取器；基于表格结构分析的方法，通过对表格结构的解析，将表格转化为相应的逻辑结构，从而抽取表格中的数据；基于本体的方法，根据给定领域本体中表格结构和内容的定义构建抽取规则。

通过对上述算法的学习和分析，结合本文研究内容，我们选取了基于例子学习的方法对表格进行***抽取。依照表格抽取的流程和算法，本文自主开发了一个金融类表格***自动抽取及结构化存储系统。系统主要包括两大部分：一是对tabula截取的表格进行分析，将表格中拆分错误的单元格合并，使每个单元格的信息存储完整，以确保最终抽取数据的准确与完整。二是利用表格抽取的相关知识和技术，通过对表格内容的匹配，判断表格结构，将表格中的每一条数据取出存入数据库表内。系统读取文件并将文件正确存储到数据库表内的正确率为73.68%。

目前系统处理的表格大多结构简单，今后将针对嵌套表格、分页表格等复杂类型的表格进行表格抽取，同时修正系统目前存在的问题，以提高系统的准确率和鲁棒性。此外，有必要制定一种更为标准的系统自动评判标准，以代替人工检查。

关键字：表格抽取，金融类表格，单元格合并，属性学习

Abstract

Financial table information extraction and structured storage applications

With the rapid development of the Internet, computers have penetrated into every corner of our daily life.As the core of modern market economy development, finance has become more closely integrated with computers.The combination of information technology and finance has promoted multiple innovations and developments in the financial sector, and information in the financial industry has also been more fully preserved.Forms are an effective way to present information in documents.It shows a wealth of information in a simple structure and is an important source of knowledge for us.How to get the information we need from the table is a topic worthy of study.

Form extraction is a common method to solve this problem and it is also the main content of this study.This paper focuses on information extraction and structured storage of financial forms.For a financial table to be extracted, the table is analyzed using the knowledge extracted from the table, and corresponding attribute-value pairs are generated and stored in a structured database table so as to complete the acquisition of the table data.

This paper investigates the existing algorithms for table extraction at home and abroad.According to the type of the form, the form extraction can be divided into a text type form extraction and a web form extraction. The extraction of the web form is the mainstream of the current research.For text-like tables, we can use Knowledge Engineering and Machine Learning to extract the tables.For web forms, the common extraction algorithms are:Based on the Wrapper learning method, extraction rules are generated by summarizing learning methods, and extractors are constructed automatically or semi-automatically;Based on the table structure analysis method, through the analysis of the table structure, the table is transformed into a corresponding logical structure, so as to extract the data in the table;Ontology-based methods build extraction rules based on the definitions of table structures and content in a given domain ontology.

Through learning and analysis of the above algorithm, combined with the content of this study, we have selected the method based on example learning to extract the table.According to the table extraction process and algorithm, this article independently developed a financial table automatic extraction and structured storage system.The system mainly includes two parts:One is to analyze the table Intercepted by tabula.Combine the wrongly split cells in the table to make each cell's information stored intact to ensure that the final extracted data is accurate and complete.The second is to use the relevant knowledge and techniques extracted from the table to determine the structure of the table by matching the contents of the table. Each piece of data in the table is retrieved and stored in the database table.The correct rate for the system to read the file and correctly store the file in the database table is 73.68%.

Most of the tables currently processed by the system are simple in structure.In the future, table extraction will be performed for complex types of tables such as nested tables and paging tables, and the existing problems in the system will be corrected to improve the accuracy and robustness of the system.In addition, it is necessary to develop a more standard system automatic evaluation criteria instead of manual inspection.

Keywords: table extraction, financial forms, cell merging, attribute learning

目录

2. 第一章绪论

总字数：3638

相似文献列表　文字复制比：5.9%(214)　疑似剽窃观点：(0)

1	Web页中表格结构识别的研究与实现 林科锵(导师：左志宏) -《电子科技大学硕士论文》- 2006-01-01	5.3%（193） 是否引证：是
2	XML模式匹配方法研究 金贤哲(导师：李瑞轩) -《华中科技大学硕士论文》- 2008-06-01	1.4%（52） 是否引证：否

原文内容　红色文字表示存在文字复制现象的内容; 绿色文字表示其中标明了引用的内容

第一章绪论

1.1 研究背景及意义

在这个经济高速发展的时代，金融行业占有重要的地位。作为国民经济的重要组成部分之一，金融已经成为了现代市场经济发展的核心。随着互联网的发展，信息技术广泛应用于各个行业，为金融行业的发展带来了巨大的革新：信息技术和金融行业的结合，使得网络成为了金融行业中一个新的交互端口，促进了金融机构的多元化创新与发展，为金融行业提供了一个新的管理模式，可以全面、方便、快捷地管理金融行业的信息，[1]同时也使金融信息资源更加丰富、完善，方便相关人士的获取。

互联网的发展为电子信息的传播和获取带来了极大的便利。然而，互联网中信息的爆炸性增长，也为我们正确获取信息带来了极大的不便，也就是我们所说的“信息爆炸，资源匮乏”。因此，如何准确快捷地获取我们所需的信息，成为了近年来研究的一个重点。信息抽取技术应运而生。

表格抽取是信息抽取中的一个重要分支。表格作为文献中的常见部分，结构简单，方便易读，是我们在书写文章时常用的表现形式，表格以直接、紧凑的形式表现数据，里面蕴含了丰富的信息资源，因此表格数据的抽取是一个极其有价值的研究课题。然而表格的设计是基于人眼阅读的，人可以根据通过阅读表格，轻松地理解表格含义并得到需要的信息，计算机却不能，对于计算机而言，“定义表格的句法和语法概念是相互混合的，表格中的单元格的语义依赖于其所在位置信息（这里指句法），但这种句法结构比自然语言要复杂的多”。因此，如何令计算机准确地抽取表格中的数据，一直是一个具有挑战性的课题。[8]

金融与信息的紧密结合，使金融领域的电子信息数据越发完善。金融行业中的许多信息和公告等都会以PDF（便携文件格式）文件的形式发布在相关网站上。本文选取了金融类网站上发布的交易报告书方面的文档，对其中出现次数较多的、具有典型特征的表格，如记录财务数据、股权结构、资产主要权属、对外担保等信息的表格，进行数据抽取和结构化存储，以供后续分析使用。

1.2课题研究的现状

1.2.1信息抽取

信息抽取的研究致力于为人们提供有效的信息获取方法和工具，以解决信息爆炸引起的问题。不同的研究人员对它有不同的定义：Proteus工程的创建者Grishman认为“信息抽取涉及到为从文本中选取出的信息创建的一种结构化表示形式（例如：数据库）”，微软亚洲研究院2005年信息抽取技术暑期研讨班则将信息抽取定义为“信息抽取是抽取和链接基于用户详细说明的相关信息的过程”。[2]目前对于*信息抽取的普遍定义是：从各种类型的文本中定位、识别和提取出需要的信息点，表示成一种统一的、结构化的形式。[3]*

