南京大學

研 究 生 毕 业 论 文

（申请工程硕士学位）

论 文 题 目 面向金融 APP 的隐私政策合规性检测系统

作 者 姓 名 吴青衡

学科、专业名称 工程硕士（软件工程领域）

研 究 方 向 软件工程

指 导 教 师 陈振宇 教授

2022 年 5 月 20 日

学　　号 ：**MF20320177**

论文答辩日期 ：

指 导 教 师 ：　　（**签字**）

Privacy Policy Compliance Detection System for Financial APP

By

Qingheng Wu

Supervised by

Associate Professor Zhenyu Chen

A Thesis
Submitted to the Software Institute
and the Graduate School
of Nanjing University
in Partial Fulfillment of the Requirements
for the Degree of
Master of Engineering

Software Institute

May 2022

学 位 论 文 原 创 性 声 明

任何收存和保管本论文的单位和个人，未经作者本人授权，不得将本论文转借他人并复印、抄录、拍照或以任何方式传播，否则，引起有碍作者著作权益的问题，将可能承担法律责任。

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不句含其他个人或集体已经发表或撰写的作品成果。本文所引用的重要文献，均已在文中以明确方式标明。本声明的法律结果由本人承担。

论文作者签名：_____

日期：　　年　　月　　日

南京大学研究生毕业论文中文摘要首页用纸

毕业论文题目：面向金融 APP 的隐私政策合规性检测系统

工程硕士（软件工程领域）专业 2020 级硕士生姓名：吴青衡

指导教师（姓名、职称）：陈振宇 教授

摘 要

移动互联网的快速发展给人们生活带来了诸多便捷，但个人信息被滥用的问题也变得愈发严重，为了保证个人信息的安全，国家陆续出台了多款法律法规来规范 APP 隐私政策的制定。相较于普通 APP 而言，金融 APP 同时受到了国家和金融行业的双重监管，因此金融 APP 的隐私政策不仅需要符合国家法律法规还要符合金融行业的要求。同时，由于金融 APP 涉及了大量的个人敏感信息，相关法规还对金融理财类、手机银行类以及网络借贷类等不同类型的金融 APP 分别规定了必要的个人信息收集范围，隐私政策里不得随意收集非必要的个人信息。如果金融 APP 的隐私政策没有符合相关规定，则面临被下架、业务停摆甚至巨额罚款的风险。

为此，本文设计了一种面向金融 APP 的隐私政策合规性检测系统，相较于传统 APP 的隐私政策检测，本文根据国家法规及金融行业规定，设计了一套专门适用于金融 APP 的检测指标；本文收集整理了大量金融 APP 的隐私政策文本，构建了一份用于机器学习的高质量数据集；本文还针对不同类型的金融 APP 提供了专项检测。本系统从隐私政策的内容合规与权限合规两个角度，给用户提供了文本合规性检测和权限合规性检测两种任务。文本合规性检测任务只需要上传隐私政策文本，它包括了隐私条款检测与金融专项检测两个功能。隐私条款检测会根据本文制定的金融 APP 检测指标，先通过 NLP 技术对文本进行处理，然后使用 BERT 模型进行分类，判断文本内容是否满足了所有检测指标；金融专项检测会根据金融 APP 的类型，检测出隐私政策中收集的个人信息是否符合必要收集的范围。权限合规性检测任务需要上传隐私政策文本和待测应用的安装包，通过静态分析和基于 SimCSE 的语义匹配技术，检测出 APP 中使用的权限与隐私政策中声明的权限是否一致。

整个系统服务端采用 Django 开发框架进行开发，前端采用 Vue 框架，算法部分使用 Python 语言实现。经过测试，本系统能够有效检测出金融 APP 隐私政策在文本内容和权限说明上的违规项与缺少项，并通过结果展示界面进行了详细地总结，能够帮助用户高效地完成隐私政策的自查与完善工作。

关键词：隐私政策检测，文本分类，机器学习，静态分析

南京大学研究生毕业论文英文摘要首页用纸

THESIS: Privacy Policy Compliance Detection System for Financial APP

SPECIALIZATION: Software Engineering

POSTGRADUATE: Qingheng Wu

MENTOR: Associate Professor **Zhenyu Chen**

Abstract

The rapid development of mobile Internet has brought a lot of convenience to people's lives, but the problem of personal information being abused has also become more and more serious, in order to ensure the safety of personal information, the state has introduced a number of laws and regulations to regulate the development of APP privacy policy. Compared with ordinary APPs, financial APPs are subject to both national and financial industry supervision, so the privacy policy of financial APPs not only needs to comply with national laws and regulations but also the requirements of the financial industry. At the same time, because financial APP involves a lot of sensitive personal information, the relevant regulations also provide the necessary scope of personal information collection for different types of financial APPs, such as financial finance, mobile banking and online lending, and the privacy policy should not arbitrarily collect non-essential personal information. If the privacy policy of a financial APP does not comply with the relevant regulations, it faces the risk of being taken down, business suspension or even huge fines.

To this end, this thesis designs a privacy policy compliance detection system for financial APPs. Compared with the privacy policy detection of traditional APPs, this thesis designs a set of detection indicators specifically for financial APPs according to national regulations and financial industry regulations; this thesis collects and organizes a large number of privacy policy texts of financial APPs and builds a high-quality dataset for machine learning; this thesis also provides special data sets for This thesis also provides special testing for different types of financial APPs. This system provides users with two kinds of tasks, text compliance detection and permission compliance detection, from the perspectives of content compliance and permission compliance of privacy policy. The text compliance task only requires uploading the privacy policy text,

which includes two functions: privacy clause detection and financial specific detection. The privacy clause detection will first process the text through NLP technology according to the financial APP detection indexes formulated in this thesis, and then use the BERT model to classify and determine whether the text content meets all the detection indexes; the financial-specific detection will detect whether the personal information collected in the privacy policy meets the necessary collection range according to the type of financial APP. The permission compliance testing task requires uploading the privacy policy text and the installation package of the app to be tested, and detecting whether the permissions used in the app are consistent with those declared in the privacy policy through static analysis and the semantic matching technology based on SimCSE.

The whole system is developed with Django development framework on the server side, Vue framework on the front end, and Python language on the algorithm part. After testing, the system can effectively detect the violations and missing items in the text content and permission description of the financial APP privacy policy, and summarize them in detail through the result display interface, which can help users efficiently complete the self-examination and improvement of the privacy policy.

Keywords: Privacy Policy Detection, Text Classification, Machine Learning, Static Analysis

目录

表 目 录 ····· **viii**

图 目 录 ····· **x**

第一章 引言 ····· **1**

1.1 研究背景 ····· 1

1.2 国内外研究现状 ····· 2

1.2.1 隐私政策文本合规性检测技术 ····· 2

1.2.2 权限合规性检测技术 ····· 3

1.3 本文主要工作 ····· 4

1.4 本文组织结构 ····· 5

第二章 相关技术概述 ····· **6**

2.1 中文分词技术 ····· 6

2.2 TF-IDF 算法 ····· 6

2.3 BERT 模型 ····· 7

2.4 SimCSE 模型 ····· 9

2.5 静态分析 ····· 11

2.6 隐私政策语料库 ····· 11

2.7 Django ····· 12

2.8 本章小结 ····· 12

第三章 系统的需求分析与概要设计 ····· **13**

3.1 系统整体概述 ····· 13

3.2 系统需求分析 ····· 14

3.2.1 功能性需求分析 ····· 14

3.2.2 非功能性需求分析 ····· 15

3.2.3 系统用例设计 ····· 15

3.2.4 系统用例描述 ····· 15
3.3 系统整体设计 ····· 20
3.3.1 系统架构设计 ····· 21
3.3.2 4+1 架构视图 ····· 23
3.4 数据库设计 ····· 26
3.5 本章小结 ····· 31

第四章 系统的详细设计与实现 ····· 32
4.1 隐私政策检测指标的构建 ····· 32
4.2 隐私政策数据集的构建 ····· 33
4.3 数据交互模块 ····· 34
4.3.1 时序图 ····· 34
4.3.2 界面展示 ····· 36
4.4 文本处理模块 ····· 38
4.4.1 网页文本获取 ····· 38
4.4.2 章节标题提取 ····· 38
4.4.3 文本段落处理 ····· 40
4.4.4 中文分词及去停用词 ····· 41
4.5 文本合规性检测模块 ····· 42
4.5.1 章节标题分类器的设计与实现 ····· 43
4.5.2 隐私条款检测功能的设计与实现 ····· 44
4.5.3 金融专项检测的设计与实现 ····· 49
4.5.4 时序图 ····· 51
4.6 权限合规性检测模块 ····· 51
4.6.1 构造敏感权限名称表 ····· 51
4.6.2 权限语句集提取方法 ····· 51
4.6.3 基于 SimCSE 的权限合规性检测方法 ····· 54
4.6.4 时序图 ····· 56
4.7 结果生成模块 ····· 57
4.7.1 时序图 ····· 57
4.7.2 界面展示 ····· 58
4.8 本章小节 ····· 60

第五章 系统测试与案例分析 ····· **62**
5.1 测试准备 ····· 62
5.1.1 测试目标 ····· 62
5.1.2 测试环境 ····· 62
5.2 算法性能测试 ····· 63
5.2.1 基于 SVM 的标题分类器性能测试 ····· 63
5.2.2 隐私条款检测中的 BERT 模型性能测试 ····· 63
5.2.3 金融专项检测中的 BERT 模型性能测试 ····· 65
5.2.4 SimCSE 模型性能测试 ····· 66
5.3 真实案例分析 ····· 67
5.3.1 隐私条款检测的测试 ····· 67
5.3.2 金融专项检测的测试 ····· 68
5.3.3 权限合规性检测的测试 ····· 69
5.4 功能测试 ····· 69
5.5 非功能性测试 ····· 71

第六章 总结与展望 ····· **74**
6.1 总结 ····· 74
6.2 展望 ····· 75

参考文献 ····· **76**

简历与科研成果 ····· **80**

致谢 ····· **81**

表 目 录

3.1 功能性需求列表 ····· 15
3.2 非功能性需求列表 ····· 16
3.3 用户登录的用例描述 ····· 16
3.4 政策文本获取的用例描述 ····· 18
3.5 隐私政策上传的用例描述 ····· 18
3.6 APK 文件上传的用例描述 ····· 19
3.7 创建检测任务的用例描述 ····· 19
3.8 查看任务列表的用例描述 ····· 20
3.9 查看任务结果的用例描述 ····· 21
3.10 下载检测报告的用例描述 ····· 21
3.11 用户表 ····· 27
3.12 任务表 ····· 29
3.13 隐私政策表 ····· 29
3.14 APK 信息表 ····· 29
3.15 文本合规性检测结果表 ····· 30
3.16 权限合规性检测结果表 ····· 30
3.17 检测报告表 ····· 30

4.1 隐私政策法规依据 ····· 32
4.2 金融 APP 隐私政策检测指标 ····· 33
4.3 每个主题的标题数统计 ····· 44
4.4 隐私条款的关键词表 ····· 45
4.5 BERT 模型最优参数 ····· 50
4.6 必要个人信息收集范围 ····· 50
4.7 权限名称表 ····· 53
4.8 SimCSE 模型最优参数 ····· 56

5.1 测试环境 ····· 62

5.2 标题分类器测试结果 ····· 63
5.3 隐私条款检测的验证集 ····· 64
5.4 隐私条款检测实验结果 ····· 65
5.5 金融专项检测的验证集 ····· 65
5.6 金融专项检测实验结果 ····· 66
5.7 权限合规性检测的验证集（部分） ····· 66
5.8 权限合规性检测实验结果 ····· 67
5.9 待测 APP 的基本情况 ····· 68
5.10 隐私条款检测的测试结果 ····· 68
5.11 金融专项检测的测试结果 ····· 69
5.12 权限合规性检测结果的测试结果 ····· 69
5.13 用户登录测试用例 ····· 70
5.14 隐私政策文本获取测试用例 ····· 70
5.15 隐私政策上传测试用例 ····· 71
5.16 APK 文件上传测试用例 ····· 71
5.17 用户创建任务测试用例 ····· 72
5.18 查看任务列表测试用例 ····· 72
5.19 查看任务详情测试用例 ····· 73
5.20 响应时间测试 ····· 73

图 目 录

2.1 BERT 逻辑图 ····· 8
2.2 无监督 SimCSE 逻辑图 ····· 10

3.1 系统用例图 ····· 17
3.2 系统整体架构图 ····· 22
3.3 逻辑视图 ····· 24
3.4 开发视图 ····· 25
3.5 进程视图 ····· 26
3.6 物理视图 ····· 27
3.7 ER 图 ····· 28

4.1 数据交互模块时序图 ····· 35
4.2 文本获取界面 ····· 36
4.3 任务创建页面 ····· 37
4.4 文件上传操作 ····· 37
4.5 任务管理页面 ····· 38
4.6 章节标题提取代码 ····· 39
4.7 隐私政策章节标题展示 ····· 40
4.8 语句段展示 ····· 41
4.9 文本段落处理代码 ····· 42
4.10 分词与去停用词代码 ····· 43
4.11 标题分类器构建流程 ····· 44
4.12 隐私条款检测流程图 ····· 46
4.13 BERT 分类流程 ····· 48
4.14 BERT 模型微调代码 ····· 49
4.15 文本合规性检测模块时序图 ····· 52
4.16 APP 权限提取代码 ····· 53
4.17 权限合规性检测流程图 ····· 54

4.18 正反例构建代码 ····· 55
4.19 SimCSE 模型构建代码 ····· 55
4.20 loss 函数构建代码 ····· 56
4.21 文本合规性检测模块时序列图 ····· 57
4.22 结果生成模块时序图 ····· 58
4.23 文本合规性检测任务结果界面 ····· 59
4.24 检测报告 ····· 60
4.25 权限合规性检测任务结果界面 ····· 61

第一章　引言

1.1　研究背景

随着移动互联网的快速发展，移动应用给人们的生活带来了诸多便利，但个人信息被滥用的问题也愈发严重，因此，APP 运行商会通过发布隐私政策的形式来明确企业责任以及使用者的权利。隐私政策（又称“隐私保护协议”、“隐私声明”等）是具有法律效应的合同条款 [1]，随着个人信息保护受到普遍关注，国内外陆续出台了相应法律来规范隐私政策的编写。国际上，欧盟于 2018 年颁布了《通用数据保护条例》(General Data Protection Regulation, GDPR) 来要求开展业务的公司根据条例内容为产品提供全面且透明的隐私政策。与此同时，我国于 2017 年出台了《网络安全法》，对个人信息保护提出了总体要求 [2]，紧接着发布了《信息安全技术个人信息安全技术规范》，提供了关于个人信息保护的标准。2019 年，中央网信办、工信部、公安部、市场监管总局四部门联合发布了《关于开展 APP 违法违规收集使用个人信息专项治理的公告》，重点对 APP 的隐私政策和个人信息收集行为评估，并对违规违法的 APP 运营商进行处罚，情节严重的会被暂停业务甚至吊销营业执照，同年年底，《App 违法违规收集使用个人信息行为认定方法》被颁布。2021 年，《个人信息保护法》与《数据安全法》被通过并开始正式执行。这些法律法规为 APP 运营商有效制定隐私政策提供了必要依据。

相较于普通 APP 而言，金融 APP 受到了国家和金融行业的双重监管 [3, 4]，因此金融 APP 的隐私政策不仅需要符合国家法律法规还要符合金融行业的要求。同时，由于金融 APP 涉及到了大量个人敏感信息，《常见类型移动互联网应用程序必要个人信息范围规定》中分别为金融理财、手机银行以及网络借贷三种类型的金融 APP 规定了必要的个人信息收集范围，即隐私政策里不得随意收集非必要的个人信息。卢智睿 [5] 对 100 款银行 APP 进行了测评，结果表明拥有相对规范隐私政策的银行 APP 仅有 5 款，其余均存在条款内容笼统，过度收集个人信息，过度使用设备权限等问题。由此可见，大量金融 APP 的隐私政策并不合规，未能对用户隐私保护起到引导和约束作用。如果金融 APP 的隐私政策不能满足法律条款和行业要求，则会面临下架、业务停摆甚至巨额罚款的风险。

目前对 APP 的隐私政策合规性检测存在两种方法，第一种是通过隐私政策自动化检测工具进行检测，目前开源的检测项目均存在不少缺陷，首先大多数的项目都只针对英文文本进行检测，中文与英文在自然语言处理上存在较大差

距，无法直接运用到中文文本的检测中；其次大多数项目参考的法规都是基于 GDPR 法规的，无法针对国内的相关法规进行检测；即使是针对了国内法规的检测项目，也缺乏专门针对金融 APP 的隐私政策检测。第二种是使用专业机构提供的检测服务，这些检测服务基本采用了人工审核与自动化检测相结合的方法，存在着价格昂贵，流程长都等种种弊端，不但单次检测的费用高达数千元，检测时间也往往在三天以上。

针对开源工具和商业化检测服务中存在的弊端以及金融 APP 隐私政策领域合规性检测上的空白，本文以我国颁布实施的隐私保护相关政策法规和《个人金融信息保护技术规范》为依据，设计了一份针对金融 APP 的隐私政策评测指标，覆盖了金融 APP 隐私政策中所需要满足的所有隐私条款。同时为了实现快速高效的自动化检测，本系统采用了机器学习算法，能够将隐私政策中的内容精准分类到对应条款下，并判断隐私政策中收集的个人敏感信息是否超出了必要性的范围。同时，本系统还能提取隐私政策中所有声明的权限，与 APP 实际使用的权限进行一致性检测，从而判断 APP 的权限使用是否合规。综上所述，本文的检测系统能够准确高效地完成金融 APP 的隐私政策合规性检测，为 APP 运营商规范制定隐私政策提供有效的建议和指导。

1.2 国内外研究现状

本文研究内容为隐私政策的合规性检测，主要涉及到内容合规与权限合规两个方面。内容合规只关注隐私政策文本内容是否符合相关法律规定；权限合规关注的是隐私政策中声明的敏感权限是否与 APP 中实际使用的敏感权限一致，因此本节将从这两个方面阐述国内外研究现状。

1.2.1 隐私政策文本合规性检测技术

2012 年，卡耐基梅隆大学的 W.Ammar 和 N.Smith 等人首先对隐私政策文本的自动分类 [6, 7] 进行了研究，他们通过自然语言分析和机器学习的方式，证明了隐私政策文本特征的可提取性以及在分类问题上的可行性。

2014 年，美国福罗里达大学的 F.Liu 等人发现了提取关键词对于隐私政策的重要性 [8]。F.Liu 直接对网页链接中的隐私政策进行了处理，从而更加方便快捷地获取隐私政策文本。R.Ramanath [9] 通过众包的方式对隐私政策中的有关片段进行标注来构建数据集。该研究为中文隐私政策数据集的构建提供了一个可参考的方向，即通过在隐私政策文本片段上进行标注来构建数据集。

2015 年，J.Bhatia 和 TDBreaux [10] 对隐私政策的透明度进行了分析，他们通过构建神经网络，提出了一种与上下文无关的方法来自动检测隐私政策中的

文本模糊性。最后该论文指明了文本中模糊词与模糊句子的模糊程度会在一定程度上决定隐私政策的透明度。

2017 年，Mathieu 等人 [11] 围绕半自动注释政策，使用语义分析技术、机器学习和自然语言处理的组合方法，设计了一种名为 PrivOnto 的语义框架对隐私政策文本进行分析，该框架用于与隐私专家合作的数据本体表示注释的隐私政策。2018 年，Harkous [12] 通过众包任务的方式对隐私政策进行标注，设计了名为 Polisis 的隐私政策自动分析框架，可以实现对隐私政策条款的自动分类。

随着 2018 年 GDPR 的出台，关于隐私政策是否符合 GDPR 规范的研究逐渐成为了热门。Contissa [13] 等人按照 GDPR 的要求，采用了基于 SVM、RNN 和 LSTM 三种机器学习方法，对 17 款热门 APP 隐私文本中的语句进行分类，结果表明了基于 SVM 的算法分类效果最好。2021 年，Liu [14] 根据 GDPR 中第 13 章的内容设置了十个检测指标，将分类任务拆分成了基于语句分类和基于规则分析的两个子任务，并人工标注了一个包含 304 篇隐私政策的语料库，然后采用了 SVM，LSTM 和 BERT 三个标准语句分类器，对语料库进行基准测试，最后发现使用 BERT 分类器在召回率和 F1-Socre 上的表现远好于其他两种分类器。

在中文领域，使用自然语言处理和机器学习的方式对隐私政策文本特别是金融类 APP 的隐私政策文本进行自动化分析的研究几乎是空白，反映了在中文领域对金融 APP 隐私政策文本合规性研究的迫切性。

1.2.2 权限合规性检测技术

权限合规性是指 APP 中实际使用的权限要与隐私政策中声明的权限一致，如果 APP 中使用的权限未在隐私政策中声明，则隐私政策的权限描述不合规。

Pandita 等人 [15] 通过自然语言处理技术对 APP 的描述信息进行了分析，并提取了相应的权限来分析权限的描述和使用是否一致。研究的主要思路是从 APP 的描述中提取与权限相关的部分，并将它们映射到具体权限名称，然后，将映射得到的权限名称与 APP 实际使用的权限进行比较。

Slavin 等人 [16] 设计了 9 种类型的隐私政策敏感信息，并通过人工的方式对 115 个隐私政策中声明的隐私信息进行了标注，一共标注了 2831 条隐私信息做为数据集。然后使用静态分析的方法从 APP 提取了敏感权限，之后通过机器学习的方法，对隐私政策中的隐私信息按照前面设计的九种类型进行分类，最后完成了 APP 中敏感权限的 API 方法与隐私政策信息之间的映射，从而判断隐私政策中声明的权限是否一致。

Yu 等 [17] 人通过使用 Standford Parser 提取了隐私条例中声明的隐私信息，然后使用静态分析获取应用的敏感行为，最后将两者进行对比分析，完成是否

一致的判断。

目前已有的权限合规性检测工作主要应用在英文 APP 与其隐私政策上，但大量中文 APP 中也存在着实际使用权限与 APP 政策中声明权限不符的情况，本文将在中文领域，实现金融 APP 的权限合规性检测。

1.3 本文主要工作

一份合格的金融 APP 隐私政策一方面要在内容描述上合规，即包含法律法规和金融行业的基本要求。另一方面要在权限使用上合规，即 APP 中使用的所有权限均要在隐私政策中进行声明。

考虑上述问题，本文设计的合规性检测系统对金融 APP 提供了两项检测任务。第一个任务是文本合规性检测，它从政策政策文本的角度进行考虑，检验了金融 APP 声明的隐私政策是不是符合国家和行业规范，同时针对金融 APP 所属类型检测了收集的个人信息是否符合必要性范围。第二个任务是检验金融 APP 实际使用的敏感设备权限是否都在隐私政策中进行了申明。

本文的主要工作分为以下几点：

（1）设计了金融类 APP 隐私政策检测指标。本文根据我国颁布实施的相关法律法规和金融领域具体的行业要求，设计一套包含多个隐私条款的金融 APP 隐私政策检测指标。

（2）构建金融类 APP 隐私数据政策数据集。首先通过华为应用市场，爬取了所有金融类 APP 隐私政策链接，并按下载量从高到底进行排序。之后通过 Python 的 newspaper 库将网页内容转化成了 txt 文本，并去掉了所有无效文本。最后按照下载量，从金融理财、手机银行类以及网络借贷三个类型中各选取了 50 份隐私政策文本，共计 150 份隐私政策做为本文中各种实验的数据集。

（3）设计了金融 APP 文本合规性检测的方法。首先对隐私政策的章节标题进行分类，确定它们的主题。然后将相同主题下的内容进行汇总，每个主题根据自身类型，选择使用关键词匹配或 BERT 模型分类的方法对内容进行分类，并根据所有主题分类结果的完整性来判断隐私政策是否合规。之后选择了分类类别为“收集的个人信息类型“的内容，并根据金融 APP 的类型，使用对应的 BERT 分类器对个人信息内容进行分类，判断它们是否为该类金融 APP 必要收集的个人信息，如果收集了非必要的个人信息，则不合规。

（4）设计了金融 APP 权限合规性检测的方法。首先通过 Python 的 Androguard 库对 APP 进行静态分析，获取 APP 中实际使用的敏感权限，并将其转化成标准描述语句，得到 APP 的使用权限集。然后根据前面分类的结果，获取类别为“申

请的设备权限”的内容，得到隐私政策的声明权限集。然后使用 SimCSE 模型对两个权限集进行语义相似度比较，如果使用权限集中的语句都能在声明权限集中找到相似的语句，则证明权限合规，反之则权限不合规。

（5）设计并实现了一套金融 APP 的隐私政策合规性检测系统。该系统提供了文本合规性检测和权限合规性检测两种任务。文本合规性检测任务需要用户上传金融 APP 的隐私政策文本并选择具体的金融类型，权限合规性检测任务需要用户上传金融 APP 的隐私政策和 APK 文件。系统还提供了任务管理，任务结果查看和检测报告下载等功能，帮助用户快速了解任务的完成情况和检测结果。

1.4 本文组织结构

本文总共分为六个章节，各章节研究内容如下：

第一章，引言。本章主要介绍了本文的课题背景与研究意义，然后从隐私政策文本合规性检测技术与权限合规性检测技术这两个方面梳理了研究现状，最后对本文的研究工作进行了总结。

第二章，相关技术概述。本章主要介绍了项目实现中使用到的自然语言处理技术、TF-IDF 算法、BERT 模型和 SimCSE 模型，介绍了国外的一些隐私政策语料库以及系统 Web 开发所用到的框架。

第三章，系统的需求分析与概要设计。首先对系统进行了一个整体的概述，接着对系统的功能需求以及非功能需求进行了分析，然后结合 4+1 视图，介绍了系统的总体设计，最后讲述了系统在持久化存储和数据库上的设计。

第四章，系统的详细设计与实现。基于上一章的需求分析与技术设计，对系统的数据交互模块、文本处理模块、文本合规性检测模块、权限合规性检测模块以及结果生成模块的具体实现进行了详细说明，并通过界面截图演示了系统的操作流程。

第五章，系统测试与案例分析。介绍了本文所进行的各项测试，主要包括各种算法的性能测试、系统的实际案例分析，系统的功能性测试和非功能性测试四个方面进行测试，以保证系统的可用性和实用性。

第六章，总结与展望。该章节总结了本文研究及开发工作的成果，并根据本文在研究和系统设计上的不足之处，给出了未来工作的改进方向。

第二章　相关技术概述

本章介绍了实现面向金融 APP 的隐私政策合规性检测系统时所用到的一些技术，包括中文分词技术、TF-IDF 算法，SimCSE 模型、BERT 模型和静态分析技术，并对国外的隐私政策语料库进行了简单的介绍。这些技术将对后面的隐私政策文本合规性检测与权限合规性检测起到重要作用，最后还介绍了本系统在 Web 开发中所要用到的 Django 框架。

2.1　中文分词技术

本文的金融 APP 隐私政策合规性检测系统涉及到了大量对文本的操作，需要运用自然语言处理 [18] 中的中文分词技术。

中文分词 [19] 是在进行中文文本信息处理时的关键技术及难点。中文与英文分词的一个重要的区别就是，英文可以直接通过单词直接的间隔完成了天然分词，而中文以字为单位的，每个词语之间没有任何分隔符，分割出语义正确的词语需要的工作更加复杂。因此对于中文 NLP 而言，分词是一个很基础的任务，是几乎所有其他任务的上游任务。现有的中文分词方法可以可分为三大类：基于字符串匹配的分词方法、基于理解的分词方法和基于统计的分词方法 [19]。其中基于统计的中文分词方法是目前最主流的方法，不少基于此类方法的中文分词器受到了广泛认可，如结巴分词器 (Jieba) 和清华中文词法分词器（THU Lexical Analyzerfor Chinese）。

本文使用的是 Jieba 分词器做为分词工具，Jieba 是一个通过 Python 实现的中文分词器，其进行分词的思路主要为：（1）加载词典文件，根据词典对句子进行扫描，构建所有分词情况的有向无环图。（2）使用动态规范算法，寻找概率最大的路径（3）对词典中没有的词语，使用 HMM 模型的 Viterbi 算法进行分词处理。（4）返回分词结果。

Jieba 分词器包含了三种模式，分别是精确模式、全模式和搜索引擎模式。精确模式能够精确切分句子，不含冗余词语；全模式会将句子中的所有词语切分出来，可能会有冗余词语；搜索引擎模式是在精确模式基础上对长词语再次切分。

2.2　TF-IDF 算法

TF-IDF [20] 是一种用于信息检索与数据挖掘的常用加权技术，文本处理中常用 TF-IDF 的方法进行关键词提取。TF-IDF 由” 词频”（Term Frequency，TF）

和”逆文档频率”（Inverse Document Frequency，IDF）两个部分组成。

“词频”是统计某个词在文本中出现的次数。但词频相同的词语并不能代表它们的重要性一样，某些词只是常见词而已。因此需要一个参数来衡量这个词语是常见词还是对该文本的关键词。如果某个词语在其他文本中很少见，但在这个文本中多次出现，那么它可能就反映了整个文本的特性，是该文本的关键词。因此，需要在词频的基础上对每个词分配一个权重，越常见的词权重越小，这个权重就叫做”逆文档频率”。当有 TF 和 IDF 后，就可以得到一个词的 TF-IDF 的值。TF-IDF 越大的词重要性就会越高。

TF-IDF 的计算一般分为以下几个步骤:

（1）获取 TF。

$$tf_{ij} = \frac{n_{ij}}{\sum_k n_{kj}} \tag{2.1}$$

其中 n_{ij} 是该词在文本中出现的次数，分母是所有词汇出现的次数总和。

（2）获取 IDF。

$$idf_i = \log \frac{|D|}{\left|\left\{j : t_i \in d_j\right\}\right|} \tag{2.2}$$

其中，$|D|$ 表示文本总数。分母表示包含词语 t_i 的文本数目。如果该词语不在语料库中，会导致分母为零，因此一般使用以下公式：

$$idf_i = \log \frac{|D|}{1 + \left|\left\{j : t_i \in d_j\right\}\right|} \tag{2.3}$$

(3) 计算 TF-IDF

$$TF - IDF = TF * IDF \tag{2.4}$$

2.3 BERT 模型

BERT 是 Google 在 2018 年公开的是一种预训练的语言表征模型。根据论文 [21] 中的表明，BERT 在 11 个自然语言处理任务中都刷新了记录，使得自然语言处理任务有了新的发展方向。

BERT 模型的逻辑如图 2.1所示，BERT 模型在静态词向量的基础上，使用双向的 Transformer 编码器 [22] 进行上下文学习，从而得到文本的向量化表示。Transformer 是一个使用自注意力的 Seq2Seq 模型 [23, 24]，Seq2Sqe 模型通常由编码器和解码器两个部分组成，编码器的作用是将长度可变的输入序列转化为

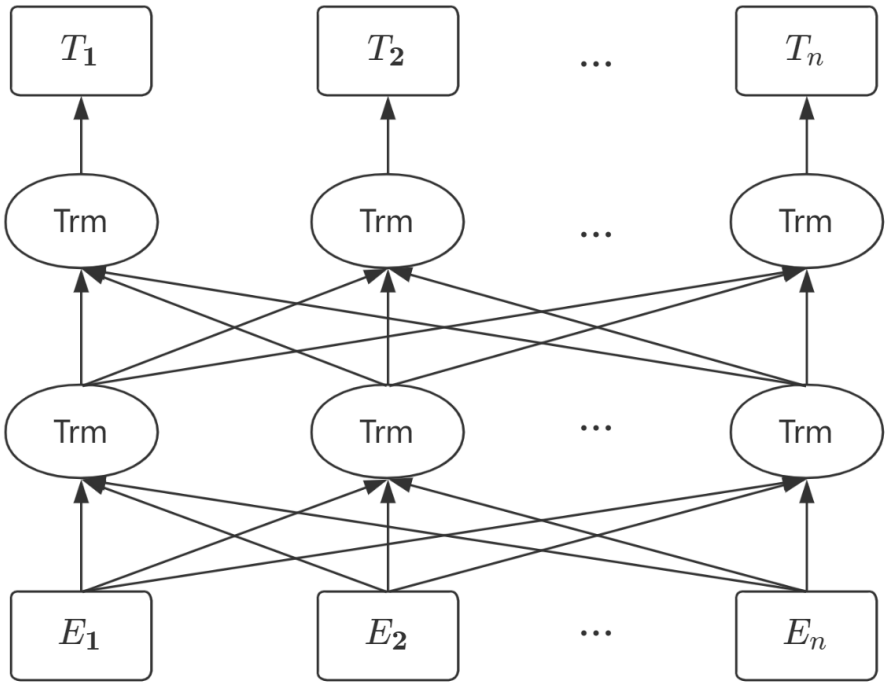

图 2.1: BERT 逻辑图

长度固定的向量，解码器将长度固定的向量解码成长度可变的输出序列。Transformer 中编码器的主要功能是对含有位置信息的句向量进行特征提取，并通过自注意力机制学习前后文信息；解码器的主要功能是解析编码器输出的向量，使其能够适应于下游任务。两部分都包含了多头注意力机制，能够从不同的维度考虑词的信息，提升模型的泛化能力；同时，两部分都包含了全连接层，能够综合考虑全局特征，Transform 还引入了归一化的功能，可以加快模型训练效率。

BERT 模型使用了 Transformer 模型中的编码器，通过多层编码器来进行特征提取。BERT 的输入采用了 Wordpeice 向量、句子向量和位置向量相加，且句子向量和位置向量都是可训练的；采用 Wordpeice 向量可以把词本身和语态时态等相关后缀分开，从而减少了词表的大小。

为了增强模型的语义表示能力，BERT 提出了两种预训练任务：

(1) 遮蔽语言模型 (masked language modeling)：对一个句子中的词语进行随机掩盖，然后用剩余词对掩盖词进行预测。BERT 中随机掩盖了 15% 的词，掩盖方式采用如下策略：80% 用 [MASK] 代替，10% 用随机词代替，10% 保持不变。

(2) 下一句预测 (next sentence prediction))：通过一个二分类方法进行下一句预测，来判断两个句子是否属于上下文关系。

BERT 的使用主要是两种方法：微调和特征提取。微调主要关注下游任务，直接使用社区中的预训练模型，然后针对具体任务调整参数进行训练即可。基于特征提取的方法是把序列输入 BERT，取 BERT 最后一层或多层 Transformer 的输出做为序列对应的特征向量，然后将得到的特征向量运用在其他的任务中。

2.4 SimCSE 模型

学习自然语言的通用语义表示一直是 NLP 的基础任务之一，SimCSE [25] 利用了对比学习的思想，可以从无监督的数据集中生成表征句子语义的向量，大幅提升了现有无监督语义匹配记录。本节主要讲述对比学习的目标以及 SimCSE 如何运用对比学习思想进行无监督学习。

对比学习的目标是在拉近正样本的同时推开负样本，关键在于如何构造 (x_i, x_i^+) 正样本对。因此，对比学习最早起源于计算机视觉中，因为对于图像而言，语义一致的正样本 x_i^+ 十分容易构造，如通过裁剪、翻转等方式。而自然语言的结构高度离散，难以构建语义一致的正例 x_i^+，传统构造 x_i^+ 的方法大多是一些数据增强的方式，如同义词替换、无关词删除等，但这些操作容易引入与负样本相似分布的”噪声”，这导致模型难以从正负对中学习到正确的语义信息。

对比学习希望学习一个高质量的语义表示空间，一般使用 alignment 和 uniformity 这两个指标对语义表示空间的质量进行评价。

alignment 计算 x_i 和 x_i^+ 的距离：

$$l_{\text{align}} = E_{(xx^+)\sim p_{\text{pos}}} \left\| f(x) - f(x^+) \right\|^2 \tag{2.5}$$

其中，p_{pos} 为正样本对的分布，$f(\cdot)$ 为编码器。

uniformity 计算语义向量整体分布的均匀程度:

$$l_{\text{uniform}} = \log E_{(xy)\sim p_{\text{data}}} \exp\left(-2\|f(x) - f(y)\|^2\right) \tag{2.6}$$

其中，p_{data} 为所有数据的分布。这两个指标越低越好，一方面是希望正样本要挨得足够近，另一方面希望语义向量要尽可能地均匀分布在超球面上，因为分布越均匀信息熵就越高，保留的信息就越多。对比学习的目标“拉近正样本，推开负样本”实际上就是在优化这两个指标。

直接用 BERT 生成的句向量做无监督语义相似度计算效果会很差，任意两个句子，直接通过 BERT 获取句向量的相似度都很高，原因在于 BERT 产生的向量存在各向异性 [26]，进而导致了分布不均匀问题。BERT-flow [27] 通过 normalizingflow 将各向异性的 BERT 向量转换到一个标准的高斯分布空间，BERT-whitening [28] 在 BERT-flow 的基础上，对向量分布做了 PCA 降维消除冗余信息，

只保留特征值的前 n 个维度，但标准化流的表达能力差，同时白化操作无法解决非线性的问题。

SimCSE 借助对比学习来达到规整表示空间的效果，但这又涉及到了正样本的构建这一难题，SimCSE 创新性地完成了无监督条件下的正样本构建。图 2.2为 SimCSE 的正样本构建方法。SimCSE 通过随机采样 dropout mask 来生成正样本，