*信息抽取的*形式主要可以分为四种：文本特性抽取、文档群集和分类、遥感信息抽取与处理、以及生物特征识别。信息抽取通过对数据的抽取和过滤，得到用户关心的数据，并将数据存储到计算机中。[2]信息抽取处理的数据对象通常指自然语言文本，尤其是非结构化文本。但从广义的角度上来讲，图像、视频、语音等其他媒体类型信息也可以作为信息抽取处理的数据源。[3]

信息抽取起源于20世纪60年代中期，早期的研究重点在于从自然文本中获取结构化信息，其中，美国纽约大学进行的Linguistic String项目和耶鲁大学有关事故理解的研究是这一阶段的代表项目，为信息抽取的发展奠定了基础。从上世纪80年代末开始，信息抽取的研究进入了蓬勃发展阶段，这主要归功于Tip ster文本项目、消息理解会议（MUC）、自动内容抽取会议（ACE）、多语言实体任务会议（MET）等的开展，目前信息抽取的研究主要侧重于多语言文本的处理、篇章分析技术、深层理解技术等，应用领域也更加广泛。[4]

1.2.2表格抽取

表格抽取是信息抽取技术的一个重要分支，兴起于上个世纪九十年代。表格抽取的研究涉及到各种类型文档中表格的定位、结构识别、内容分析、抽取有价值的信息并存储等技术。[3]目前国内外对信息抽取的研究主要集中在文本抽取，对表格抽取技术的研究涉及不多。

表格抽取起源于上世纪90年代*中期，M.Hurst等人开启了对表格抽取技术的研究。早期表格抽取的工作主要围绕ASCII文件或由光学字符识别（OCR）得到的文件中的表格展开，*[3]研究的内容主要是对表格的识别和单元格分类，早期对表格的抽取方法通常有以下几种：1.知识工程技术（Knowledge Engineering）：利用表格的格式化线索生成启发式规则，进行表格抽取。2.机器学习技术（Machine Learning）：通过机器学习的相关技术，如决策树、支持向量机、隐马尔可夫模型、条件随机域等，结合表格中的一些特殊符号，生成相关启发式规则对表格的结构进行分析，得到相应的“属性-值”对。[5]

90年代末期，web表格抽取技术的研究兴起。web表格信息抽取是从web表格中抽取出语义一致的、结构化表示的数据和知识。目前对web表格进行抽取的方法大致可以分为3种：1.基于Wrapper学习的方法，通过归纳学习方法生成抽取规则，自动化或半自动化地构造抽取器。2.基于表格结构分析的方法，通过对表格结构的分析，生成一种合适的逻辑数据结构保存表格内容，通过对逻辑结构的处理来抽取表格中需要的数据信息。3.基于本体的方法，根据具体的领域本体中对表格结构和内容的定义产生抽取规则。

web表格信息抽取可以分为表格识别和内容抽取两个部分。表格识别指从网页中定位出表格区域并分析表格的结构，内容抽取则是为表格中的数据生成正确的“属性-值”对并结构化表示。web表格信息抽取的关键技术包括表格定位、结构识别、内容整合和数据抽取等。

有关web表格的研究，国内外也取得了许多研究成果。台湾学者Chen等人率先开始了对web*表格抽取技术的研究，研究内容涉及到表格的定位、结构识别以及*数据抽取等。BYU研究小组的Embley等人使用了基于本体的技术抽取web表格数据，他们将表格抽取的过程具体分成了表格理解、数据整合和信息抽取等部分。[6]Tengli等人的研究则采用了学习的方法，他们通过对样本表格中属性名称的学习和启发式规则来完成对表格的抽取。[7]Pivk等人将web表格的抽取又进一步细分为物理层、结构层、功能层和语义层，研究了表格的规整、定位、识别和分析等技术。Zhai等人研究的方法由两步组成：一是利用标记字符的编辑距离等信息识别网页上数据记录区域，二是利用基于树匹配的部分对齐技术从数据记录区域中对齐和抽取数据。Gatterbauer等人创建了VENTex方法，这种方法将表格的拓扑结构、样式等线索与CSS2 Visual Box Model构造相应的启发式

规则相结合，对表格进行抽取，做到了抽取方法与表格所属领域无关。吴扬扬等人基于语义和数据特征，提出了一种新的数据提取方法，方法包括web列表的识别和元组的抽取。林科锵、林琳基于BYU研究小组的成果，将表格抽取分解成表格定位、结构识别和内容抽取，由此建立了一个基于本体的通用web表格信息抽取系统模型。[8]

1.3本文研究的内容

本文的研究内容包括表格抽取中表格的结构识别、内容抽取以及数据存储部分。具体而言，对于一个给定的金融类表格，划分出其属性与数据区域，判断表格的展开方式和表格结构，从而得到所需的“属性-值”对，将其存入结构化的数据库表中。本文的研究主要围绕以下几方面展开：

学习表格数据抽取方面的相关知识，了解国内外有关表格抽取的研究现状，调研并学习现有的表格抽取方面的算法，对各种算法进行分析比较，选择合适的算法来实现本文中表格的抽取。

根据本文的研究内容设计一个表格抽取及结构化存储系统，系统要求实现对csv文件的处理、表格结构的分析以及数据的抽取与存储等。

1.4 本文的组织结构

本文由五章***构成，组织结构如下：***

第一章：本章简要介绍了本文的研究背景和意义，阐述了本文研究涉及的信息抽取与表格抽取的历史与现状，描述了课题的研究内容。

第二章：针对本文研究课题，对研究涉及的相关理论知识进行了介绍，包括表格的结构基础、表格抽取的综述、常见的抽取方法概述和其优缺点，以及本文处理的文件的语法结构。

第三章：具体描述了表格抽取及结构化存储系统的设计与实现，首先对系统的开发环境及部分工具进行了简单的介绍，概述了系统的主要功能和流程，接着对系统中每个模块的功能、设计思路、实现原理以及模块的重难点进行了详细讲解，主要模块有文件的预处理、单元格的合并、属性名称的学习、属性单元格的匹配、以及数据库存储。

第四章：对系统的各个模块分别进行了功能测试，同时对系统进行了整体测试，对测试的实验数据和实验结果进行统计记录，根据实验结果对各个模块进行分析，寻找错误与不足之处，思考可能的解决方法。

第五章：对全文工作的总结，总结目前工作的成果，指出了目前工作的不完善和待解决之处，以及后续工作方向。

3. 第二章相关理论知识

总字数：3156

相似文献列表　文字复制比：9.4%(297)　疑似剽窃观点：(0)

1	Web页中表格结构识别的研究与实现 林科锵(导师：左志宏) -《电子科技大学硕士论文》- 2006-01-01	5.4%（171） 是否引证：是
2	Web中的行情数据获取与预测研究 于春燕;胡学钢; -《计算机工程与应用》- 2009-07-11	2.8%（88） 是否引证：否
3	基于DBpedia的材料知识抽取系统设计与实现 李欣(导师：张晓明;程煜) -《河北科技大学博士论文》- 2015-12-01	1.0%（31） 是否引证：否

原文内容　红色文字表示存在文字复制现象的内容; 绿色文字表示其中标明了引用的内容

第二章相关理论知识

2.1表格基础

表格是多组数据之间逻辑关系的二维表示，由行和列交错组合而成，其中单元格是表格最基本的组成元素。表格与列表不同，列表只是一系列相似的数据或数据记录，数据之间不含有语义关系和层次结构，而表格之间的数据是相互关联的。

通常，一个表格由表格题目和主体内容组成，主体内容又可以分为四个区域：ULC、列表头、行表头和数据区（如图2-1）。ULC是表格左上角的单元格，一般来说，如果ULC的内容为空，则该表格是一个二维表格，ULC的内容非空时，则需要根据表格的语义信息进一步确定表格的维数和展开方式；所有的数据元素构成数据区；表头由所有的属性元素组成，根据表头的分布位置，表头又可以分为行表头和列表头，当表头是表格行的开头时，表头属于行表头，当表头是表格列的开头时，表头属于列表头。[5]

图2-1 表格结构

根据物理结构的不同，可以将表格分为文本表格、光学字符（OCR）表格、web表格、excel表格、PDF表格等，不同类型的表格，我们需要采用不同的方法将表格解析为逻辑表格来处理。文本表格是指在ASCII文件中存储的表格，文件以空格、制表符等特殊符号作为分隔符区分单元格，对于这类表格，我们可以利用知识工程或机器学习的方法进行数据抽取。web表格可以进一步分为标记表格和非标记表格，标记表格可以利用表格的相关标签来识别、抽取表格，而非标记表格的处理则相对复杂一些，表格中的各个部分并没有像标记表格一样直接用标签标记出来，需要通过编写算法识别出表格中的各个部分。光学字符（OCR）表格是指通过扫描得到的表格，这类表格主要通过图像处理的方法来处理表格。[5]

单元格从功能上可以分为属性单元格和数据单元格。属性单元格构成表头，定义了数据的属性，数据单元格则可根据属性单元格提供的语义信息，形成一组组互相关联的数据记录。根据属性单元格在表格中的位置，我们可以把表格的展开方式分

为三类：按行展开，按列展开和混合展开（如图2-2）。对于按列展开的表格，一般第一行或从顶行开始的几行为属性单元格，这些行组成属性区域，剩余行构成数据区域；按行展开与按列展开相似，只是它的属性区域分布在第一列或从最左面开始的几列；混合展开则是上述两种展开方式的结合，它既包含有行表头也有列表头。[5]

图2-2 表格展开方式

根据本文所需处理的表格的常见类型，我们只处理最基本的二维表格，不包含混合型的表格，也不包括列表。其中表格的展开方式只处理按行展开和按列展开类型，暂不处理混合类型。

2.2表格抽取概述

表格抽取是信息抽取的重要分支之一，研究内容包括从各种类型的文档中对表格进行定位、结构识别、内容分析、抽取有价值的信息并存储等。表格抽取的过程主要包括表格识别和表格内容抽取，表格识别是指从各种类型的文档中定位目标表格所在位置并分析表格的结构，表格内容抽取是指从表格中提取出相应的“属性-值”对并结构化表示结果。

根据表格抽取的过程，表格抽取的关键技术有表格定位、结构分析、内容整合、结果表示等。表格定位指从文档页面中找到表格所在区域，难点在于从“假表格”等噪音中区分出真正的表格，这里，“假表格”指那些在文档中用于布局而非真正展示数据的表格，可以利用基于机器学习分类、人工构造规则分类和本体辅助分类的方法区分；结构分析要求通过对表格结构的分析，生成表格对应的逻辑结构形式，具体要求实现对标题行和主体区域的识别、表格展开方式识别以及属性区域和数据区域的识别；内容整合是指在构造的逻辑结构的基础上识别并规整表格的内容，解决方案有基于规则的整合、基于决策树的整合和基于本体的整合；结果表示将整合后的内容以结构化的形式表示出来，整合后的结果一般呈“属性-值”对的形式，可以表示为相应的“属性-值”对二元组、XML和关系数据库等形式。[8]

2.3常见表格抽取方法

早期的表格信息抽取主要围绕ASCII文件或由光学字符识别（OCR）得到的文件中的表格展开，表格抽取的方法通常分为以下几种：1.知识工程技术（Knowledge Engineering）：利用表格的格式化线索生成启发式规则。2.机器学习技术（Machine Learning）：通过机器学习的相关技术，如决策树、支持向量机、隐马尔可夫模型、条件随机域等，结合表格中的一些特殊符号，制定相关启发式规则来识别表格的结构，得到相应的“属性-值”对。[5]

随着网络的快速发展，web表格抽取的研究兴起并逐渐成为了当今表格抽取研究的主流，关于web表格抽取的方法，主要有以下几种：

1.利用web页面中与表格有关的HTML标签构造数据抽取器。根据页面中描述表格的HTML标签，构造合适的抽取规则来抽取我们需要的信息，然后将数据按一定格式存储即可。这种方法的抽取效果良好，但缺点也是显而易见的：抽取器依赖于网页的结构，然而现实中网络页面不是一成不变的，一旦页面的格式发生改变，抽取器就必须要重新编写，因此这种方法太过费时费力。

2.通过学习的方法构造抽取器，这种方法原理上与上一种方法相同，但它利用了自动化、半自动化的手段来构造抽取器，如基于例子的学习，这样可以减轻构造抽取器时的繁重工作。

3.基于表格的结构分析，将web表格转化为相应的逻辑结构对表格进行抽取。这种方法包括基于树结构和基于视觉线索两种抽取方式。基于树的抽取模式将网页转化为树状结构，根据特定标记定位表格，另一种方式将web页面进行解析，利用解析得到的视觉、空间信息对表格进行抽取。

4.利用自然语言处理（NLP）的方法来抽取表格，这是一种有效的处理方法，但需要大量的例子训练，且执行速度较慢。

5.本体的方法，本体是概念体系明确的部分描述，利用本体对抽取页面的类型进行描述，并根据领域特点设计出本体框架，以用于后续的匹配规则设计。这种方法可以独立于抽取的页面格式，当领域改变时，也只需要改变应用本体即可，应用效果良好，但本体构造中的本体框架设计十分困难，而且，它的应用范围有所限制：必须是数据丰富的，从本体宽带方面来说是窄的web文档。[9]

此外，还可以通过基于语义和数据特征识别表格中的关系元组等对表格进行数据抽取，这里不作介绍。

2.4 csv文件概述

csv（*.csv）文件作为一种通用的文件格式，以纯文本的方式存储数据，常用于电子表格或者数据库的应用。CSV，即COMMA SEPARATED VALUE，指以半角逗号作为分隔符，将各字段分离出来的一种纯文本文件，其文件内容由ASCII字符集中的字符构成。csv文件可以很容易地被导入各种PC表格及数据库中，从而用于不同程序之间数据的交互。[10]csv文件中有如下语法约束：