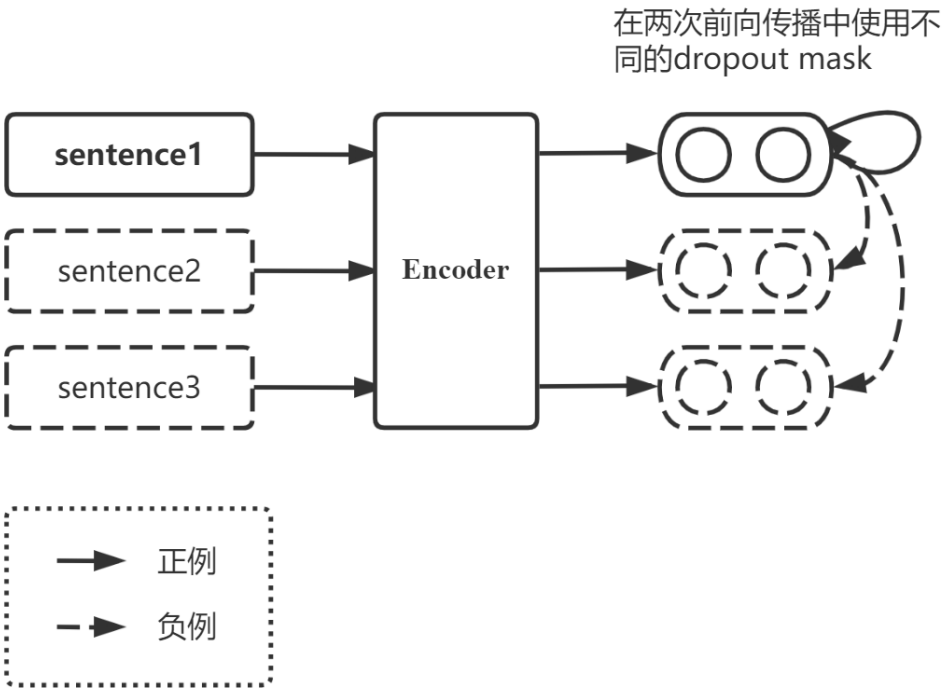

图 2.2: 无监督 SimCSE 逻辑图

由于 BERT 的编码器是一个随机的 dropout mask 编码器，因此输入一个句子两次会得到两个不同的向量，产生的这两个向量作为训练中进行对比的正样本，同一个 batch 中其他句子产生的向量则作为负样本，以此来构造一个对比学习的无监督训练，则模型的训练目标为

$$\ell_i = -\log \frac{e^{\operatorname{sim}\left(\mathbf{h}_i^{z_i}, \mathbf{h}_i^{z_i'}\right)/\tau}}{\sum_{j=1}^{N} e^{\operatorname{sim}\left(\mathbf{h}_i^{z_i}, \mathbf{h}_j^{z_j'}\right)/\tau}} \tag{2.7}$$

其中 sim 是相似度计算，τ 是一个用于平滑调节的温度因子，整个公式的意义是让正样本之间的距离最小，同时让正样本与其他负样本之间的距离最大。

这种通过改变 dropout mask 生成正样本的方法可以看作是数据增强的最小形式，因为是同一个语句通过不同的 dropdout 产生的，既可以保证在向量上的

差异，同时又保证了两者在语义上的完全一致。

2.5 静态分析

静态分析是指在不需要软件运行的情况下对软件行为进行检测的方法 [29]。APP 的静态分析的主要对象是静态代码，因此需要对 APK 文件进行操作。通常的方法是需要使用逆向分析技术对 APK 进行反编译，获取反编译代码或者 Android 源码。常用的反编译工具有 APKTool、dex2jar 和 AndroGurad 等，反编译后主要关注 Smail 文件和 AndroidManifest.xml 文件。Smail 文件是 APK 反编译成 Java 源码的中间代码；AndroidManifest.xml 用于管理应用的权限信息，当 APP 运行中需要使用权限时，需要在 AndroidManifest.xml 中进行，来查询验证系统是否拥有相应权限。

APP 的静态分析不需要运行 APP 或模拟 APP 的行为，只需要对反编译得到的源码和相关文件进行操作即可，执行效率更高。本文将使用 Androguard 对金融 APP 进行静态分析。

2.6 隐私政策语料库

对隐私政策进行自动化分析的一个重要前提就是构建语料库或数据集，本节将对国外一些著名的隐私政策语料库和数据集进行了介绍。

OPP-115 语料库 [30] 是 115 个网站隐私政策的集合，每项隐私政策都由三名法学研究生进行阅读和注释，该数据集中通过注释定义了十种类型的数据实践，目前该数据集被广泛用于学术研究中。APP-350 语料库 [31] 是由 350 个 APP 的隐私政策组成的语料库。该语料库建立的目的是为了研究 APP 个人信息的收集与使用是否符合其隐私政策中所描述的收集使用规则，该项目通过动静态分析来完成隐私政策的合规性分析。MAPS 隐私政策数据集 [31] 由 441626 隐私政策的 URL 组成。这些隐私权政策是在 2018 年 4 月至 5 月由移动应用程序隐私系统（MAPS）识别并经过 Google Play 商店应用程序分析而构建的。PrivaSeer 语料库 [32] 是一个可用于分析、理解和简化隐私政策的大规模隐私政策语料。它的特点是规模明显大于之前可用的任何语料库，其中包含了一百万个英语网站的隐私政策。

目前中文领域缺乏业界认可的隐私政策语料库，本文将自己收集金融 APP 的隐私政策文本，来构建用于机器学习的语料库。

2.7 Django

Django 是一种优秀的 Python 网络框架，能够开发安全且易维护的网站。Django 是一个 MTV 的设计模式。MTV 是 Model、Template 和 View 的首字母缩写，分别代表着模型、模版和视图，它们各自的职责则包括:

Models: 负责定义 Python 模型类对象，提供数据库管理和查询记录的功能。

View: 负责实现业务逻辑，接收 HTTP 请求并返回响应。

Templates: 负责页面展示、业务逻辑写入模板等操作。

Django 中的 URL 映射器可以将 HTTP 请求定位到相应的视图。当它接受一个请求时，会将这个请求的 url 在映射表中进行匹配，找到对应的视图，然后执行视图函数。

Django 上手简单，使用方便，拥有活跃的社区和丰富的文档，适合 Web 项目的快速开发。

2.8 本章小结

本章主要就本系统相关的技术进行了概述说明。首先对中文分词技术以及本文使用到的 Jieba 分词器进行了介绍。然后了介绍了 TF-IDF 算法，该算法主要用来统计词频，用于章节标题的分类。之后介绍了在文本合规性检测和权限合规性检测中需要用到的 BERT 模型和 SimCSE 模型。最后介绍了静态分析和一些常见的隐私政策语料库，并对系统开发所用到的 Django 框架进行了简单说明。

第三章　系统的需求分析与概要设计

本章对系统的需求分析与概要设计进行说明。首先对系统的总体需求以及需要实现的功能进行了概述，接着分析了系统的功能需求和非功能需求，并描述了系统用例。然后根据需求分析的结果，介绍了系统的总体设计以及 4+1 视图。最后给出了系统各个实体之间关系的 ER 图，并根据 ER 图设计了数据库。

3.1　系统整体概述

随着个人信息安全的重视，国内关于隐私保护的法律条款陆续出台，金融 APP 隐私政策的制定变得越来越严格。一份不合格的隐私政策不仅可能导致金融 APP 被应用商场下架，甚至会让公司面临巨额罚款，而人工审核的方式不但需要专业人员，而且消耗的时间很长。为了解决这一问题，本文设计了一个面向金融 APP 的隐私政策合规性检测系统，能够帮助金融类 APP 快速发现隐私政策中的问题，完成隐私政策的自查和修改。

本文阐述的面向金融 APP 的隐私政策合规性检测系统，提供了文本合规性检测和权限合规性检测两项功能。文本合规性检测任务分为隐私条款检测和金融专项检测，隐私条款检测以国家法律法规和金融行业的相关规定做为依据，对金融类 APP 的隐私政策的合规性进行一个全面且彻底地检测，并对其中的违规或缺失部分进行记录；金融专项检测会对金融理财类、手机银行类以及网络借贷类这三种类型的金融 APP 进行检测，判断隐私政策中对个人信息的收集是否超过了必要范围。权限合规性检测任务提取政策中的权限语句，并与 APK 文件中实际申请的权限进行对比，如果一致则合规，否则不合规。两个检测模块将都会展示检测结果，其中文本合规性检测由于涉及的内容较多，提供了检测报告供用户下载。

根据需求描述和目标，从功能上看，金融 app 的隐私政策合规性检测系统可以分为数据交互模块、文本处理模块、文本合规性检测模块、权限合规性检测模块和结果生成模块。数据交互模块给用户提供文件上传，任务创建和任务查看等功能。文本处理模块负责给其他模块提供一些文本处理方面所需要的功能，如章节标题提取，段落提取，中文分词以及去停用词等。文本合规性检测模块根据相关法规对金融 APP 的隐私政策进行隐私条款检测和金融专项检测。权限合规性检测模块对隐私政策中声明的权限与 APP 的实际操作中获取的权限进行对比。结果生成模块根据检测结果生成检测结果界面，并为文本合规性检测提供

了检测报告，用户可直接下载。

从具体业务交互流程上看，用户在任务创建界面先要输入任务名称，任务描述，APP 名称。然后选择任务类型，任务类型分为文本合规性检测和权限合规性检测。对于文本合规性检测任务，用户需要选择待检测金融 APP 的类型并上传金融 APP 的隐私政策隐私文本，用户可以通过文本获取界面，直接输入隐私政策的网页链接来获得隐私政策的文本；对于权限合规性检测任务，用户需要上传金融 APP 的隐私政策文本和 APK 文件。所有上传的文件都将存储在服务器中。上传成功文件后，点击创建任务即可开始执行检测。

在文本合规性检测任务中，首先进行隐私条款检验，系统先读取用户上传的隐私政策，然后调用文本处理模块提取章节标题，接着使用训练好的 SVM 模型，对章节标题进行主题分类，将同一主题的内容汇集在一起。不同的主题会根据自身类型，采用基于 BERT 的分类器或通过关键词匹配的方式对内容进行分类。最后结合隐私政策检测指标来判断隐私政策是否符合合规，并对缺少的隐私条款进行记录。完成隐私条款检测后进行金融专项检测，首先获取分类结果为“收集的个人信息类型”的内容，然后根据 APP 具体的金融类型，调用对应的 BERT 分类器对这些内容进行一个二分类，判断这些个人信息是否为必要收集的个人信息。最后将这些检测的结果信息保存到数据库中。

在权限合规性检测中，系统会先从隐私政策文本中获取分类结果为“申请的设备权限”的内容，做为声明权限集。然后对 APK 文件进行静态分析，获取实际使用的敏感权限，通过参照权限映射表将它们转化为对应的标准描述做为使用权限集，之后使用基于 SimCSE 的语义相似度检测，判断使用权限集中的语句是否能够在政策权限集中找到相似的语句，如果均能找到则合规，否则不合规。检测结果会被保存到数据库中。

结果生成模块会根据数据库中的结果信息，为相应的任务生成结果展示界面，对于文本合规性检测任务还提供了检测报告下载功能。

3.2 系统需求分析

需求分析是软件开发的一个重要环节，本节将分别介绍系统的功能性需求分析、非功能性需求分析以及用例图。

3.2.1 功能性需求分析

系统的功能性需求如表 3.1所示，主要包括了用户登录、政策文本获取、隐私政策文件上传、APK 文件上传、创建任务、查看任务列表、查看任务结果以及下载检测报告。

表 3.1: 功能性需求列表

需求编号	需求名称	需求内容
R1	用户登录	用户在登录界面通过已授权的账号和密码进行登录，登录成功后可随时登出。
R2	政策文本获取	用户可以通过输入隐私政策的网页链接来获取文本。
R3	隐私政策上传	在创建任务时，用户能够选择并上传 APP 的隐私政策文件。
R4	APK 文件上传	在创建任务时，用户能够选择并上传 APP 的 APK 文件。
R5	创建任务	用户可以选择任务类型并创建任务，任务类型分为两种：文本合规性检测和权限合规性检测。
R6	查看任务列表	用户能够查看自己创建过的任务，所有任务通过列表的形式呈现，列表中每一行都是一个任务记录。用户可以通过任务列表了解各项任务的基本情况，包括任务名称、APP 名称、任务类型和任务状态等信息，并提供了搜索和分页功能。
R7	查看任务结果	用户通过点击任务列表操作栏的“查看”按钮，可以查看任务结果，包括任务相关信息，任务执行时间，任务完成时间以及任务检测结果的概述。
R8	下载检测报告	对于文本合规性检测任务，用户可以在任务结果界面里通过点击“下载”按钮来下载检测报告。

3.2.2 非功能性需求分析

系统的非功能性需求如表 3.2所示，主要从可靠性、可用性、可扩展性、响应时间以及兼容性这几个方面来制定。系统的非功能性需求保障了软件的质量并加强了用户体验。

3.2.3 系统用例设计

根据上文中对系统功能性需求的分析，可以得到如图 3.1所示的系统用例图，主要包含：用户登录、政策文本获取、隐私政策上传、APK 文件上传、创建任务、查看任务列表、查看任务结果、下载检测报告这 9 个系统用例。

3.2.4 系统用例描述

下面对各个功能模块进行分析，对每个用例进行详细描述，包括用例名称、参与者、用例描述、触发条件、前置条件、后置条件、正常流程以及异常流程。

表 3.3给出了用户登录的用例描述。用户进入登录页面后，需要使用被授权的账号，在输入框中输入账号和密码并点击“登录”按钮跳转至程序主页面，如果用户账号没有被授权或用户密码错误，系统会进行相应的提示。

表 3.4给出了用户获取隐私政策文本的用例描述。用户可以直接通过输入隐私政策网址的方式获取隐私政策文本。点击左侧导航栏的“文本获取”进入网

表 3.2: 非功能性需求列表

需求名称	需求内容
可靠性	系统涉及到隐私政策网址的输入以及特定类型文件的上传，因此存在网址链接无效，文件类型异常等问题，系统应妥善处理这些问题
易用性	系统界面简洁友好，各种操作都有及时的反馈，如用户在文件上传时会显示进度条，用户创建任务成功后会提示任务创建完成
可扩展性	系统的前后端应该进行分离，方便界面的优化；系统的各个功能模块之间应高度解耦，方便后续功能的扩展与改进
响应时间	为了保证用户操作的流畅性，需要规定页面响应时间。如文件上传成功反馈延时不高于 1 秒，创建成功后跳转的响应时间不超过 1.5 秒，结果查看的响应时间不超过 2 秒等
兼容性	本系统为 Web 应用，应兼容各主流浏览器

表 3.3: 用户登录的用例描述

ID	UC1
名称	用户登录
参与者	用户
用例说明	实现用户的登录
触发条件	用户进入登录界面
前置条件	用户获得授权账号
后置条件	无
正常流程	1. 用户输入用户和密码 2. 用户点击“登录”按钮
异常流程	1a. 如果用户名不存在，提示用户不存在 1b. 如果用户名存在但密码错误，提示密码错误

页链接转文本的界面，在输入框内输入隐私政策的网址，然后点击“获取文本”按钮，网页内容会自动转换为 txt 格式并下载到本地。

表 3.5给出了用户上传隐私政策文件的用例描述。用户进入创建任务界面后，在隐私政策上传一栏中，点击“选取文件”按钮选择要上传的隐私政策文本，然后点击“上传文件”进行上传，当用户上传文件数大于 1 或选取非 txt 类型的文件上传时，系统会进行提示，文件上传过程中会有进度条进行显示。

表 3.6给出了用户上传 APK 文件的用例描述。用户进入创建任务界面后，在 APK 上传一栏中，点击“选取文件”按钮选择要上传的 APK 文件，然后点击“上传文件”按钮进行上传，当用户上传文件数大于 1 或选取非 APK 类型的文件上传时，系统会进行提示。APK 在上传后会校验 md5 码，如果数据库中已经记录了相同的 APK 文件，会提示用户。

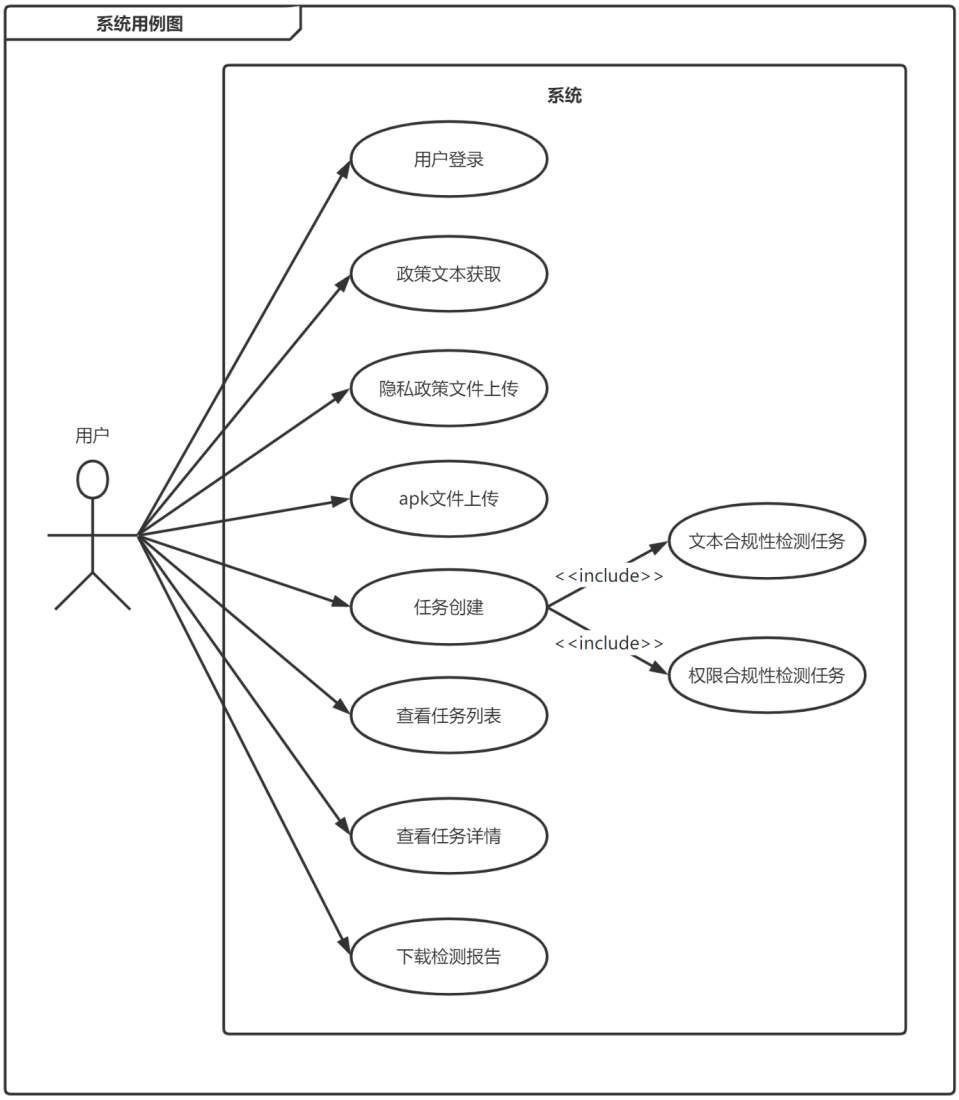

图 3.1: 系统用例图

表 3.7给出了用户创建任务的用例描述。用户登录进入主页面后，点击左边导航栏的创建任务按钮，即可进入任务创建界面。用户需要依次填入任务名称，任务描述和 APP 名称，其中任务名称和 APP 名称是必填，接着用户需要选择任务类型，共有两种类型的任务可以选择，分别是文本合规性检测任务和权限合规性检测任务。对于文本合规性检测任务，用户需要选择待测金融 APP 的具体类型，并上传隐私政策文本；对于权限合规性检测任务，用户需要上传隐私政策文件和 APK 文件。上传完对应的文件后，用户点击“创建任务”按钮执行检测任务。