1.csv文件的每一行都由一个或多个字段组成，每一行代表一条记录，每一行以回车符作为结尾。最后一行可以没有换行符。

2.若一个字段的内容中含有半角逗号，则该字段需要用双引号将其括起来。

3.若一个字段的内容中含有换行符，则需要用双引号将该字段括起来。

4.若一个字段的内容中含有双引号，则该字段首尾用双引号括起来，字段中出现的双引号用两个双引号代替表示。

5.第一行可能是一个头信息，这一行与后面的行格式相同，有相同字段数。

6.每一行记录由半角逗号将字段分隔开，每条记录的字段数要求相等。每条记录的最后一个字段后不可以使用半角逗号。字段中的空格不被忽略。

一般来说，绝大部分的csv文件中的字段都是用双引号括起来的。

2.5小结

本章介绍了本文研究所需的理论基础，包括所需处理文件格式（csv文件）的概述和语法知识，表格的组成成分和分类，以及表格抽取理论的概述、相关方法的介绍和比较等。

第三章系统设计与实现

3.1系统的总体流程

表格抽取的过程可以分为表格定位、结构识别、数据抽取以及抽取数据的表示等。根据表格抽取的过程，结合本文的研究内容，本文自主开发了基于csv文件的表格自动抽取及结构化存储系统。系统由文件预处理、结构识别、数据抽取和数据存储几个部分组成。对于待抽取的表格，首先对表格中拆分错误的单元格进行合并处理，接着识别表格中的属性单元格，对表格的展开方式进行分析，判断表格的结构，将表格中的数据生成相应的“属性-值”对，将数据存储到数据库表内，从而实现了对csv文件中表格数据的自动抽取与结构化存储。系统的难点在于如何识别文件中拆分错误的单元格并将其正确合并以及对表格结构的判断。

3.1.1系统设计的整体要求

系统要求对于一个金融类型、以csv文件格式存储的表格（这里特指交易报告书中的描述标的资产概况、股权结构、标的资产最近两年及一期主要财务数据、标的资产最近三年及一期主要财务数据（如涉及借壳上市）、标的资产主要权属、对外担保及主要负债情况的表格），系统能够自动分析出其属性区域和数据区域，并将数据项逐个存入数据库表中。

3.1.2系统的主要功能模块

根据系统的实现流程，本文将系统分为以下几个功能模块（如图3-1）。

图3-1 系统流程图

系统中各模块的介绍如下：

预处理模块：将待抽取的csv文件读入，将文件中的表格转化成二维数组的形式存储。

单元格合并模块：分析表格中是否存在拆分错误的单元格，若存在，分析其错误类型，并根据其错误类型将拆分错误的单元格正确合并。

属性名称学习模块：通过对训练集中表格的属性单元格的学习，获得可与实验集表格内容匹配的属性名称，作为属性名称匹配模块中的匹配模板。

属性名称匹配：将待抽取的表格中每一项内容与系统预先编写的模板匹配，识别表格中的属性单元格。

展开方式分析：根据匹配后的结果，分析表格内属性行或属性列的位置，划分出表格的属性区域和数据区域，从而确定表格的展开方式。

数据库存储：根据上述分析结果，将表格中的数据生成相应的“属性-值”对，存入数据库表内。

3.2开发工具介绍和分析

3.2.1开发环境

系统使用java语言作为开发语言，配置如下：

开发环境：MyEclipse 2015

服务器：tomcat 6

数据库：MongoDB 3.6.3

3.2.2开发工具的介绍和分析

1.tabula

tabula是一种从PDF文档中提取表格的工具， tabula允许用户使用简单易懂的界面将PDF文档中的数据提取到CSV或Microsoft Excel电子表格中。可以在Mac，Windows和Linux上运行，使用方便。tabula广泛应用于各种规模的新闻机构，为这些机构提供调查性报道，机构包括ProPublica、伦敦时报、外交政策、LaNación（阿根廷）、纽约时报和圣保罗（MN）先锋报等；许多研究人员利用tabula将PDF报告转换为Excel电子表格、CSV和JSON文件，以用于分析和数据库应用程序。[11]本文选择tabula 1.0作为截取表格的工具，将由tabula截取保存的csv文件作为数据源。

2.MongoDB

MongoDB是一个基于分布式文件存储的、由c++语言编写而成的数据库，致力于为Web应用提供一种可扩展的、高性能的、数据存储的方法和工具。MongoDB介于关系数据库和非关系数据库，提供的功能是非关系数据库中最丰富的、最像关系数据库的。可以处理非常松散的数据结构，存储较复杂的数据类型，支持的查询语言十分强大，几乎可以实现类似关系数据库表单查询的绝大部分功能，并且支持对数据建立索引[12]。根据本文的研究内容，选择MongoDB作为数据存储的数据库。

3.3系统各功能模块的设计与实现

3.3.1预处理模块

对于一个待处理的csv文件，预处理模块将文件内容逐行读入，转化为相应的二维数组，以用于系统的后续处理。csv文件以半角逗号作为分隔符，将各字段分离出来，当字段中存在特殊字符时，该字段首尾以双引号括起来。根据csv文件的语法规则，预处理模块通过对csv文件中特殊字符的识别将每一行记录分割为多个字段，从而得到数组中每一项的内容。

csv文件中的特殊字符有半角逗号、双引号和换行符。当遇到特殊符号时，我们需要判断该字符是分隔符还是字段里的内容：若为分隔符，则当前字段内容已经存储完毕，可以开始下一字段的存储；若为字段里的内容，则只需将其当做普通字符，存入当前字段即可。系统对于换行符的处理较为复杂，由于模块使用readline()函数逐行读入文件，因此字段内的换行符会对每行记录的读取造成干扰，使得部分行记录读取不完整，例如对于这样一条记录：

项目，”2015年12月31日/\n2015年度”，”2014年12月31日/\n2014年度”

读取时我们希望得到这样的结果：

项目，”2015年12月31日/\n2015年度”，”2014年12月31日/\n2014年度”

但实际上我们的读取结果是：

项目，”2015年12月31日/

行记录内容的缺失可能会造成数组内容的分割错误。因此，如何完整地读入一行记录内容是本模块的难点。

为了处理字段内换行符，系统首先将初始读入的每条记录存入列表中，然后对内容读取不完整的记录进行合并，再将列表转化为数组。csv文件规定，若字段内容中含有换行符，则该字段首尾需用一对双引号括起来。根据这条规则，结合对csv文件内容的观察，我们可以得到这样的规则：若读取的一条记录中的双引号个数为奇数，则分割出这条记录的换行符是字段里的换行符，记录内容不完整，需对其进行合并处理；若这条记录中的双引号个数为偶数，则这条记录已经读取完毕，无需其他操作。根据这条规则，我们对列表进行遍历，对于需要合并的记录，把它与后续相连的记录合并，若合并后双引号个数为偶数，则当前记录读取完毕，合并结束，若为奇数，则继续与后续相连的记录合并，直到合并后记录的双引号个数为偶数为止。这样便可以切分出每一行记录。根据最终记录的个数和每一条记录中半角逗号数可以计算出表格的行列数，以用于接下来数组的创建。

将文件中每行记录正确切分后，便可以利用记录中的半角逗号将记录拆分成字段，转化为数组s。这部分的重点在于对半角逗号性质的判断，csv文件规定：若半角逗号在字段内，则用一对双引号将这个字段括起来。由此我们可以得出以下结论：当遇到半角逗号时，若当前记录已读入的内容中双引号个数为奇数，则此半角逗号属于字段中的内容，为普通字符，存入当前字段中；若当前记录已读入的内容中双引号个数为偶数，则此半角逗号为分隔符，当前字段存储完毕，更新数组下标，开始下一字段的存储。依照上述结论遍历每一条记录后，便可以得到对应的数组s。

分割字段时我们并没有对记录中的双引号做特殊的处理，只是将双引号当做普通字符直接存入字段中。数组生成后为了方便系统后续对表格内容的匹配和存储，我们需要去除字段首尾的双引号：遍历数组中的每一项，若一项内容的首尾均为双引号，则将首尾的双引号去除，从而将文件成功地转化为数组结构。

3.3.2单元格合并模块

tabula在截取表格时经常会出现单元格拆分错误的问题，导致单元格内容存储不完整。内容不完整的单元格对于后续属性名称的匹配、数据的存储等都是影响巨大的。然而有关这方面的研究内容很少，所以单元格合并模块既是系统的重点，也是难点。这一部分要求对于预处理模块生成的数组s，寻找拆分错误的单元格，若存在拆分错误的单元格，还需判断其错误类型，并对拆分错误的单元格进行合并处理，从而得到内容正确的表格。

tabula截取表格时有两种方式，一种是按字符流截取，根据每个字符在PDF页面上的绝对位置截取表格，一种是按边界截取，通过对表格中单元格分界线的识别实现对表格的截取。截取过程中，由于单元格内的多行内容、单元格内字符的位置、中文表格边界线识别问题等，会产生拆分错误的单元格，本文将拆分错误的单元格归纳为以下几种情况：

1.行拆分错误：当一个单元格内有多行内容时，tabula会为每一行内容生成一个单元格，即一个含有n（n>1）行内容的单元格被从上到下拆分成n个单元格。根据拆分后单元格中非空单元格的分布位置，行拆分又可以分为对称的（如图3-2）和不对称的（如图3-3），对于对称的单元格拆分情况，拆分后的n行单元格可以围绕其中间一行进行合并，而不对称的单元格拆分则可以将拆分后的n个单元格从下到上依次合并。

图3-2 行拆分错误（对称情况）：上图为源表格，下图为tabula截取结果

图3-3 行拆分错误（不对称情况）：上图为源表格部分截图，下图为对应部分tabula截取结果

2.列拆分错误：对于表格中的一列单元格，若每个单元格之间内容的分布位置差别较大，tabula按字符流方式截取表格时，这一列单元格可能会被拆分成两列（如图3-4），拆分前这一列里的每个单元格的内容存放在拆分后两列中对应行的两个单元格中的其中一个里，另一个为空白单元格。

图3-4 列拆分错误：上图为源表格，下图为tabula截取结果

3.空白行列错误：tabula按边界线截取表格时，截取的表格中可能会出现n行或n列（n≥1）完全空白的单元格（如图3-5）。

图3-5 空白行列错误：上图为源表格，下图为tabula截取结果

单元格合并模块主要针对上述几种错误进行识别和处理。首先需要判断表格中是否存在拆分错误的单元格，如果存在，再判断其拆分错误的类型。这个问题可以借助每行、每列中空白单元格的个数来解决。为了更方便地处理数据，系统引入了一个新的二维数组n，来记录表格中每一项是否为空白单元格。数组n中每一项与数组s中每一项相对应，若数组s中一项为空白单元格，则数组n中对应的项标记为0，否则标记为1。通过对每行、每列非空单元格个数进行统计来判断当前表格是否存在拆分错误的单元格：若表格中的一行存在空白单元格，则表格中可能存在行拆分错误或者空白行，若表格中的一列存在空白单元格，则表格中可能存在列拆分错误或空白列，若表格中不存在空白单元格，则表格中没有拆分错误的单元格，可直接进行系统的下一步处理。

确定表格中可能存在的单元格拆分错误类型后，即可针对不同的错误类型分别进行处理。下面对每一种拆分错误的解决方法分别进行介绍。

1. 行拆分错误（对称情况）

对称情况下的被拆分形成的行可以围绕这些行正中间的一行形成对称，通过定位中间的这条对称轴及两边围绕它的行数进行合并，但是对称轴形成的情况是复杂的。在定位对称轴时经常可以遇到以下几种问题：

（1）递归合并：合并的n行单元格中，除了中间那条对称轴，可能还存在其他的对称轴（如图3-6）。这样可能无法正确得到两边对称的行数。

图3-6 递归合并示例表格：上图为源表格，下图为tabula截取结果局部图

（2）不同行交叉合并：拆分后的某些行可能会与原本不属于同一单元格内的行形成对称（如图3-7），使得不同单元格错误地合并为一个。

图3-7 交叉合并示例表格：上图为源表格部分截图，下图为tabula截取结果局部图

（3）不需要的行合并：原本不需要合并的行之间互相对称（如图3-8），导致多个单元格内容合并在一起，造成了不必要的合并。

图3-8 不需要的行合并示例表格

因此对称情况下行拆分错误处理的难点在于对称轴的定位。为了减少对上述错误对称轴的识别，我们以一列中非空白单元格个数最少的列为基准进行遍历寻找，系统认为只有这列中非空白单元格所在的行才可能形成对称轴。通过对对称的行拆分错误情况进行归纳分析，生成以下规则：

（1）对于某一行，若其上一行与其下一行中的每一项，从前到后两两一组，其n值全部相等，则这两行围绕中间那行形成对称。

（2）根据合并的递归性，对于某一行，可以从内到外逐行判断，若当前行的上i行和下i行分别对称，则判断其上第i+1行和下第i+1行是否对称，直到判断的两行不对称为止。

（3）若某一行的上下两行不存在空白单元格，则这两行不参与合并。

根据上述规则，系统以之前选定的列为基准，对数组进行遍历处理，定位出对称轴的位置以及对称轴两边的最大对称行数，将其存入数组sum中。之后遍历sum数组，若一行的sum值大于0，则当前行为对称轴，将其上sum行到其下sum行，从上到下依次合并，最后将空白行删除即可。需要注意的是，这样选定的对称轴虽然不包含多余的对称轴，却不一定能定位表格中

所有的对称轴，合并后的数组可能仍存在待合并的行（对称情况）。

2. 行拆分错误（不对称情况）

不对称的行拆分情况则简单许多。对于含有空白单元格的某一行，若这一行中每一个非空的单元格，它上面行中对应的单元格都是非空的，就将这一行合并到上一行。

但是在实际处理时，由于表格中可能本来就存在空白的单元格，而系统无法区分原表格中存在的空白单元格与tabula截取表格时产生的空白单元格，所以这种错误的处理效果并不好，鉴于这种拆分错误的单元格对后续属性名称的识别和表格展开方式的识别的影响较小，目前系统对这种拆分错误不作处理。