表 3.8给出了查看任务列表的用例描述。用户创建任务后，可以通过左边的导航栏中的“管理任务”来查看所有已创建的任务。为了方便用户查找，系统提

表 3.4: 政策文本获取的用例描述

ID	UC2
名称	隐私政策文本获取
参与者	用户
用例说明	用户通过隐私政策网址获取隐私政策的文本
触发条件	用户进入“文本获取界面”
前置条件	无
后置条件	无
正常流程	1. 用户输入隐私政策的网址 2. 用户点击“获取文本”按钮 3. 隐私政策以 txt 格式被下载到本地
异常流程	2a. 如果网址格式错误或无法打开，系统提示“网页链接无效”

表 3.5: 隐私政策上传的用例描述

ID	UC3
名称	隐私政策文件上传
参与者	用户
用例说明	实现用户隐私政策文件的上传
触发条件	用户进入任务创建界面
前置条件	无
后置条件	隐私政策文本上传至服务器，数据库中新增记录
正常流程	1. 用户在侧边导航栏点击“创建任务”按钮进入创建任务界面 2. 用户在“隐私文本上传”一栏中，点击“选取文件”按钮 3. 用户选取文本类型的文件，文件出现在界面中等待上传 4. 用户点击“上传文件”按钮，界面显示文本文件上传进度，进度达到 100% 后，系统显示“隐私政策文本上传成功”
异常流程	2a. 如果已经用户已经选取了文件，系统会提示用户“只能选取一个文件上传” 3a. 如果文件类型不是 txt 类型，系统会提示用户“文件类型必须为 txt 类型”

供了任务筛选功能，用户可以通过任务类型筛选器，任务状态筛选器和任务名称搜索框来查找目标任务。任务列表中的每一行都是一条任务记录，显示了该任务的基本信息，包括任务名称、任务描述、任务类型、金融 APP 名称、创建时间以及运行状态。运行状态共分为三种，分别是“运行中”，“已完成”和“失败”。对于运行状态为“已完成”的任务，用户都可以点击操作栏下的“查看”按钮来查看任务的结果。对于不需要的任务，用户可以点击“删除”按钮对任务进行删除。

表 3.6: APK 文件上传的用例描述

ID	UC4
名称	APK 文件上传
参与者	用户
用例说明	实现用户隐私政策文件的上传
触发条件	用户进入任务创建界面
前置条件	无
后置条件	APK 文件上传至服务器，数据库中新增记录
正常流程	1. 用户在侧边导航栏点击“创建任务”按钮进入任务创建界面 2. 用户在“APK 文件上传”一栏中，点击“选取文件”按钮 3. 用户选取 APK 类型的文件，文件出现在界面中等待上传 4. 用户点击“上传文件”按钮，界面显示 APK 文件的上传进度，进度达到 100% 后，系统显示“APK 文件上传成功”
异常流程	2a. 如果已经用户已经选取了文件，系统会提示用户“只能选取一个文件上传” 3a. 如果文件类型不是 APK 类型，系统会提示用户“文件类型必须为 APK 类型” 4a. 如果上传 APK 文件的 md5 码在数据库中有记录，系统会提示用户“APK 文件已存在，无需上传”

表 3.7: 创建检测任务的用例描述

ID	UC5
名称	创建任务
参与者	用户
用例说明	用户创建检测任务
触发条件	用户点击“任务创建”
前置条件	用户完成登录
后置条件	数据新增任务记录，执行任务流程
正常流程	1. 用户在侧边导航栏点击“创建任务”按钮 2. 用户输入任务名称 3. 用户输入任务描述 4. 用户输入 APP 名称 5. 用户在任务类型栏中选择要创建的检测任务 6. 根据选择的检测任务上传需要的文件 7. 用户点击”创建任务”按钮，系统显示“任务创建成功”
异常流程	7a. 如果用户未填写任务名称，系统会提示“请填写任务名称” 7b. 如果用户未填写 APP 名称，系统会提示“请输入 APP 名称” 7c. 如果用户未选择任务类型，系统会提示“请选择任务类型” 7d. 如果用户没有上传相应的文件，系统会提示”请上传文件”

表 3.8: 查看任务列表的用例描述

ID	UC6
名称	查看任务列表
参与者	用户
用例说明	1. 用户可以获取自己创建的全部任务的任务列表 2. 用户可以通过筛选器和搜索框来搜索目标任务 3. 用户可以点击“查看”按钮查看某个任务的任务结果 4. 用户可以点击“删除”按钮来删除某个任务
触发条件	用户点击“任务管理”
前置条件	用户完成登录
后置条件	无
正常流程	1. 用户在侧边导航栏点击“任务管理”按钮 2. 用户通过翻页的方式浏览任务列表 3. 用户使用筛选器或搜索找到对应的任务 4. 用户点击某项任务操作栏下的“查看”按钮，查看任务的详细情况 5. 用户点击某项任务操作栏下的“删除”按钮，弹窗显示是否删除，点击确定后删除任务并刷新列表
异常流程	2a. 如果通过搜索框搜索的任务不存在，则会返回一个空列表。 4a. 如果任务处于“运行中”或“未完成”状态，则无法查看检测结果。

表 3.9给出了查看任务检测结果的用例描述。用户在任务列表中点击某个任务“查看”按钮，即可进入该任务的任务结果界面。任务界面会显示任务的完成时间和运行时间，对于文本合规性检测任务，会展示了隐私条款检测和金融专项检查的总结，并提供一个检测报告的下载按钮。对于权限合规性检测任务，会展示隐私政策中声明的权限和 APP 使用的权限，并对不一致的权限进行显示。

表 3.10给出了下载检测报告的用例描述。用户在文本合规性检测任务的结果界面中点击“下载”按钮，即可自动下载该任务的检测报告，报告格式为 PDF。报告对隐私政策的违规行为进行了总结，并用表格展示了所有主题内容和隐私条款内容的检测情况。

3.3 系统整体设计

本节将对本文面向金融 APP 的隐私政策合规性检测系统的总体设计进行详细描述，该详细设计建立在上一节对系统的需求分析以及用例分析的结果之上。本节首先展示了系统的整体整体架构图，基于该图介绍了系统各个模块的功能以及它们之间的联系。然后，本节通过 4+1 视图对本文系统的总体设计进行多角度的描述。

表 3.9: 查看任务结果的用例描述

ID	UC7
名称	查看任务结果
参与者	用户
用例说明	用户可以在任务列表里查看“已完成”状态的任务结果
触发条件	用户点击任务列表中某个任务的“查看”按钮
前置条件	用户查看任务列表
后置条件	无
正常流程	1. 用户进入任务结果界面 2. 界面对任务的执行情况以及检测结果进行了显示

表 3.10: 下载检测报告的用例描述

ID	UC8
名称	下载检测报告
参与者	用户
用例说明	用户查看任务的检测结果
触发条件	用户在文本合规性检测类型任务的界面中点击“下载”
前置条件	任务处于“已完成”状态且任务类型为“文本合规性检测”
后置条件	无
正常流程	1. 用户点击“下载”按钮，下载检测结果 2. 报告以 PDF 格式被下载到本地 3. 打开检测报告查看具体内容

3.3.1 系统架构设计

面向金融 APP 的隐私政策合规性检测系统是一个 Web 项目，采用前后端分离的开发方式降低了系统的耦合性并提高了开发效率，系统分为前端，服务端和存储层三个大块。本系统的架构设计如图 3.2所示。

本系统的前端采用了轻量且高性能的 Vue 框架，UI 组件库选用的是 Elment UI，它是一款成熟且适合快速开发的界面展示框架，在方便上手的基础上保证了页面的美观。用户的操作请求将被发送到 Nginx 反向代理服务器中，再通过负载均衡将请求交付服务器端，待服务器端处理完该请求后，结果会自动返还给客户端。

本系统的服务端主要使用了 Django 框架，该框架有着文档丰富、易于开发，快速部署等特点，能与许多第三方组件进行集成。Django 框架的设计模式借鉴了 MVC 框架的思想，可分成 Model（模型）、Template（模板）和 View（视图）三个部分，因此被称为 MTV 框架。此外，Django 还有一个 urls 分发器，它能

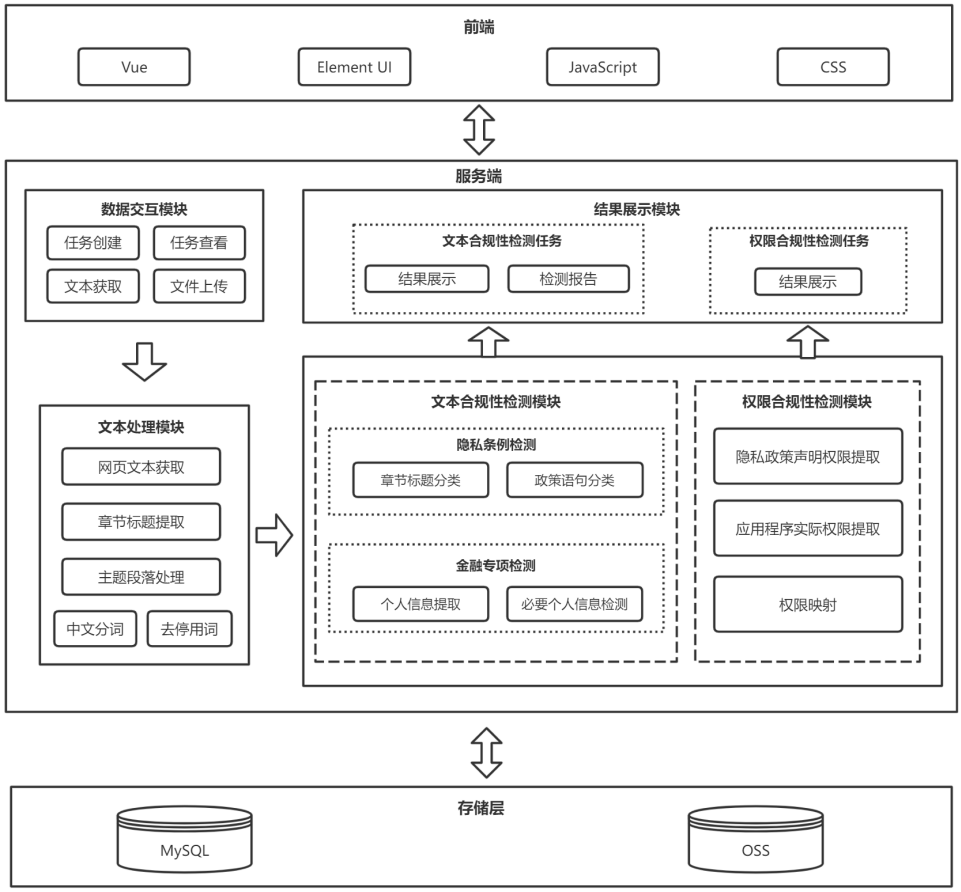

图 3.2: 系统整体架构图

够将不同 URL 的请求分发给对应的 View 处理，View 再调用相应的 Model 和 Template。

服务端共有五个模块，其中文本合规性检测模块和权限合规性检测模块是该系统的两个核心模块。

文本合规性检测模块分为隐私条款检测和金融专项检测两个部分，隐私条款检测的主要功能是检测 APP 的隐私政策是否符合隐私政策检测指标的要求，该模块首先会调用文本处理模块处理隐私政策文本。然后对隐私政策进行两轮分类，第一轮分类按章节标题进行分类，为该章节标题分配一个主题；第二轮分类基于每个主题的特点，分别采用了关键词匹配的分类方法和基于 BERT 分类器的分类方式。最后会将分类的结果进行记录保存。金融专项检查为金融理财类、手机银行类以及网络借贷类这三种类型的金融 APP 提供了检测，主要检测了隐私政策中收集的个人信息是否为必要信息。两项检测的结果信息会被保存到数据库中。

权限合规性的主要功能是检测 APP 隐私政策中声明的权限是否与 APP 实际使用的权限一致。该模块先通过分类为“申请的设备权限”的内容获取声明权限集，再使用 Androguard 解析的 APK 文件，获取 APP 的使用权限集，之后通过 SimCSE 进行语义相似度分析，判断声明权限集是否包含了使用权限集。检测的结果信息会被保存到数据库中。

存储端主要采用 MySQL，MySQL 是目前最主流的关系型数据库，主要负责存储结构化的数据，包括用户信息、任务信息和任务结果信息等。隐私政策，APK 文件和检测报告等文件存储在服务器上，并在 MySQL 对应的表中记录存储路径。

3.3.2 4+1 架构视图

在软件结构的设计中，只用单一的视图无法进行完整的描述，本小节采用了“4+1”视图模型，从逻辑视图、进程视图、物理视图、开发视图和场景视图这五个方面进行了对系统进行详细描述。每一个视图只关心系统的一个侧面，5 个视图结合在一起反映了系统软件体系结构的全部内容。场景视图类似于 3.2 小节中的系统用例图，在此处不再重复，接下来将从其余四个视图的角度对系统进行说明。

逻辑视图对系统逻辑进行分层，从业务角度来介绍系统，描述了系统最终提供给用户的功能及服务，具体如图 3.3所示。TaskConstructService 用于检测任务的创建，TaskSubmitService 用于填写任务相关信息以及上传 APP 隐私政策和 APK 文件，然后通过 FileInteraction 将任务相关信息和文件存入 MySQL 或服务器中，TextAnalysisService 对隐私政策文本进行处理，提供分词、标题提取和段落提取等功能，方便其他服务的使用。APKAnalysisService 主要对 APK 文件进行解析，通过静态分析获取敏感权限。TextComplianceService 提供了文本合规性检测，通过调用 PolicyDetectionService 类进行隐私条款检测以及调用 SpecialSubjectService 进行金融专项的检测。PermissionDetectionService 提供了权限合规性检查检测。每种检测任务的结果会在 StorageService 中进行存储。TaskQueryService 提供了对检测任务的查询功能，调用 ResultService 展示任务结果，ResultService 从 StorageService 中获取结果相关的数据，并在页面中进行展示。

开发视图从开发者的角度来介绍系统，将系统的开发架构描绘出来，开发视图主要分为 UI 和 Service Logic 两部分，如图 3.4所示。

UI 部分使用 Vue 作为开发框架，使用 JavaScript 作为业务逻辑的开发语言，使用 Element UI 组件库提高了开发速度和效率。用户通关 UI 层创建任务和管理任务，并通过 UI 层传送文本数据和 APK 文件到服务端，此外，用户还可以通

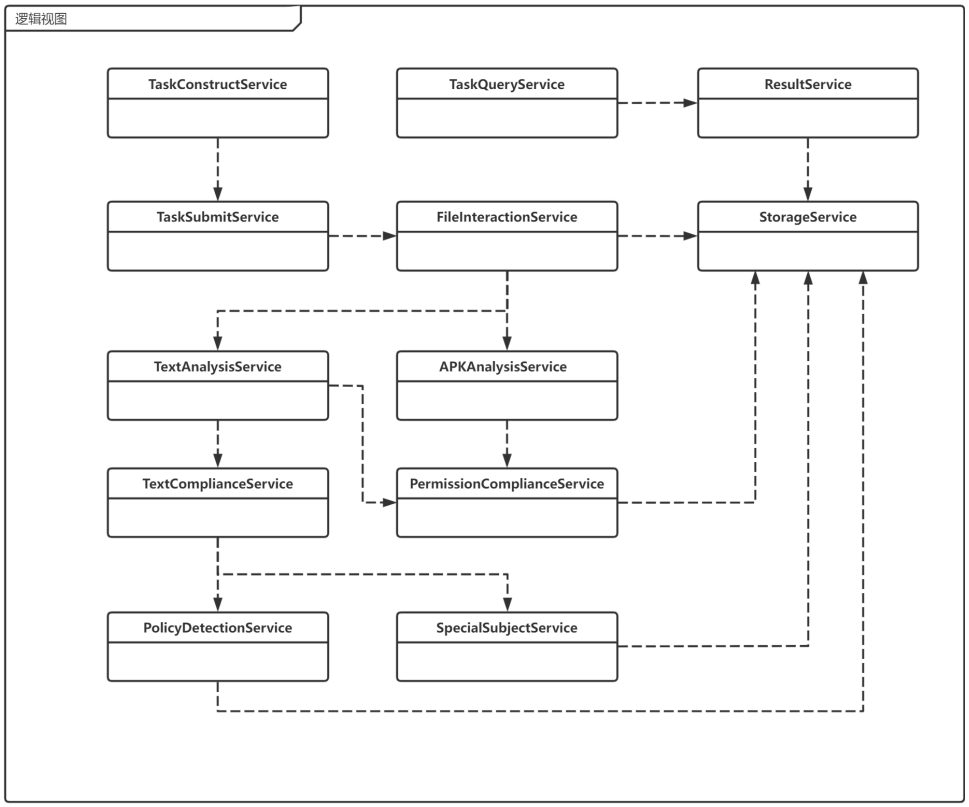

图 3.3: 逻辑视图

过 UI 层下载检测报告。

Server Logic 是业务的逻辑层，为了方便展示，这里将 Django 的 MTV 模式抽象成了 MVC 模式，分为 Controller，Service 和 DAO 三个部分。Controller 层面向外部的 HTTP 的请求进行接受和转发。TaskConstructController 提供构建任务的相关接口，TaskQueryController 构建任务查询的相关接口。TextComplianceController 构建文本合规性检测任务的相关接口，ConsistenceComplianceController 构建权限合规性检测任务的相关接口。Service 层负责实际的业务逻辑处理，实现 Controller 向外提供的接口，主要通过调用子模块实现业务逻辑。DAO 层负责实体类与数据库之间的交互。