3. 列拆分错误

列拆分错误将一列的内容分成两列展示，这两列的内容是互补的。也就是说，对于拆分后形成的两列，这两列中非空单元格数量的总和是原来一列中非空单元格的个数，小于等于表格的行数，并且对于这两列中每一行的两个单元格，最多只有一个单元格的内容是非空的，不可能出现两个非空单元格。根据这种互补性，我们对表格的列进行遍历，寻找可能合并的列，将两列中的右面那列的内容移动到左面那列，再删除右面的列即完成了对列拆分错误的处理。

4. 空白行列

空白行列既可以在tabula截取表格时生成，也会在处理行列拆分错误时产生。对于空白行列的处理，只需要分别对表格进行逐行、逐列遍历，若一行或一列中非空白单元格个数为零，就将其删除，从而得到没有空白行列的表格。

根据多次实验经验可知，列合并错误对行合并有一定的影响，而空白行列的错误在每一次合并后都可能产生，因此在单元格合并模块中，我们采用以下流程判断、处理各种拆分错误（如图3-9）：

图3-9 单元格合并流程

3.3.3属性名称学习模块

属性名称学习模块相对独立于表格抽取及结构化存储系统，学习模块通过对训练集表格中属性名称的学习，为之后的属性名称匹配模块提供匹配的模板。属性名称的学习采用了基于例子的学习方法，利用统计等方式对训练集表格中的属性名称进行学习，本模块的难点在于同义词的寻找。通过对常见表格抽取算法的学习与比较，我们利用了Tengli等人在表格抽取时使用的方法[6]实现本模块功能。

Tengli等人使用了基于例子的学习方法，在学习过程中利用相对编辑距离寻找合并了同义词。他们认为表格中有两种类型的信息：标签信息和数据。标签是数据的属性，也就是我们所说的属性名称。如果表格中的一些标签已知，那么这些信息可以用来识别表格的结构，反过来它可以帮助为数据单元格赋值正确的标签。由于表格以半结构化形式表现数据，标签以规律的结构出现，即连贯的标签单元格出现在一行或一列。定位表格中已知的标签可以帮助从数据的行和列中区分标签的行和列。根据其系统提供的标记了标签的示例表格，应用如下算法进行学习：

1.提取示例表格中的标签并对其建立索引。

2.计算标签的相对字符串编辑距离，数值小于0.09的标签被合并在一起。相对编辑距离是编辑操作次数（Levenstein,1996）与字符串长度的比率。相对编辑距离度量是不对称的，因此我们选择两个相对编辑中的较大的距离进行比较。

3.将处理后的标签进行排序，供后续抽取算法使用。

我们利用相对距离编辑算法来寻找同义词。编辑距离（Minimum Edit Distance，MED），又称Levenshtein距离，是指将一个字符串转化为另一个字符串的最少编辑操作次数。在转化过程中，我们只可以使用替换（substitution，s）、插入（insert，i）或者删除（delete，d）操作。相对编辑距离的使用可以帮助我们寻找并合并同义词。[13]相对编辑距离的阈值是一个最大性能的经验值。

通过对上述算法进行修改，实现了系统中的属性名称学习模块。我们人工识别训练集表格里的属性名称，将其取出存入同一个txt文件中，通过对文件内容的遍历，对属性名称进行统计排序，得到初始的属性名称序列，对序列中的属性名称，每两个一组计算它们的相对编辑距离，高于阈值的一对属性名称即为同义词，对同义词进行记录并合并，从而得到最终的属性名称序列。为了提高我们的学习效率，我们将训练集中的表格分类，分别进行学习，其中对于阈值的设定，我们对于不同的阈值分别进行实验，对比实验结果，最终选择0.65作为阈值。

系统采用动态规划的方法实现编辑距离的求解，定义D(i,j)为源字符串src的子串src(0,i)变化为目标字符串dst的子串dst(0,j)的编辑距离，那么D(src.length(),dst.length())即源字符串src到目标字符串dst的编辑距离。其初始化及递归式如下：

初始化：

D(0,0)=0

D(i,0)=D(i-1,0)+del[x(i)];1<i<=N

D(0,j)=D(0,j-1)+ins[y(j)];1<j<=M

递归式：

D(i,j)=min{D(i-1,j)+del[x(i)],D(i,j-1)+ins[y(j)],D(i-1,j-1)+sub[x(i),y(j)]}

3.3.4属性名称匹配

属性名称匹配模块通过将数组与编写的模板进行匹配，可以识别出表格里的部分或全部属性名称。模板是根据属性名称学习模块得到的属性名称序列编写的正则。将数组中的每一项与正则中的每一项进行匹配，若匹配成功，则这一项的内容是属

性名称，系统设置了mark数组来存储数组s中每一项的属性，mark值为1代表表格中对应的这一项是属性单元格，mark值为0代表表格中对应的这一项是数据单元格。

匹配模块中的识别效果主要依赖于编写的正则，如何使正则能够准确的表达模板的内容是这个模块的难点，也是整个系统的难点之一。正则的调整主要通过属性名称学习模块的学习结果和匹配过程中的实验现象来调试。

3.3.5展开方式分析

展开方式分析根据属性名称模块中的匹配结果确定表格的属性行或属性列位置以及表格的展开方式，从而确定表格的结构。由于表格以半结构化形式表现数据，属性单元格的出现是规律的，即连贯的属性单元格出现在一行或一列，因此定位表格中已知的属性单元格可以帮助从表格的行和列中区分属性单元格的行和列。例如，如果一列中50%的单元格被确认为属性名称，那么这一列中剩下的单元格也是属性名称。这意味着如果一列中大部分单元格是属性名称，那么剩余的单元格也是属性名称。

根据上述规则，我们通过比较每行或每列中属性单元格的比例值来确定属性区域。对于行来说，比例值指一行中属性单元格个数与列数的比，同理可得列的比例值是一列中属性单元格个数与行数之比。比例值越大，则这一行或这一列是属性区域的可能性就越大。首先，系统通过对mark数组的遍历，得到每行属性单元格的比例值，在所有的行中选择比例值最大的行作为备选的属性行，同理，通过对mark数组的遍历也可以选择出备选的属性列。备选的属性行与属性列中比例值更大的那个就是我们最终确定的属性区域，由属性区域的位置也可以确定表格的展开方式。

需要注意的是，当表格的行数或列数为2时，仅对比备选的属性行与属性列的比例值很可能会造成判断错误。例如，对于图3-10所示表格（加粗内容为识别到的属性单元格），每行的比例值依次为：

0.5 0.0 0.5 0.0 0.5 0.0 0.0 0.5 0.5

每列的比例值依次为：

0.44 0.11

按上述算法分析得到的属性区域为第1行，这显然是不正确的。因此当表格的行数或列数为2时，只有当较大的比例值大于阈值时，我们才认为上述判断有效，否则，认为左面列（列数为2时）或上面行（行数为2时）为属性区域，这里阈值取0.5。

图3-10 示例表格

3.3.6数据库存储

数据库存储模块将分析处理好的表格根据其展开方式按行或按列将数据项逐个取出，存入数据库表中。数据库的操作步骤比较简单，只要连接数据库，选择相应的集合，然后将每条记录以文档的形式插入即可。

数据库存储模块的难点在于文档的生成，对于MongoDB数据库，插入文档，实质上是插入一组组“属性-值”对，其中属性名称为属性，数据项为值，对于按行展开的表格，将每一数据行中的每一项与其对应的属性名称进行配对，生成“属性-值”对，添加到文档中，然后将多条文档插入数据库即可，生成“属性-值”对时要注意对行属性的判断（表格中可能有多个属性行）。对按列展开的表格，处理原理相同。

3.4小结

本章节详细介绍了系统的开发环境和系统流程、功能模块，并详细的对每个模块的功能和实现原理、算法进行了介绍。

第四章实验结果及分析

4.1实验结果

实验选取了项目中的24篇文档作为实验数据源，其中的8篇文档作为训练集，剩余16篇作为实验集。利用tabula工具从PDF文档中随机截取文章中标的资产概况、股权结构、标的资产最近两年及一期主要财务数据、标的资产最近三年及一期主要财务数据（如涉及借壳上市）、标的资产主要权属、对外担保及主要负债情况部分的表格，以csv文件格式存储，作为实验对象。对系统中的每一个模块分别进行测试，人工评判实验结果。其中属性名称学习模块不做单独测试，其余各个模块测试结果如下：

1.预处理模块：将文件中表格正确转化为二维数组并去除每个字段首尾的双引号（如果存在的话）则视为成功。本模块共测试333个表格，其中319个表格测试成功，正确率为95.80%。

2.单元格处理模块：将转化成功的二维数组中存在的拆分错误的单元格合并，使得处理后的表格所有单元格合并正确，并且其内容全部存储完整视为成功。本模块测试了预处理模块中处理成功的319个表格，其中281个表格测试成功，正确率为88.09%。

3.属性名称匹配与展开方式分析：测试时将这两个模块一起测试，对于处理后的二维数组，能正确匹配其中部分或全部属性名称并正确地定位属性区域和数据区域以及表格的展开方式，则视为成功。测试选取单元格处理模块处理后的304个表格（包括含有不对称行拆分的表格）中的训练集之外的203个表格进行测试，根据测试表格的类型分为4大类进行测试，测试结果分别为：

财务数据表格：共50个，正确识别47个，准确率为94.00%；

主要资产表格：共66个，正确识别54个，准确率为81.82%；

股份结构表格：共72个，正确识别72个，准确率为100%；

基本信息表格：共15个，正确识别15个，准确率为100%；

总体准确率：188/203=92.61%。

4.存储模块：将分析处理后的数组中的数据逐项存入数据库表内，存储正确则视为成功。本模块测试处理正确的188个表格，全部都可以存储成功，正确率100%。

总体测试时，将文件读入并正确存储到数据库表内视为成功，共测试228个表格，正确存储168个，正确率为73.68%。

4.2分析

根据上述实验结果，结合系统抽取失败的表格和代码，对每个模块存在的问题进行分析，并寻找可能的解决方案。每个模块的分析如下：

1. 预处理模块：本模块转化失败的表格均属于跨页表格，tabula截取跨页表格时，对上下两个子表格是分别处理的，因此两个子表格可能分别出现不同的单元格拆分错误，导致上下两个子表格列数不同，从而在将表格转化为数组时出现数组下标越界问题。例如图4-1所示的一个4列跨页表格，tabula处理时，上页子表格增加了一列空白单元格，使得上页子表格列数变为5，下页子表格列数仍为4，与源表格列数相等。预处理模块选择较小的列数值4为数组的二维下标范围，导致表格转化时，上页子表格存储出现了数组下标越界问题。

图4-1 跨页表格示例：上图为源表格局部截图，下图为对应部分tabula截取结果

对于这类跨页表格，可以考虑在截取表格时将上下两个子表格分别保存到两个csv文件中，将两个子表格分别进行预处理和单元格合并处理后再合并。或者考虑以较大的列数值为基准建立数组，以保证转化过程中不会发生数组下标越界问题。

2. 单元格合并模块：造成单元格合并失败的原因有很多，如没有对行拆分错误（不对称情况）单元格进行合并、复杂结构无法处理、跨页表格列未对齐等。下面针对常见的几种错误情况分别进行分析。

（1）不对称行拆分错误：没有对不对称行拆分错误的单元格进行合并，是单元格合并失败的主要原因。由3.3.2节单元格合并模块部分的讲解可知，对于一行含有空白单元格的行，若对于这一行中所有的非空白单元格，它上面一行对应位置的单元格都是非空的，就可以进行不对称行拆分的合并。实验中符合这种合并条件的表格可以分为两种：需要合并的（如图4-2）和不能合并的（如图4-3）。系统需要判断这一行中空白单元格的性质来判断表格是否需要合并。对于需要合并的表格来说，这些空白单元格是在tabula处理表格时额外增加的，而对于不需要合并的表格，这些空白单元格是源表格中本来就存在的，其内容为空。但是系统目前无法判断空白单元格的性质，所以系统目前无法对不对称行拆分错误的单元格进行处理。

对于空白单元格性质的判断，目前还没有比较有效的方法，可以考虑对源表格中存在的空白单元格进行标记，如在空白单元格中输入“-”等特殊符号，以区分于tabula截取时产生的空白单元格，然后再应用系统中的方法处理表格。

图4-2 存在不对称行拆分错误的表格示例：上图为源表格，下图为tabula截取结果

图4-3 表格示例：示例表格符合合并条件，但不需要合并

（2）复杂结构表格：目前系统只能处理简单的二维表格，对于结构复杂的表格的处理情况不是很好。例如图4-4所示表格，表格的第三列中存在嵌套单元格，导致系统无法正确判断单元格拆分错误的类型，因此无法处理。

图4-4 复杂结构表格示例

对于这些复杂结构的表格，可以将表格先进行内容规范化，将嵌套单元格中的内容复制到每一个子单元格。例如，对图4-4的表格进行内容规范化后，表格转化为图4-5所示表格，然后再对表格进行合并处理。

图4-5 图4-4表格内容规范化后表格

（3）跨页表格：对于一些跨页表格，tabula截取后下面子表格的列无法正确对应到上面子表格的每一列，导致部分单元格的内容无法正确合并。例如图4-6所示的一个两列的表格，tabula截取表格时，将下面子表格处理为一个只有一列的表格，这一列内容对应上面子表格的第一列，导致下面子表格的内容无法正确合并到上面子表格中。

对于这类问题，目前还没有有效的解决方法，只能在系统处理前预先对文件进行修改，人工将两个子表格中每一列对应。

图4-6 跨页表格示例：上图为源表格部分截图，下图为截图部分对应的tabula截取结果

（4）对称轴判断错误：对于对称情况下的行拆分错误，系统在定位对称轴时，以一列中非空白单元格最少的一列为基准

进行判断，对于大部分表格，我们都可以正确定位对称轴，但对于少数表格却会定位错误。例如图4-7所示表格，系统首先选定第一列为基准列，然后找到了两条对称轴（图中阴影行为对称轴），其中第一条定位错误，同时影响了对第二条对称轴两边对称行数的判断。若系统以第二列为基准列，则可以正确定位对称轴并进行合并。