进程视图从系统集成人员的角度描述系统的数据通信过程，本系统的进程视图如图 3.5所示。在本系统中，系统的启动是用户创建检测任务开始的。用户创建检测任务时，首先会填写任务必要信息并上传相应的文件。文件上传成功后，就开始任务的创建，任务创建成功后启动对应任务的执行进程。任务执行进程根据任务类型先从存储层中获取任务所需的相应文件并执行相应的服务。任务执行完成后，会将检测结果进行存储。主进程如果要查看任务结果，会调用结

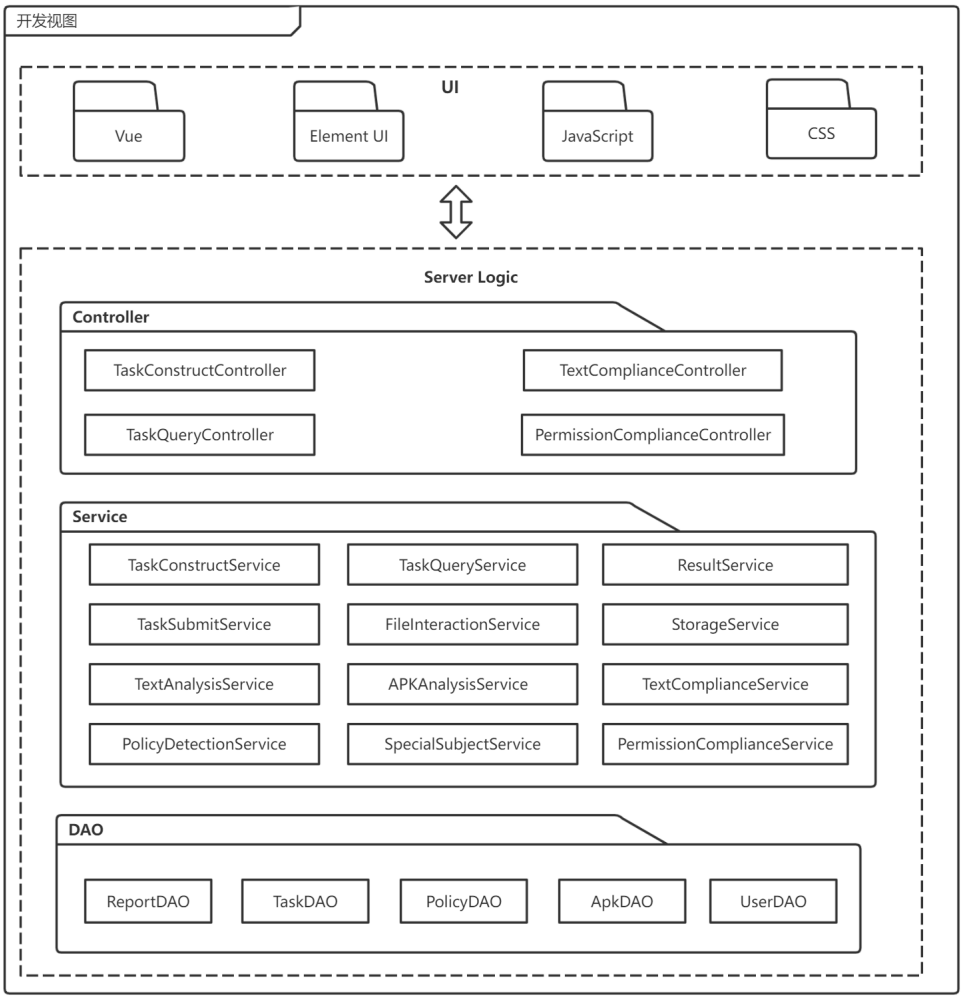

图 3.4: 开发视图

果生成进程，该进程会先从存储层中获取结果数据，然后根据任务类型生成报告，最后将检测结果返回给主进程。

物理视图反映了系统在真实部署状态下的运行模式，方便运维人员快速地理解系统运行情况。本系统的物理视图如图 3.6所示。本系统的服务由用户主动发起，用户通过本地浏览器主动发送 http 请求，申请访问金融 APP 隐私政策合规性检测系统，Nginx 服务器进行代理转发，让用户的请求到达应用服务器。本系统的服务端基于 Django 框架开发的，通过 Rquest 模组解析用户发送的 Http 请求报文执行业务，然后根据业务类型返回 JsonResponse 和 FileResponse 报文。应用服务器上部署了各种服务，包括文本合规性检测服务，权限合规性检测服务，文本处理服务，结果生成服务等。服务端与 Mysql 数据库相连接进行数据存储更新等操作，一些数据隐私政策、APK 文件和检测报告等则以文件形式存储在服务器上。

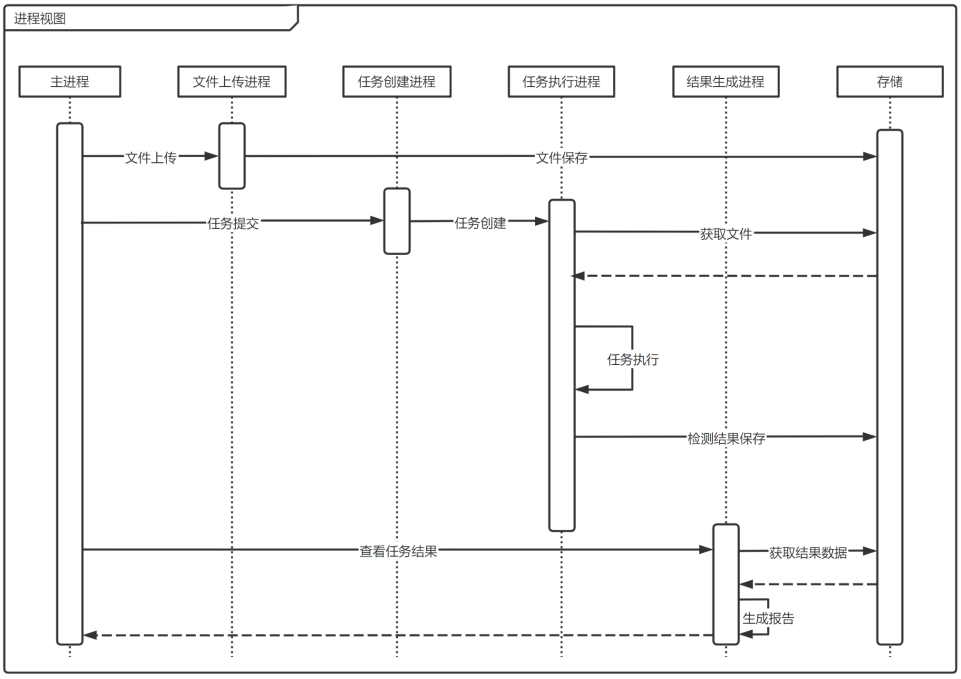

图 3.5: 进程视图

3.4 数据库设计

本系统的 ER 图如图 3.7所示，ER 图表现了数据对象的各属性与实体之间的关系，以及各个实体之间的关系。User 表记录了用户的账号和登录信息，Task 表是系统的检测任务，检测任务有两种类型，文本合规性任务与权限合规性检测，Privacy 表记录了用户创建文本合规性检测任务或权限合规性检测任务时上传的隐私政策文件信息，APK 表记录了用户创建权限合规性任务时上传的 APK 文件信息，PrivacyResult 表记录文本合规性检测任务的结果信息，PermissionResult 表记录了权限合规检测任务的结果信息，Report 表记录了文本合规性检测任务生成的检测报告文件的相关信息。其中 User 表和 Task 表是一对多的关系，一个用户可以拥有多个任务，Task 表和 Privacy 表以及 APK 表是一对一关系，即一个检测任务里最多包含一个隐私政策和一个 APK 文件信息。Task 表和 Report 表是一对一关系，一个检测任务只会生成一个报告，Task 表与 PrivacyResult 以及 PermissionResult 表是一对一关系，即一个检测任务最多只有一个任务结果。

本文系统使用 MySQL 数据库对结构化数据进行存储主要存储内容有：用户信息、任务信息、隐私政策文件信息，APK 文件信息，检测结果等数据，按照需求分析以及系统设计，定义如下持久化对象。

表 3.11展示了用户信息表，该表存储了用户的信息。表中的 ID 为一个用户

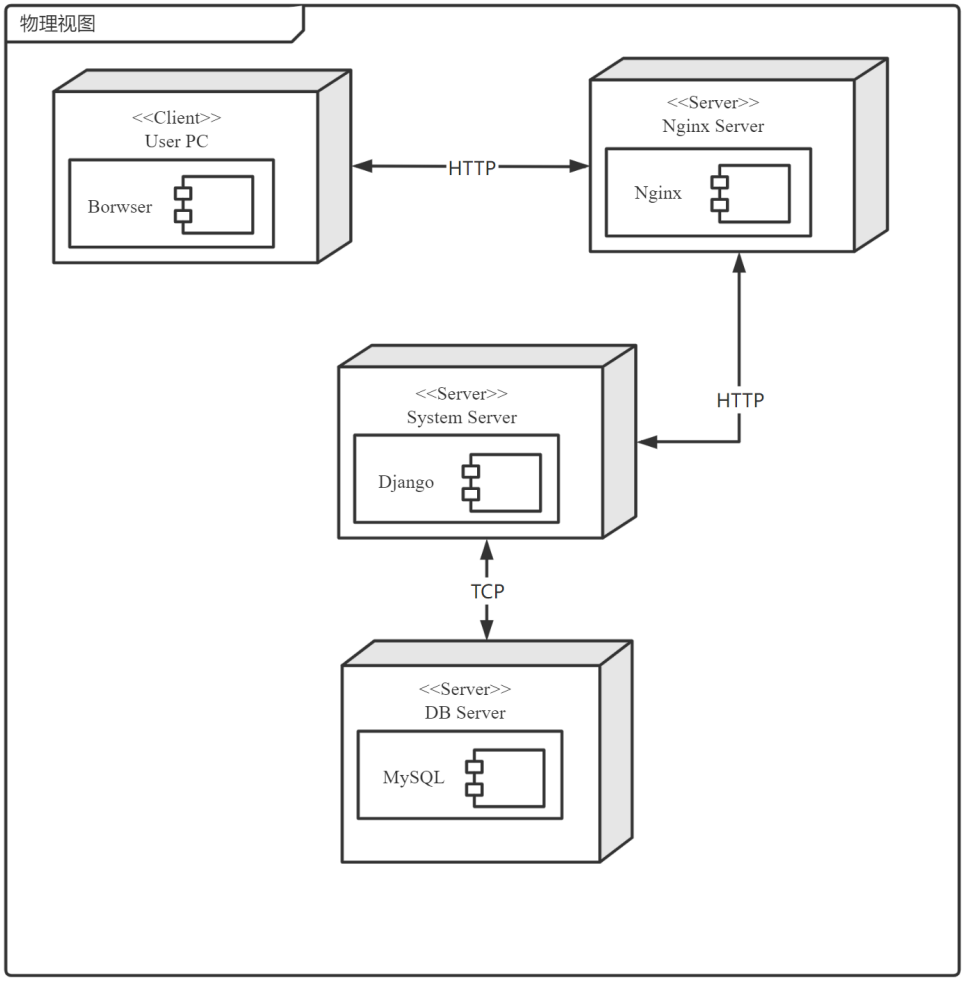

图 3.6: 物理视图

的主键，作为与其他表相连接时的外键；user_name 是用户名，用于用户登录时使用，password 是用户的登录密码；is_online 是一个布尔型变量，判断用户是否在线。为该用户的创建时间。

表 3.11: 用户表

字段名	字段类型	描述
id	INT(11)	自增 ID，主键
user_name	VARCHAR(200)	用户名
password	VARCHAR(200)	密码
is_online	TINYINT	用户是否在线

表 3.12展示了检测任务表，该表存储了检测任务的相关信息。表中的 id 表示一个任务的主键，作为与其他表相连接时的外键；task_name 是任务的名称；

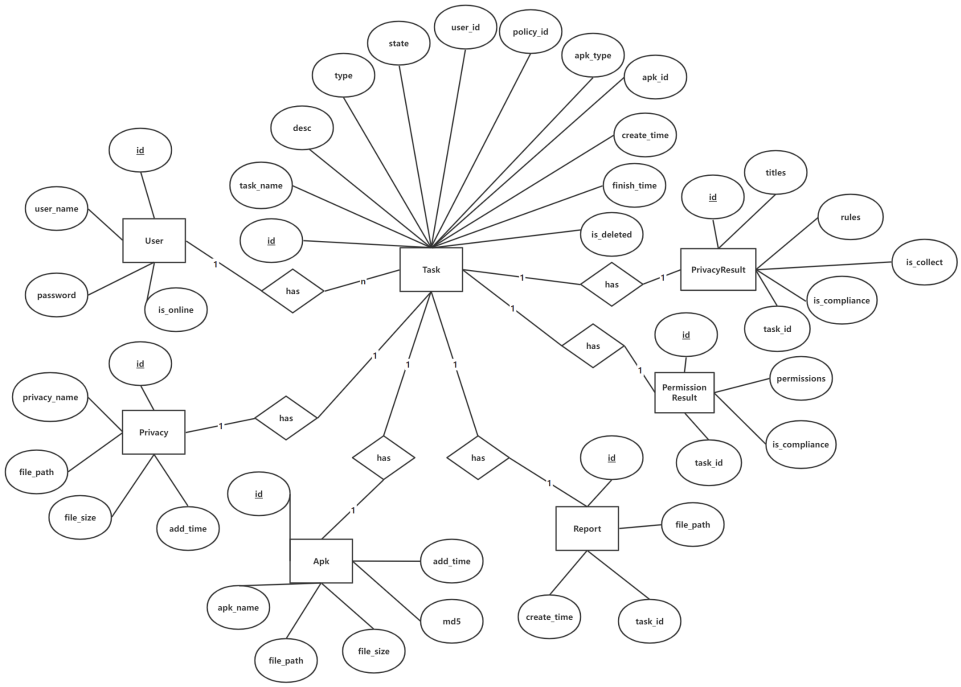

图 3.7: ER 图

desc 是任务是描述；app_name 是待检测 APP 的名称，这三项都在创建任务时填写，其中 desc 为选填；app_type 是金融 APP 的类型，共分为三种，分别是金融理财类、手机银行类以及网络借贷类；type 是任务类型，共分为两种，分别是文本合规性检查和权限合规性检查；state 是任务状态，共分为三种，分别运行中、已完成和失败；user_id 是用户的主键，用来关联 User 表；policy_id 是隐私政策文件表的主键，用来关联 PrivacyPolicy 表，apk_id 是 APK 文件表的主键，用来关联 APK 表；create_time 是用户创建该任务的时间；finish_time 是该任务运行完成的时间；is_deleted 用来判断该任务是否已被删除。

表 3.13展示了隐私政策表，该表存储了检测任务中上传的 APP 隐私政策的相关信息。表中的 id 是隐私政策表的主键，作为与其他表相连接时的外键；policy_name 是隐私政策表名，通过直接读取上传的文件名获取；file_path 是隐私政策上传后的存储路径，file_size 是隐私政策的文件大小；add_time 是隐私政策被添加的时间。

表 3.14展示了 APK 信息表，该表存储了检测任务中上传的 APK 文件的相关信息。表中的 id 是 APK 信息表的主键，作为与其他表相连接时的外键；apk_name 是 APK 名，通过直接读取上传的文件名获取；file_path 是 APK 文件上传后的存储路径；file_size 是 APK 的文件大小；md5 用来检测是否被重复上传。add_time

表 3.12: 任务表

字段名	字段类型	描述
id	INT(11)	自增 ID，主键
task_name	VARCHAR(200)	任务名
desc	VARCHAR(200)	任务描述
app_name	VARCHAR(200)	待检测的 APP 名称
app_type	TINYINT	待检测的 APP 类型
type	TINYINT	任务类型
state	TINYINT	任务状态
user_id	INT(11)	创建任务的用户主键
policy_id	INT(11)	上传的隐私政策主键
apk_id	INT(11)	上传的 APK 主键
create_time	TIMESTAMP	任务的创建时间
finish_time	TIMESTAMP	任务的结束时间
is_deleted	TINYINT	任务是否被删除

表 3.13: 隐私政策表

字段名	字段类型	描述
id	INT(11)	自增 ID，主键
privacy_name	VARCHAR(200)	隐私政策名
file_path	VARCHAR(200)	隐私政策存储的路径
file_size	INT(11)	隐私政策文本大小
add_time	TIMESTAMP	隐私政策被添加的时间

是隐私政策被添加的时间。

表 3.14: APK 信息表

字段名	字段类型	描述
id	INT(11)	自增 ID，主键
apk_name	VARCHAR(200)	APK 应用名
file_path	VARCHAR(200)	APK 存储的路径
file_size	INT(11)	APK 文件大小
md5	VARCHAR(200)	APK 的文件的 md5 值
add_time	TIMESTAMP	APK 被添加的时间

表 3.15展示了文本合规性检测结果表，该表存储了文本合规性检测任务的结果。表中的 id 是该表的主键，作为与其他表相连接时的外键；titles 是分类的章节题目与对应主题，是一个 Json 字符串；rules 是分类的隐私条款及对应数量，是

一个 Json 字符串；is_compliance 用来判断隐私条款检测任务是否合规；is_collect 用来判断金融专项检测任务是否合规；task_id 是该结果信息所属任务的主键。

表 3.15: 文本合规性检测结果表

字段名	字段类型	描述
id	INT(11)	自增 ID，主键
titles	TEXT	分类的章节题目及对应主题
rules	TEXT	分类的隐私条款类型及数量
is_compliance	TINYINT	隐私条款检测结果是否合规
is_collect	TINYINT	金融专项检测结果是否合规
task_id	TIMESTAMP	结果所属任务主键

表 3.16展示了权限合规性检测结果表，该表存储了权限合规性检测任务的结果。表中的 id 是该表的主键，作为与其他表相连接时的外键；permisions 是检测出的实际使用权限与隐私政策中对应声明的权限，是一个 Json 字符串；is_compliance 用来判断任务检测结果是否合规；task_id 是该结果信息所属任务的主键。

表 3.16: 权限合规性检测结果表

字段名	字段类型	描述
id	INT(11)	自增 ID，主键
permisions	TEXT	权限对应关系
is_compliance	TINYINT	检测结果是否合规
task_id	INT(11)	结果所属任务主键

表 3.17展示了检测报告表。该表存储了文本合规性检测任务生成的检测报告相关信息。表中的 id 是检测报告表的主键；file_path 是报告后的存储路径；task_id 是报告所属的任务的主键；create_time 是报告的生成时间。

表 3.17: 检测报告表

字段名	字段类型	描述
id	INT(11)	自增 ID，主键
file_path	VARCHAR(200)	报告生成后存储的路径
task_id	VARCHAR(200)	报告所属的任务主键
create_time	INT(11)	报告生成时间

3.5 本章小结

本章首先介绍了系统的整体概述以及五个模块的功能，然后对系统的功能需求和非功能需求进行了分析，并展示了系统用例图，接着通过 4+1 视图对系统的整体架构设计进行了全方位的讲解，最后对系统的数据库设计以及 ER 图进行了说明。

第四章　系统的详细设计与实现

本章主要介绍金融 APP 隐私政策合规性检测系统的详细设计与实现。本章节先介绍了隐私政策检测指标的构建以及隐私政策数据集的收集。然后根据第三章的需求分析与概要设计，对本系统五个模块的具体实现进行了详细介绍，并通过时序图和核心代码展现了各个模块的运行过程以及具体功能，并对系统的功能页面进行了截图展示。

4.1　隐私政策检测指标的构建

目前，关于金融 APP 隐私政策的合规性检测尚未形成一个通用的考核标准。通过阅读了大量金融 APP 的隐私政策文本，发现隐私政策的内容结构比较固定，普遍由政策基本信息、个人信息的收集与使用、个人信息的共享与披露、个人信息的存储与保护、cookie 及其相关技术的使用，未成年人特殊保护和用户权利保障等几个部分组成。在学术研究方面，大量学者均基于隐私政策的内容结构设计了大量隐私政策的测评指标，包含信息的收集、使用、保存、对外提供、用户权利、cookie 技术等方面。在法律规范方面，我国颁布了一系列法律法规与标准要求，同时金融领域也提出了诸多行业规则，以规范和指导相关企业制定隐私政策，相应法规如表 4.1所示。

表 4.1: 隐私政策法规依据

类型	名称	执行时间
法律	《中华人民共和国网络安全法》	2017.06.01
	《中华人民共和国电子商务法》	2019.01.01
标准文件	《个人金融信息保护技术规范》	2020.02.13
	《信息安全技术个人信息安全规范》	2021.10.01
政策文件	《App 违法违规收集使用个人信息自评估指南》	2019.03.01

因此，本文基于《中华人民共和国网络安全法》、《个人金融信息保护技术规范》、《信息安全技术个人信息安全规范》、《App 违法违规收集使用个人信息行为认定方法》和《App 违法违规收集使用个人信息自评估指南》等相应法规，结合了学术界已有的测评维度并参考了金融 APP 隐私政策的内容特征，涉及了一份金融 APP 隐私政策检测指标，具体内容如表 4.2所示。该规范共包含 9 个主题以及 25 个隐私条款，即任何一个 APP 的隐私政策内容必须要满足这 25 个隐

私条款，否则该 APP 的隐私政策不符合法律规范。

表 4.2: 金融 APP 隐私政策检测指标

主题	隐私条款
A1 政策基本信息	B1 运营商情况 B2 政策时效
A2 收集与使用	B3 收集和使用个人信息的目的 B4 收集和使用个人信息的方式 B5 收集的个人信息类型 B6 申请的设备权限 B7 信息的第三方获取
A3 存储与保护	B8 个人信息存储地点和期限 B9 个人信息安全保护措施 B10 发生安全事故处理方式
A4 共享与披露	B11 共享个人信息的目的 B12 共享的个人信息类型 B13 接收方的类型或身份 B14 征得授权同意的例外 B15 个人信息跨境传输的说明
A5 使用 cookie 等技术	B16 使用该技术的目的 B17 停用此类技术的方法
A6 未成年人特殊保护	B18 未成年人使用的说明
A7 用户权利	B19 如何访问个人信息 B20 如何修改个人信息 B21 如何删除个人信息 B22 改变授权同意的范围 B23 如何注销账号
A8 通知和修订	B24 隐私政策变更后如何通知用户
A9 联系方式	B25 提供公司的联系方式

4.2 隐私政策数据集的构建

目前，互联网上存在的中文隐私政策数据集较少，专门针对金融 APP 隐私的数据集几乎是空白。同时，APP 的隐私政策具有较强的时效性，一般三个月到半年就会更新一次，几年前的隐私政策数据集并不能很好地适应当下的法律环境，因此需要自己构建金融一份 APP 的隐私政策数据集，本文将从以下几个方面考虑：

(1) 样本来源：目前主流的安卓应用市场有华为应用市场，应用汇和小米应用市场。其中华为应用市场中的 APP 分类最细致，上架审核最严格，其 APP 的隐私政策更加适合做为数据集，因此选取了华为应用市场作为隐私政策的样本

来源。华为应用市场网页的每款 APP 介绍里均含有该 APP 的隐私政策链接，可以通过编写爬虫程序来获取所有 APP 的隐私政策网页链接。然后再编写转化程序，通过访问隐私政策的网页链接，将网页内容提取到 txt 文本中，从而获取该应用隐私政策文本。

(2) 样本选取条件：由于本系统需要对金融理财、手机银行以及网络借贷这三种类型的金融 APP 进行检测，因此需要获取这三个种类的金融 APP 隐私政策样本。由于每个公司法务部的水平不一致，应当要选取隐私政策编写得更加规范的作为样本。因此，本次隐私政策的样本选取将从以下两个方面进行考量：一是金融 APP 的类别，根据华为应用市场的划分，金融 APP 分为了股票基金，贷款，理财，银行四个大类，其中股票基金和理财对应金融理财类，银行对应手机银行类，贷款对应网络借贷类，在筛选样本时还需要保证三个类型数量的均衡。二是该 APP 的下载量，下载量是衡量一个软件普及度和用户量的重要指标，拥有更多用户的互联网企业会承担更高的社会责任，有理由相信下载量更大的 APP 制定的隐私政策更加完善。因此本文将会在爬取隐私政策链接同时获取该 APP 的下载量，并选取下载量靠前的金融 APP 的隐私政策做为研究样本。

(3) 样本容量：为了实验的准确性，样本容量不宜太小，因此本次实验计划选取 3 个类别的隐私政策样本各 50 个，共计 150 个政策样本。

根据上述的隐私政策选取标准，通过以下步骤获取隐私政策数据集。

获取隐私政策链接：通过 python 编写爬虫程序，按金融理财、手机银行和网络借贷这三个类别，从华为应用市场爬取该类别下所有 APP 的隐私政策链接，并按下载量从高到低进行排序。

获取隐私政策文本：删除所有非 html 格式的链接（如 pdf，doc 等），然后通过 Python 的 newspaper 库将所有 html 转化成隐私政策的 txt 文本。由于部分 html 链接指向的并非隐私政策，而是官网链接，或者网页无效，导致解析失败，因此需要对这些文本进行剔除，还有一些字数过少内容明显不合规的也需要剔除。最后根据获取了共计 150 个隐私政策文本做为后续实验的数据集。

4.3 数据交互模块

数据交互模块用来处理用户的文本获取，任务创建，文件上传，任务查看等请求。该模块主要涉及 Task，Privacy，Apk 这三个模型类，通过 model.py 直接映射在数据库中。

4.3.1 时序图

数据交互模块的时序图如 4.1所示，具体功能通过以下的方法进行实现。

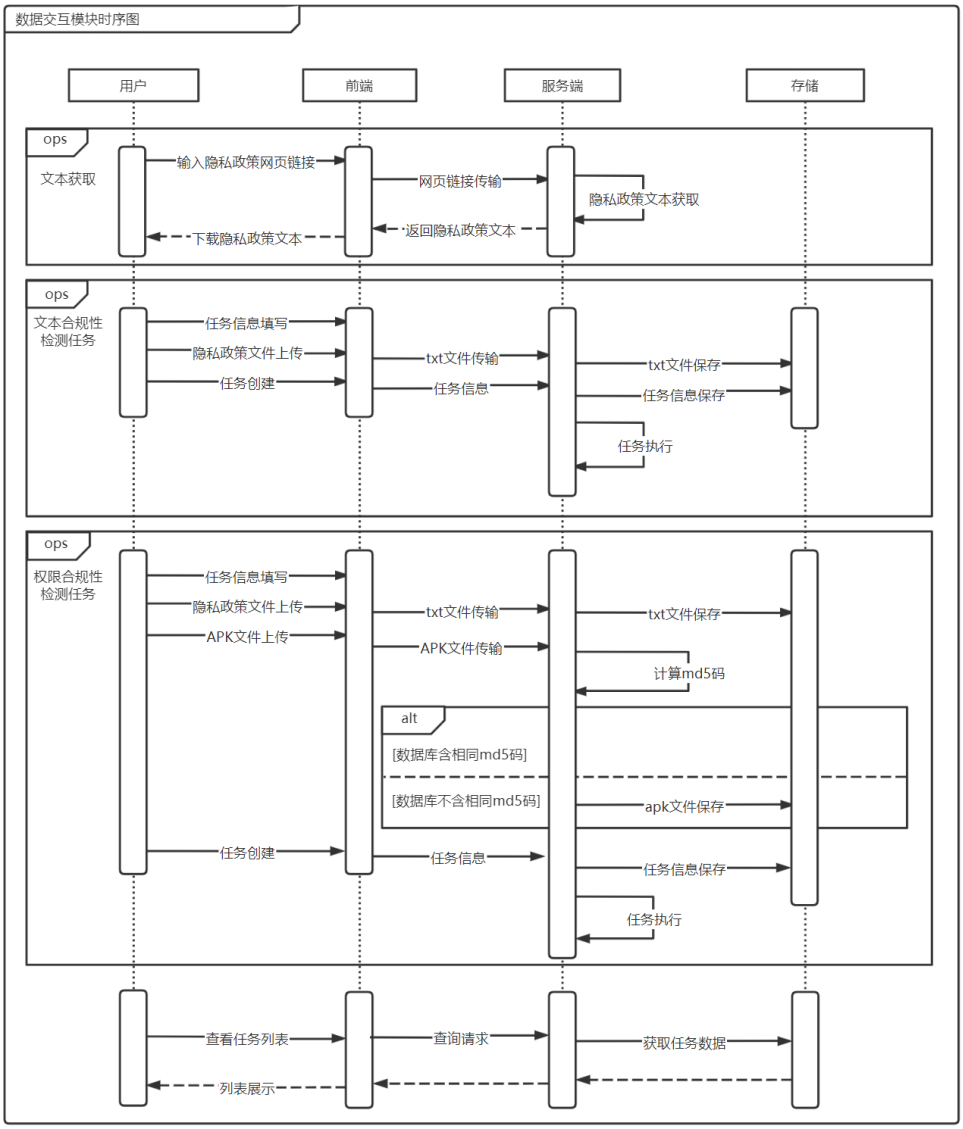

图 4.1: 数据交互模块时序图

TextAcquisition 负责文本获取功能。TextAcquisition 从前端的 HttpRequset 请求中解析出用户输入的隐私政策网页链接，然后调用文本处理模块的网页文本获取功能生成 txt 文件，然后将 txt 文件写入 FileResponse 中返回给前端。

PrivacyUpload 负责隐私政策文本的上传。PrivacyUpload 从前端的 HttpRequset 请求中解析出用户上传的隐私政策文件和文件名，并将文件存储在服务器中，然后将文件信息通过创建 Privacy 类的方式存储在 MySQL 中，最后将新记录的主键写入 JsonResponse 中返回给前端。

APKUpload 负责隐私 APK 文件的上传，APKUpload 从前端的 HttpRequset 请求中解析出用户上传的 APK 文件和文件名，由于 APK 文件较大，直接存储

会造成空间的压力，因此 APKUpload 会先计算 APK 文件的 md5 码，并在数据库中查询，如果找到相同的 md5 码，直接该条记录的主键写入 Response 中返回给前端。否则会将 APK 文件存储在服务器中，并通过 Apk 模型类将文件信息存储在 MySQL 中，然后将新记录的主键写入 JsonResponse 中返回给前端。

TaskCreate 负责任务的创建，通过前端 HttpRequest 获取任务名称，APP 名称，任务描述，任务类型，APP 类型以及文件的主键等信息，然后通过新建 Task 模型类的方式在 MySQL 中进行保存，之后调用 TaskExecute 进行任务的执行。

TaskShow 负责任务的展示，考虑到单个用户的任务规模不会太大，直接将用户给前端返回了用户所有的信息。使用 Django 的 filter 方法获取数据库中 Task 表的记录，然后通过 serializers.serialize 将这些记录序列化，再使用 JsonResponse 将任务信息通过 Json 格式返回给前端。

4.3.2 界面展示

如图 4.2是系统的文本获取界面，用户可以在页面中输入某个 APP 隐私政策的网址链接来获取隐私政策文本。该功能通过调用文本处理模块的网页文本获取功能实现，隐私政策文本将会尽可能地保留原始页面的段落格式。生成后的隐私政策文本会通过浏览器自动下载到本地。

图 4.2: 文本获取界面

如图 4.3是系统的任务创建界面，用户在该界面首先需要填写任务名称、任务描述和 APP 名称，其中任务名称和 APP 名称为必填项。接着需要选择任务类型，共有两种任务可以选择，分别是文本合规性检测任务和权限合规性检测任务。文本合规性检测任务需要上传隐私政策文件并选择待检测金融 APP 的类型，本系统为网络借贷类、金融理财类和手机银行类这三种类型的 APP 提供了金融专项检测，可以检测出隐私政策是否收集了非必要的个人信息。权限合规性检测任务需要上传隐私政策文件和 APK 文件。

选取文件后会在页面自动生成文件信息，文件上传时会有进度条的显示，上传成功后会有成功标志的显示。文件上传操作如图 4.4所示。完成以上工作后，

图 4.3: 任务创建页面

点击“创建任务”按钮，系统会自动进行任务的创建工作。

图 4.4: 文件上传操作

如图 4.5是系统的任务管理页面。主要展示了任务的名称、APP 名称，任务类型，任务的状态，任务的描述，任务的创建时间和任务操作。页面还对任务列表进行了分页处理，顶部提供对任务的搜索功能，可以输入关键字来查询系统中的任务。任务列表默认按照任务开始时间进行排序。列表中每一条都有两个操作项”详情”和“删除”，用户可以对状态为“已完成”的任务进行查看，对不

需要的任务进行删除。

图 4.5: 任务管理页面

4.4 文本处理模块

文本处理模块负责给其他模块提供一些文本处理方面所需要功能，主要包含了文本类型转换，章节标题提取，文本段落处理，中文分词，去停用词等功能。

4.4.1 网页文本获取

网页文本获取功能是将隐私政策的网页链接转化成隐私政策文本形式，考虑到隐私政策文本的页面结构会对后面的，需要在转换的同时对页面结构进行保留，因此这里选取了 python 的 newspaper 库对 url 进行解析，newspaper 库能够十分快速地提取正文内容，使用过程不需要考虑 HTTP Header、网页解析和网页源代码架构等问题，同时还能保留网页的文文本段落结构。提取后的每段之间会有额外的空行，还需要去除空行的操作。

4.4.2 章节标题提取

章节标题提取功能能提取每个隐私政策中每个段落章节前的标题，方便隐私合规性检测中对标题进行分类的操作。分析了收集的 150 份隐私政策文本的章节标题特征，发现其中 137 份都使用了“一、”，“二、”等汉字数字加顿号的形式做为一级标题，6 份使用了“一.”，“二.”等汉字数字加小数点的形式做为一级标题，4 份使用了“(一)”，“(二)”等括号加汉字数字的形式做为一级标题，2 份使用了“1、”，“2、”等阿拉伯数字加顿号做为标题，1 份的标题前面没有任何的数字化标注。通过以上规律，并结合正则表达式，编写了相应程序完成了对章节标题的提取。章节标题提取的核心代码如图 4.6所示。

```
def extract(content):
    #……
        #判断是标题是哪种格式
        while 1:
            try:
                if tag == 0:
                    tag = tag +1
                    str = extractTitle(content,"一、","二、")
                    break
                if tag == 1:
                    tag =tag +1
                    str = extractTitle(content,"一.","二.")
                    break
                #.......
            except:
                #.......
        #提取标题
        while tag==1:
            string = extractTitle(content,"一、","二、")
            #......
def extractTitle(content, s1, s2):
    global flag
    restr1 = s1+'(.*?)'+s2
    restr2 = s1+'(.*)'
    try:
        s = re.findall(restr1, content, re.S)[-1].strip()
        s=s.split("\n")[0]
        return s.split(" ")[0]
    except:
        s = re.findall(restr2, content, re.S)[-1].strip()
        flag = 1
        s=s.split("\n")[0]
        return s.split(" ")[0]
```

图 4.6: 章节标题提取代码

除去没有数字标注的一篇外，总共对 149 篇文本的章节标题进行了提取，成功提取了 145 篇，提取成功率为 97.3%，部分提取的章节题目如下图 4.7所示。对章节标题提取失败的文本进行分析，发现原因均是它们自身的章节标题书写不规范，如“付临门”APP 中，出现了第四章节之后就是第七章节的情况，所以导致了提取错误。将这些文本进行修改后，均成功完成了提取。

1441	陕西信合	我行（社）如何收集和使用您的个人信息
1442	陕西信合	我行（社）如何使用
1443	陕西信合	我行（社）如何共享、转让、公开披露您的个人信息
1444	陕西信合	我行（社）如何存储和保护您的个人信息
1445	陕西信合	您控制个人信息的权利
1446	陕西信合	我行（社）如何处理儿童的个人信息
1447	陕西信合	本政策如何更新
1448	陕西信合	如何联系我行（社）
1449	雪球	本隐私政策的管辖法律和适用范围
1450	雪球	我们如何收集和使用你的个人信息
1451	雪球	我们如何存储你的个人信息
1452	雪球	我们如何使用Cookie和同类技术
1453	雪球	我们如何委托处理、共享、转让、公开披露你的个人信息
1454	雪球	我们如何保护你的个人信息
1455	雪球	你管理你个人信息的权利
1456	雪球	未成年人个人信息保护
1457	雪球	本隐私政策的更新及如何联系我们
1458	雪球	争议解决
1459	青岛银行	我行如何收集和使用您的个人信息
1460	青岛银行	我行如何使用
1461	青岛银行	我行如何共享、转让和公开披露您的个人信息
1462	青岛银行	我行如何存储和保护您的个人信息
1463	青岛银行	我行如何使您拥有管理个人信息的权利
1464	青岛银行	未成年人信息的保护
1465	青岛银行	第三方收集和使用您信息的情况
1466	青岛银行	本政策如何更新
1467	青岛银行	如何联系我行