对此，我们可以考虑选择一个更合理的基准列来定位对称轴。或者当表格列数较小时，增加基准列的数量，通过对称轴的多次定位，综合判断对称轴位置，进行单元格的合并。

图4-7 对称轴定位错误示例表格：上图为源表格，下图为tabula截取结果

（5）其他拆分错误：有些表格的单元格拆分错误不属于系统中可判断处理的错误，如图4-8所示表格，由于源表格中阴影区单元格内容位置的偏差，使得tabula截取的表格中无法形成正确的对称轴。

这种错误暂时没有有效的解决方法，目前只能通过人工调整文件内容，使表格中的单元格拆分错误尽可能符合系统可处理的类型，方便系统处理。

图4-8 其他错误类型表格示例：上图为源表格，下图为tabula截取结果

3. 属性名称学习模块：属性名称学习模块主要是为系统提供匹配模板。模板的学习依赖于学习算法和训练集。因此本模块的优化可以从训练集入手，通过调整训练集，使其涵盖的表格类型更加广泛，调整阈值，提高对同义词的判断与处理。

4. 属性名称匹配与展开方式分析：通过对实验结果的分析，这部分失败的原因主要在于属性名称匹配失败，属性名称的匹配依赖于系统编写的正则，根据表格与正则匹配的情况具体又可以分为两种类型。

（1）同义词匹配失败：对于图4-9所示表格，系统编写的正则中虽然含有“原值、累计折旧、净值、成新率”，但由于示例表格中这些词的后面含有单位，使得正则无法与之匹配。这说明目前正则的匹配能力还不够高，无法识别一个词的所有表现形式。因此可以通过对正则的改进，使其尽可能涵盖每个属性名称的所有表现形式。

图4-9 同义词匹配失败表格示例

（2）正则涵盖不完整：有些表格中的属性名称及其同义词均不存在于正则中，导致表格中的属性名称无法匹配。这些表格主要属于资产类表格，常见无法匹配的有软著类型表格、土地情况表格和担保情况表格等（如图4-10）。这类问题可以通过对正则的优化来解决：更改训练集，令训练集中包含的表格类型尽可能全面，这样训练得到的模板才可能尽量地涵盖各种表格，从而使编写的正则可以匹配更多的表格。

图4-10 正则涵盖不完整表格示例

此外，由于对实验结果的评估涉及到表格中的具体内容，目前对实验结果正确性的评估只能依靠人工检查，没有一个自动化的评判方法，耗费了大量人力。下需要为系统生成一个合理的、规范的自动评判标准，使系统能够更加方便快捷地对表格的抽取结果进行判断。

6. 第五章结论

总字数：520

相似文献列表　文字复制比：10.4%(54)　疑似剽窃观点：(0)

1	金融领域信息的自动抽取与分析方法 马晶晶(导师：王晓龙) -《哈尔滨工业大学博士论文》- 2012-12-01	10.4%（54） 是否引证：否

原文内容　红色文字表示存在文字复制现象的内容; 绿色文字表示其中标明了引用的内容

第五章结论

本文自主开发了一个表格信息自动抽取及结构化存储系统，针对交易报告书中的部分金融类表格，采用了基于例子的学习方法，将表格中数据项逐个取出存入数据库表内，同时基于tabula的工作原理处理了表格中拆分错误的单元格。

系统对表格的抽取正确率仍有待提高，其问题主要在于复杂结构的表格无法正确处理、跨页表格的列不对应、以及训练集的设置问题，接下来将主要围绕这三个方面对系统进行改进。如对表格内容进行规整、将跨页表格分别处理、更换训练集等。同时针对系统的表格抽取结果形成一个自动化的评判标准。

参考文献

[1] 苏溢. 浅析信息化对金融行业作用[J]. 中小企业管理与科技(上旬刊), 2013(3):236-236.

[2] 潘小燕. 半结构化文本中的表格信息抽取技术的研究[D]. 哈尔滨工业大学, 2007.

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

[4] 李保利, 陈玉忠, 俞士汶. 信息抽取研究综述[J]. 计算机工程与应用, 2003, 39(10):1-5.

[5] 赵春玲. 基于本体的网页中非标记表格抽取的研究与实现[D]. 哈尔滨工业大学, 2007.

[6] Embley D W, Tao C, Liddle S W. Automatically Extracting Ontologically Specified Data from HTML Tables of Unknown Structure.[C]// International Conference on Conceptual Modeling. Springer-Verlag, 2002:322-337.

[7] Tengli A, Yang Y, Ma N L. Learning table extraction from examples[C]// 2004.

[8] 赵洪, 肖洪, 薛德军,等. Web表格信息抽取研究综述[J]. 数据分析与知识发现, 2008, 24(3):24-31.

[9] 王放, 顾宁, 吴国文. 基于本体的WEB表格信息抽取[J]. 小型微型计算机系统, 2003, 24(12):2142-2146.

[10] 李旭, 马力. VB6在CSV文件格式处理中的应用研究[J]. 信息技术, 2009(7):26-28.

[11] https://tabula.technology
[12] https://baike.baidu.com/item/mongodb/60411?fr=aladdin
[13] https://baike.baidu.com/item/编辑距离/8010193?fr=aladdin

致谢

本论文的研究在各位老师、同学的帮助下顺利完成，在此感谢邹淑雪老师对我一直以来的教导，老师严谨的态度、丰富的经验给我提供了许多研究思路和帮助，使我受益匪浅，在此向邹老师表示衷心的感谢。

感谢王岩老师和李玉华老师，在论文的开题、设计阶段等阶段给予了我耐心的指导，在此向他们表示感谢。

感谢项目组的熊梦婷学姐和李相臣同学等人在我研究阶段给我的知识和技术上的各种帮助，一起讨论、研究问题，互相勉励，使我的研究顺利进行。

最后，向辛勤培育和教导我成长的父母表达我最诚挚的敬意和最由衷的感谢，感谢他们这些年来对我的默默支持和辛勤付出！

指　标
疑似剽窃文字表述

1. 最后，向辛勤培育和教导我成长的父母表达我最诚挚的敬意和最由衷的感谢，感谢他们这些年来对我的默默支持和辛勤付出

说明：1.总文字复制比：被检测论文总重合字数在总字数中所占的比例

2.去除引用文献复制比：去除系统识别为引用的文献后，计算出来的重合字数在总字数中所占的比例

3.去除本人已发表文献复制比：去除作者本人已发表文献后，计算出来的重合字数在总字数中所占的比例

4.单篇最大文字复制比：被检测文献与所有相似文献比对后，重合字数占总字数的比例最大的那一篇文献的文字复制比

5.指标是由系统根据《学术论文不端行为的界定标准》自动生成的

6.红色文字表示文字复制部分;绿色文字表示引用部分

7.本报告单仅对您所选择比对资源范围内检测结果负责

amlc@cnki.net

http://check.cnki.net/

http://e.weibo.com/u/3194559873/