图 4.7: 隐私政策章节标题展示

4.4.3 文本段落处理

文本段落处理功能主要是根据章节标题获取该章节下的文本内容，然后对这些内容进行分句处理。在研究隐私政策文本时，发现隐私政策中对某项隐私条款进行阐述时，通常会采用一段一段的形式，里面可能包含了若干个句子，如图 4.8中的“语句段 3”所示；或者通过分条列举的方式陈述，如图 4.8中的“语句段 1”所示。这对于人工阅读来说一种很正常的表达方式，但如果按照传统的以句号或换行符做为间隔进行分句处理的话，会将这些这些语句段分成多个单独的语句，这不仅增加了人工标签的难度，还会导致某些句子缺少完整的上下文语境，不利于机器学习分类。

因此，本文在进行分句处理时，会直接对每个自然段进行保留。对于多个句子分条陈述的情况，通过分析总结了一般规律，这些语句一般分为起始句和枚

(五) 共享、转让、公开披露个人信息时事先征得授权同意的例外

在以下情形中，共享、转让、公开披露个人信息无需事先征得您的授权同意：

语句段1

1. 与国家安全、国防安全有关的；
2. 与公共安全、公共卫生、重大公共利益有关的；
3. 与犯罪侦查、起诉、审判和判决执行等有关的；
4. 为应对突发公共卫生事件，或者紧急情况下为保护自然人的生命健康和财产安全所必需；
5. 所收集的个人信息是个人信息主体自行向社会公众公开的；
6. 从合法公开披露的信息中收集的您的个人信息的，如合法的新闻报道、政府信息公开等渠道；
7. 为订立、履行您作为一方当事人的合同所必需，或者按照依法制定的劳动规章制度和依法签订的集体合同实施人力资源管理所必需；
8. 用于维护所提供的产品与/或服务的安全稳定运行所必需的，例如发现、处置产品与/或服务的故障；
9. 为公共利益实施新闻报道、舆论监督等行为，在合理的范围内处理个人信息；
10. 学术研究机构基于公共利益开展统计或学术研究所必要，且对外提供学术研究或描述的结果时，对结果中所包含的个人信息进行去标识化处理的；
11. 为履行法定职责或者法定义务所必需；
12. 法律法规规定的其他情形。

五、我们如何保存及保护您的个人信息

(一) 个人信息保存期限 语句段2

为了您能够使用蜻蜓点金的多样化产品服务以及有效处理可能产生的争议，我们将按照《中华人民共和国证券法》、《网络安全法》、《证券基金经营机构信息技术管理办法》、《证券期货投资者适当性管理办法》等相关法律法规对于数据留存的强制性要求，在您使用蜻蜓点金APP期间（使用期间是指您注销账户之前）及相关行为的留存期间届满前（例如《证券法》规定您的开户资料等信息至少保留20年）保存您的个人信息以及交易记录。在超过上述期限后，我们会对您的信息进行删除或进行匿名化处理（即会对您的个人信息进行处理以使您的身份无法被任何第三方识别）。 语句段3

(二) 个人信息的保存位置 语句段4

您的个人信息均储存于中华人民共和国境内，如需跨境传输，我们将会单独征得您的授权同意。 语句段5

图 4.8: 语句段展示

举句，起始句是对某项说明的一个概括，而枚举句是对该说明的具体解释，它们均符合以下规则：

（1）起始句以冒号结尾。

（2）枚举语句中最后一个枚举句以句号结尾，其余枚举句以分号结尾。

结合以上规则，编写段落处理模块的代码如图 4.9所示。

4.4.4 中文分词及去停用词

中文文本由单个的汉字组成，不能像英文文本一样直接通过英语单词进行分词，而是需要根据词的特点来进行更好的划分。在中文文本处理中，分词是一个基本步骤。好的分词算法可以更好地表达句子意义。本文采用了中文分词工具 Jieba。Jieba 分词分为三种模式，精确模式，全模式和搜索引擎模式，经过测试发现精确模式对隐私政策进行分词的效果最好，因此文本处理模块将采用了 Jieba 分词的精确模式对文本进行分词处理。

为了避免一些出现频率较高且无关词的干扰，分词后一般都会进行去除停用词的操作。在隐私政策文本中，存在部分的词汇，如果不删掉会对文本有一定影响。在这里，先对 150 份隐私政策文本进行分词处理，然后统计了所有词语出现的频率，人工筛选了高频词汇中无意义的词，并编写了停顿词表。中文分词与去停用词代码如图 4.10所示。

```
#根据章节标题获取该章节的文本
def extractSegment(content,titlelist):
    segments = []
    for i in range(len(titlelist)):
        s1 = titlelist[i]
        if i == (len(titlelist)-1):
            regex = s1 + '(.*)'
        else:
            s2 = titlelist[i+1]
            regex = s1+'(.*?)'+s2
        ss = re.findall(regex, content, re.S)
        s = ss[0].strip()
        s='\n'.join(s.split('\n')[0:-1])
        segments.append(s)
    return segments
#按照分句规则进行分句
def extractSentence(segments):
    for i in range(len(segments)):
        sentence ="
        flag = 0
        for s in segments[i].split("\n"):
            sentence +=s
            if s.endswith("：  "):
                flag =1
            else:
                if flag==1 and (s.endswith("；  ") or s.endswith("。")):
                    if s.endswith("。"):
                        flag=0
                        sentence = "
                else:
                    flag = 0
                    sentence = "
```

图 4.9: 文本段落处理代码

4.5 文本合规性检测模块

文本合规性检测模块是本系统的核心模块之一。本节先介绍了一种章节标题的分类方法，可以将隐私政策的章节标题分类成对应的主题，这样能够快速地将同一主题的内容汇总在一起，既减轻了人工对数据集进行标注的操作，又省去了对一些简单主题的分类操作。接下来介绍了隐私条款检测功能和金融专项检测功能的具体设计与实现。最后通过时序图对整个模块的运行流程就行了说明。

```
# 根据停用词表文件生成停用词表
def stopwordslist(filepath):
    stopwords = [line.strip() for line in open(filepath, 'r', encoding='utf-8').readlines()]
    return stopwords

# 对句子进行分词处理
def seg_sentence(sentence, filepath):
    sentence_seged = jieba.cut(sentence.strip(),cut_all = False)
    stopwords = stopwordslist(filepath)
    outstr = ''
    for word in sentence_seged:
        if word not in stopwords:
            if word != '\t':
                outstr += word
                outstr += " "
    return outstr
```

图 4.10: 分词与去停用词代码

4.5.1 章节标题分类器的设计与实现

一个隐私政策文本的长度基本在 5000 字到 25000 字之间，同时涉及了较多的隐私条款。对整个文本中的所有内容进行人工打标签的话，存在阅读时间长，专业需求高，繁琐程度大等问题。由于隐私政策文本通常都由若干个个章节组成，每个章节都有一个标题，而 4.1 节构建的隐私政策检测指标里也分为了主题和隐私条款。因此，可以先根据每一章的章节题目，给该章节分配一个主题标签，再将同一主题标签下的内容汇集在一起，这样既有利于人工打标签，又方便对不同主题进行共性分析。

本小节将实现一个对章节标题进行自动分类的方法。首先从 4.1 节的数据集中选取了 120 个金融类 APP 隐私政策做为训练集，然后使用文本处理模块的标题提取功能，获取这些隐私政策文本的章节标题，共收集了个 1183 个标题。接着根据隐私政策检测指标中的主题，人工为每个标题打上了对应主题的标签。需要值得注意的是，对于主题“政策基本信息”，绝大多数隐私政策文本中不存在相关的章节标题，而是把政策基本信息的描述放在了开头或结尾部分，因此该项主题不进行统计。除此之外，与所有主题均毫无关联的则打上“其他”标签，共计九种标签。对于每个类型的标签，统计了出现的总标题数，以及去重后的标题数，如表 4.3所示。

由于章节标题的句子长度均较短，且句子的相似度较高。因此这里使用 TF-IDF 算法对标题进行特征提取，并结合 SVM 算法实现标题分类模型。模型的构建流程如图 4.11所示。首先将打好标签的标题集，调用文本处理模块进行中文

表 4.3: 每个主题的标题数统计

标签名	总数	去重后总数
收集与使用	215	87
存储与保护	162	71
共享与披露	88	35
使用 cookie 等同类技术	109	58
未成年人特殊保护	112	39
用户权利	132	43
通知和修订	118	45
联系方式	104	28
其他	143	85

分词并去掉停用词，然后构建词频矩阵，之后再使用 TF-IDF 计算每个词语的 TF-IDF 值，对标题进行特征提取。最后构建 SVM 模型并进行训练，并将训练好的模型进行保存。词频矩阵的构建，TD-IDF 的计算以及 SVM 模型的构建均通过 python 的 sklearn 库完成。

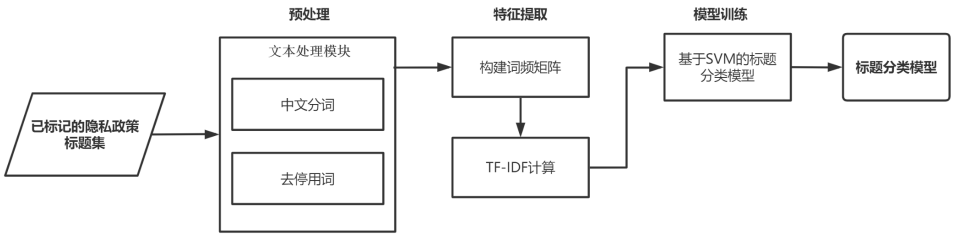

图 4.11: 标题分类器构建流程

4.5.2 隐私条款检测功能的设计与实现

隐私条款检测的作用是检测隐私政策文本是否满足金融 APP 的隐私政策检测指标，因此需要对文本内容进行分类操作。对不同主题下的文本内容和该主题对应的隐私条款进行分析，发现可以将主题分为以下两类：

1. 内容较少，隐私条款含关键词

对于“隐私政策基本信息”、“使用 cookie 等同类技术”、“未成年人特殊保护”“通知与修订”、“联系方式”这五个主题，这些主题下的段落篇幅都较少，且隐私条款可以通过某些关键词进行快速判断。对于这类主题，采用构建关键词表的方法进行匹配。

构建了隐私条款的关键词表如表 4.4所示。用关键词进行匹配的方法设计如

下：先用关键词表中的词语在对应主题的文本中搜索匹配，如果匹配则表明隐私政策符合这项隐私条款，并开始下一项隐私条款关键词的搜索匹配；如果没有匹配到或不存在该主题，则会在类别为“其他”主题下的文本中进行搜索匹配，还没匹配到的话，则表明该金融 APP 的隐私政策缺少这项隐私条款。

表 4.4: 隐私条款的关键词表

主题	隐私条款	关键词
政策基本信息	运营商情况	公司名称 我司名称 公司简介 公司介绍 注册地址
	政策时效	更新时间 生效时间 更新日期 生效日期 发布时间 发布日期
使用 cookie 等技术	使用该技术的目的	为了 以便 登录方便
	停用此类技术的方法	清除 删除 停用 拒绝
未成年人特殊保护	未成年人使用的说明	18 岁 未成年人 监护人 儿童
用户权利	如何访问个人信息	查阅 查询 访问
	如何修改个人信息	更正 补充 修改 更新
	如何删除个人信息	删除 撤销
	改变授权同意的范围	授权同意 撤回同意
	如何注销账号	注销
通知和修订	政策变更后如何通知用户	及时通知 重大变更 消息推送
联系方式	提供公司的联系方式	传真 电话 邮件

2. 内容较多，隐私条款不含关键词

对于“收集与使用”、“存储与保护”和“共享与披露”这三个主题，它们对应的段落篇幅普遍较长，且没有能够显著区分隐私条款的关键词。无法直接通过关键词表的方式进行匹配。因此这里决定了使用机器学习的方式进行分类，分类器采用了基于 BERT 的文本分类模型，为这三个主题都分别设计了一个对应的 BERT 分类器，该主题下的语句段会通过该分类器进行分类，每个语句段都会被预测一个隐私条款的类别。

隐私条款检测的总体流程如图 4.12所示，首先通过文本处理模块来获取隐私政策文本中每个章节的标题，然后调用标题分类器对隐私政策的章节标题进行进行分类，分类器会将每个章节的标题都会被分类到某个主题下，确认好标题的类别后，就开始调用文本处理模块，获取属于标题的段落文本。

根据每个主题下隐私条款的特征，将这些段落文本分为了两类，一类是可以通过关键词明显区分的，若匹配上了，则将该句子分类到该标签下；另一类隐私条款难以通过关键词直接区分，因此为这些无法通过关键词匹配进行分类的主题，都单独训练了一个分类器，这里采用了 BERT 做为分类的模型，将该

主题下的文本段按一定规则切割成若干语句段后，使用对应的分类器进行分类。最后对所有分类结果统计，如果该政策文本能够涵盖所有检测指标，则证明该文本符合相关的法律法规。

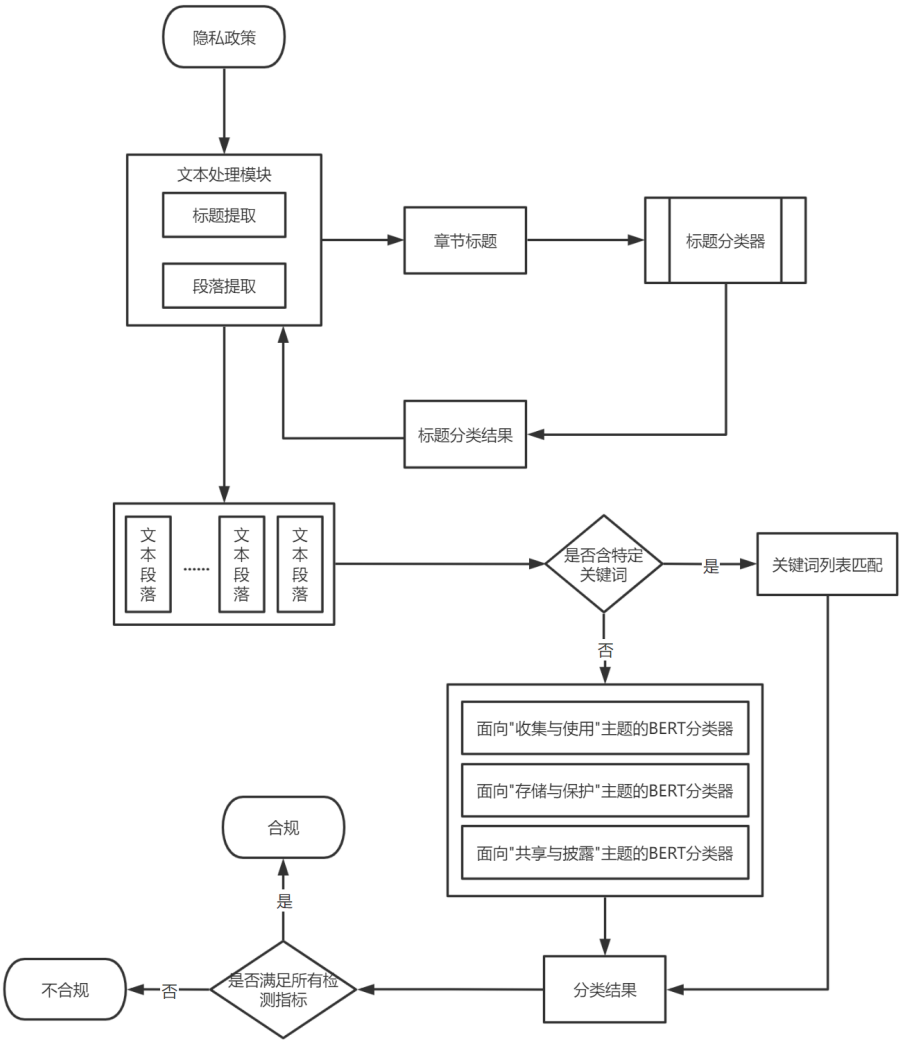

图 4.12: 隐私条款检测流程图

下面将讲述 BERT 分类模型的构建过程。由于 BERT 模型允许输入最大文本长度是 512，首先对 150 篇隐私政策中待分类的语句段进行统计，字数最长为 497 字，最短为 5 字，平均长度为 163 字，符合 BERT 分类的条件。

(1) 模型设计

BERT 分类器的输入是已经分割好的语句段，输出是该语句段对应的隐私条款类别。分类过程如图 4.13所示，首先通过 Tokensizer 将输入的语句段切分成一个字符列表 Tokens，将 Tokens= ([CLS],Tok_1, Tok_2,...,Tok_n，[SEP]) 作为 BERT 模

型的输入,其中特殊字符 [CLS] 添加在序列的开头，用于学习整个语句的语义表示；特殊字符 [SEP] 添加在序列的结尾，用于标识语句段的结束。然后这些字符通过 BERT 的嵌入层得到一个嵌入向量，嵌入层的作用是将每个字转换为固定维的向量表示形式。Tokens 一共会经过三个嵌入层，分别得到表示当前字的向量 E_W，表示当前字位置的位置向量 E_P，表示上下句信息的向量 E_S，最后将 3 个向量的结果直接相加，得到了嵌入向量 $E=(E_{[CLS]},E_1,E_2,E_{x_n},...,E_{[SEP]})$ 之后，将向量 E 输入多层 Transfermer 编码层，得到输出向量 $T=(T_{[CLS]}, T_1, T_2, T_n,..., T_{[SEP]})$，对于文本分类任务，首个字符 [CLS] 的输出向量能够更公平地融入其他字的语义信息，因此选用它来做为表征整个句子语义的向量。

最后将句向量 $T_{[CLS]}$ 输入给一个线性分类器，线性分类器的输出隐藏向量 $H = (h_1,h_2,...,h_n)$，经过 softmax 后得到当前文本属于某一个类别的概率列表 $P = (p_1, p_2,...,p_n)$，其中 m 代表着该主题下隐私条款的数量，p_i 代表当前语句段属于第 i 个指标的概率。softmax 公式可以表达如下：

$$p_i(h) = \frac{\exp(h_i)}{\sum_{j=1}^{m} \exp\left(h_j\right)}$$

(2) 模型实现

本文使用了 HuggingFace 提供的 Tranformers 库，利用它可以加载社区里的预训练模型，只需要关注下游任务的微调即可。预训练模型选取了 Chinese-BERT-wwm 模型 [33]，该模型网络结构共有 12 层，隐藏层包含 768 维，采用 12 头注意力机制，共包含 110M 个参数。wwm 的意思是 Whole Word Masking（全词掩盖），传统 BERT 只对单个 token 进行掩盖，而中文和英文的 token 不同，英文最小的 token 是一个单词，而中文最小的 token 是一个字，中文的词由多个字组成，包含了更多信息，全词 mask 能够对词进行了掩盖，效果优于对单个字进行掩盖。

接着需要将对预训练模型进行微调以适合本系统的分类任务，模型的微调策略代码如图 4.14所示，在 BertClassifyModel 也中先定义预处理模型和分词器。Linear 层包含两个参数，第一个参数 768 是来自于 Chinese-BERT-wwm 隐藏层的数量，第二个参数是输出的类别数，这个根据不同主题下隐私条款的数量进行设置，如对“收集与使用”训练分类器时，类别数设置为 6。forward 中需要先得到 input_ids 和 attention_mask，input_ids 表示输入的字符在模型字典中的 ID，attention_mask 用来告诉模型哪些词是不用处理的填充符。表征句子语义的向量选取了输出序列第一个位置的 [CLS] 向量。由于涉及到多分类，因此本系统在 [CLS] 向量后接了一个 Linear 层用于分类。

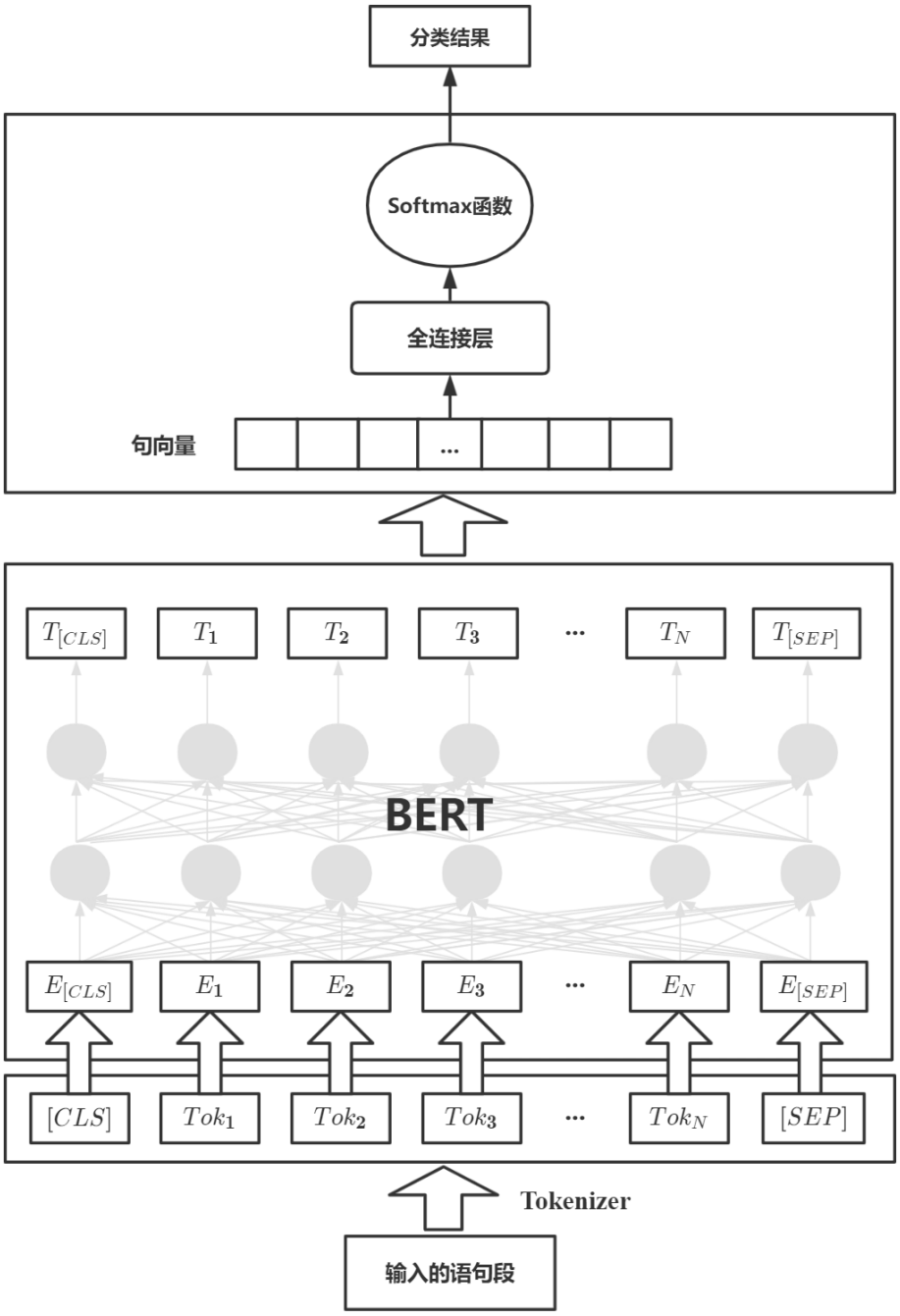

图 4.13: BERT 分类流程

模型效果通过交叉熵损失函数进行评估, 交叉熵损失函数可以衡量真实的隐私条款分布与模型预测的隐私条款分布之间的差异。当模型分类效果不足时, 模型的误差 loss 就会越大, 当模型分类效果很好时, 模型的误差 loss 就会降低。其

```
class BertClassifyModel(nn.Module, pretrained="hfl/chinese-bert-wwm-ext", num_labels):
    def __init__(self,nums):
        super(BertClassifyModel,self).__init__()
        self.model = BertModel.from_pretrained(pretrained)
        self.tokenizer = BertTokenizer.from_pretrained(pretrained)
        self.classifier = nn.Linear(768, num_labels)

    def forward(self,text_list):
        batch_tokenized = self.tokenizer.batch_encode_plus(text_list, add_special_tokens=True,
                                    max_length=512, padding = 'max_length')
        input_ids = torch.tensor(batch_tokenized['input_ids'])
        attention_mask = torch.tensor(batch_tokenized['attention_mask'])
        bert_output = self.model(input_ids,attention_mask=attention_mask)
        cls_output = bert_output[0][:,0,:]
        output = self.classifier(cls_output)
        return output
```

图 4.14: BERT 模型微调代码

计算公式如下:

$$\text{loss} = -\sum_{j=1}^{m} y_j \log p_j$$

其中 p 分别当前语句段属于某一个隐私条款类别的概率，y 表示该语句段的真实类别，m 表示隐私条款类别数。

(4) 训练集与测试集

使用文本处理模块和章节标题分类器对 4.2 小节中收集的 150 款 APP 的隐私政策进行处理，获取“收集与使用”、“存储与保护”和“共享与披露”这三个主题下的语句段，然后根据隐私政策检测指标中的隐私条款对其进行人工标注，没找到对应隐私条款的标注为“其他”。然后按 8:2 的比例划分训练集与测试集。

(3) 模型最优参数

结合相关研究 [34] 和多轮实验测试，选取了 BERT 模型参数的最优值如表 4.5所示，三个主题下的 BERT 模型在训练时均选用了这些参数。

4.5.3 金融专项检测的设计与实现

《常见类型移动互联网应用程序必要个人信息范围规定》将金融 APP 分为了三类，分别是网络借贷类、金融理财类以及手机银行类，这三类 APP 分别规定了必要个人信息的收集范围，如表 5.11所示。

金融专项检测的功能，就是根据金融 APP 的类型，判断隐私政策中收集的个人信息是否超出了该类金融 APP 的必要范围。因为隐私条款检测已经对文本

表 4.5: BERT 模型最优参数

参数名称	参数含义	最优值
batch_size	批训练样本数	32
epochs	迭代次数	4
learning_rate	学习率	2e-5
Dropout	神经元随机失活概率	0.3

表 4.6: 必要个人信息收集范围

类型	必要个人信息
网络借贷类	用户移动电话号码 用户姓名 用户证件的类型、号码及有效期限 用户银行卡号码
金融理财类	用户移动电话号码 用户姓名 用户证件的类型、号码及有效期限 用户证件影印件 用户资金账户 用户银行卡号码或支付账号
手机银行类	用户移动电话号码 用户姓名 用户证件的类型、号码及有效期限 用户证件影印件 用户资金账户 用户银行卡号码 用户银行预留移动电话号码 转账收款人的姓名 转账收款人的银行卡号码及开户银行信息

内容进行了分类，因此金融专项检测可以直接获取类别为“收集的个人信息类型”的内容。然后对这部分内容进行一个二分类的判断，判断个人信息是否为必要收集的。

因此，本系统为这三类金融 APP 分别设计了对应的 BERT 二分类模型判断输入的语句是否为必要收集的个人信息。模型的输入是隐私条款检测中分类结果为“收集的个人信息”的语句，这些语句表明了隐私政策中声明收集的个人信息。与上一节相同，同样先将输入的语句通过 Tokensizer 切分成一个字符列表 Tokens，并在前面个加上特殊字符 [CLS]，用来学习整个语句的语义表示，BERT 模型同样会为输入语句的生成一个向量 T=($T_{[CLS]}$, T_1, T_2, T_n,..., $T_{[SEP]}$)，其

中 $T_{[CLS]}$ 用表征句向量，然后 $T_{[CLS]}$ 经过线性分类器和 softmax 函数，输出结果是一个二分类标签 y，其中 y_1 表示语句中收集的个人信息是必要的，y_0 表示语句中收集的个人信息是非必要的。

4.5.4 时序图

文本合规性检测模块整体流程的时序图如图 4.15所示，主进程调用文本处理服务的标题提取功能，获取该任务上传的隐私政策文本的章节标题，然后调用分类服务进行章节标题的分类，分类服务会先通过文本处理模块进行中文分词和去停用词。之后，分类服务会调用已经训练好的 SVM 分类器进行标题分类，分类完成后会将打好主题标签的标题返回给主进程，主进程再调用文本处理服务获取每个主题对应的文本内容，然后再调用分类服务，分类服务会根据每个主题的特性，选择关键词匹配的方法进行分类或调用对应主题的 BERT 分类器对内容进行分类。分类完成后，会将分类结果进行统计，记录每个隐私条款对应的语句段。然后将“收集的个人信息类型”类别的语句段传给分类服务，分类服务会根据金融 APP 的类型，调用对应的 BERT 二分类器，判断这些个人信息是不是都符合必要收集的范围，并返回检测结果给主进程。

完成所有检测操作，主进程会将所有检测的结果信息生成 PrivacyResult 模型类，存储到 MySQL 中。

4.6 权限合规性检测模块

权限合规性检测模块是为了判断隐私政策中声明的敏感权限是否与 APP 实际申请的敏感权限一致。

4.6.1 构造敏感权限名称表

权限合规性检测需要构造敏感权限名称表，包含了权限名和它对应的标准描述。首先通过 Android 官网查阅具体的敏感权限，共包含了 10 个权限组以及 27 个权限名。然后阅读官网对权限的解释，并结合隐私政策中对设备权限的相关描述，总结了每个敏感权限对应的标准描述。构造的权限名称表如 4.7所示

4.6.2 权限语句集提取方法

(1)APP 实际使用的敏感权限

APP 可以通过动态分析和静态分析进行检测，静态分析不需要程序运行就能进行，执行效率高；动态分析需要在程序运行期间进行检测，检测的准确率更高，但检测速度慢。考虑到本系统只需要提取出 APP 所申请的敏感权限，不需要运行程序就能完成，因此选择了静态分析的方法。

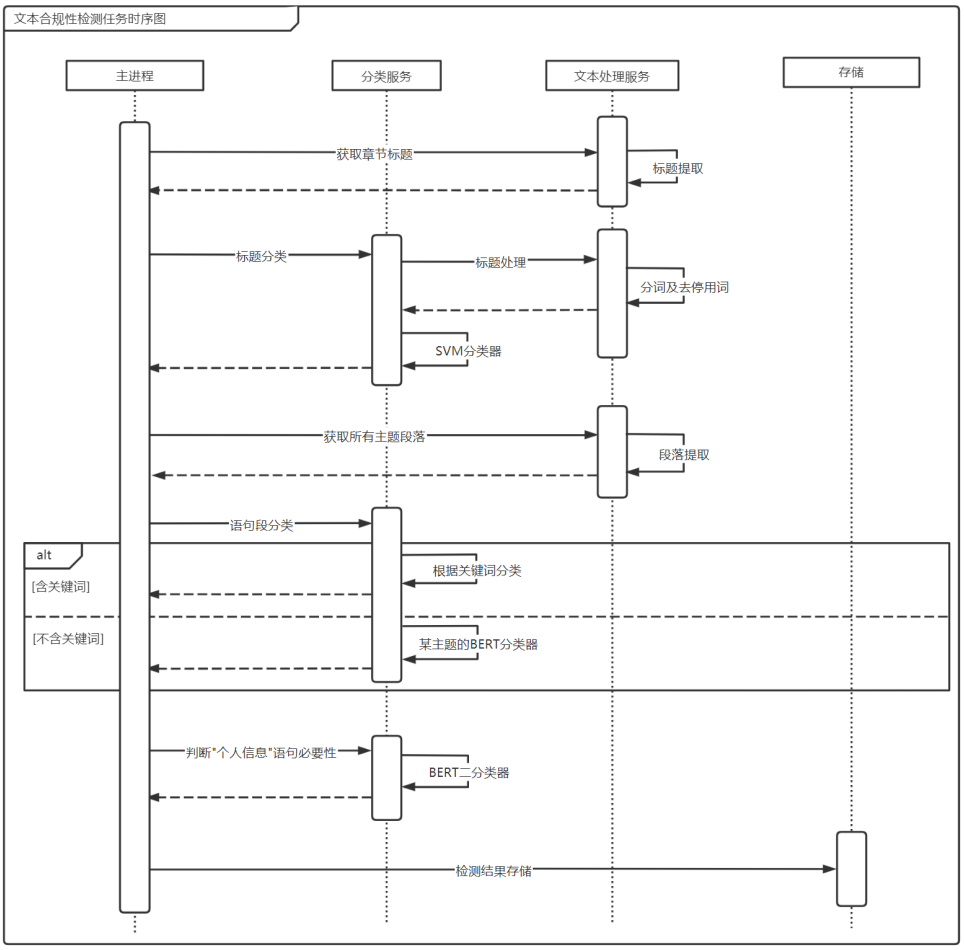

图 4.15: 文本合规性检测模块时序图

本系统使用了 Python 的 Androguard 库来解析 APK 文件。通过 Androguard 的 AnalyzeAPK 方法可以解析得到三个对象是 a，d 和 dx，分别表示 APK 对象、对象数组 DalvikVMFormat 和 Analysis 对象。通过 a 对象，就可以找到有关 APK 的所有信息了，例如包名称、权限、AndroidManifest.xml 或其资源。这里调用 a 对象的 get_permisssion 方法可以获取该 APP 所有申请的权限，然后将所有申请的权限与 27 个敏感权限进行一一比较，即可获得该 APP 所申请的敏感权限。APP 敏感权限的提取代码如图 4.16所示。

(2) 隐私政策中声明使用的设备权限

在隐私条款检测指标中的“收集与使用”主题下有一个名为“申请的设备权限”的隐私条款，上一节中为“收集与使用”、“存储与保护”、“共享与披露”这三个主题分别设计了 BERT 分类器，这里可以使用“收集与使用”中的 BERT 分类器，来获取分类结果为“申请的设备权限”的语句段。因为一个语句段里面

表 4.7: 权限名称表

权限组	权限名	标准描述
CALENDAR	READ_CALENDAR	允许读取用户日历数据
	WRITE_CALENDAR	允许写入用户日历数据
CAMERA	CAMERA	允许使用摄像头
CALL_LOG	READ_CONTACTS	允许读取用户通讯录
	WRITE_CONTACTS	允许写入用户通讯录
	GET_ACCOUNTS	允许从服务账户中获取账户列表
CONTACTS	READ_CALL_LOG	允许读取用户通话记录
	WRITE_CALL_LOG	允许写入用户通话记录
	PROCESS_OUTGOING_CALLS	允许查看在拨出电话期间拨打的号码
LOCATION	ACCESS_FINE_LOCATION	允许获取基于 GPS 等的精准地理位置
	ACCESS_COARSE_LOCATION	允许获取基于 IP 等粗略的地理位置
MICROPHONE	RECORD_AUDIO	允许使用麦克风进行录音
PHONE	READ_PHONE_STATE	允许获取 IMSI、IMEI 等设备唯一标志信息，以及手机通话状态
	READ_PHONE_NUMBERS	允许读取用户的本机电话号码
	CALL_PHONE	允许直接拨打电话
	ANSWER_PHONE_CALL	允许接听拨入的电话
	ADD_VOICEMAIL	允许向邮件中添加语音附件
	USE_SIP	允许拨打或接听 SIP 网络电话
	ACCEPT_HANDOVER	允许继续进行在其他 APP 中发起的通话
SENSORS	BODY_SENSORS	允许访问身体内部状况相关的传感器数据
SMS	SEND_SMS	允许发送短信
	RECEIVE_SMS	允许接受短信
	READ_SMS	允许读取短信或彩信内容
	RECEIVE_WAP_PUSH	允许接受 WAP 推送信息
	RECEIVE_MMS	允许接受彩信
STORAGE	READ_EXTERNAL_STORAGE	允许读取外置存储
	WRITE_EXTERNAL_STORAGE	允许写入外置存储

```
def extract_danger_permission(filename):
    a, d, dx = AnalyzeAPK(filename)
    permissions = []
    for p in a.get_permissions():
        p = p.split('.')[-1]
        if p in danger_permission:
            permissions.append(p)
    return permissions
```

图 4.16: APP 权限提取代码

可能包含了多个权限相关的描述，因此这里进行了分句处理，用逗号为最小的切割符号进行分割，这样就可获取该隐私政策的权限短句集。

4.6.3 基于 SimCSE 的权限合规性检测方法

考虑到敏感权限的类型多达 27 种，每个隐私政策文本中涉及的权限短语又众多，如果采用有监督的分类方法，人工打标签的方式耗时较长。因此本文决定采用基于无监督语义相似度匹配的方法进行检测。

设从 APP 中提取的敏感权限标准描述集合 $A=[a_1, a_2, a_3, a_4,...,a_n]$，n 代表集合 A 中的语句总数；设从隐私政策中提取的权限短句集 $B=[b_1, b_2, b_3,b_4,...,b_m]$，m 代表集合 B 中的语句总数。从 A 集合中选取一个句子 a_i 与 B 集合中的所有句子进行余弦相似度计算，并设置好一个阈值，如果 B 中存在一个句子 b_j 与 a_i 的余弦相似度大于这个阈值，则这两个句子相似，也就说明 APP 中使用的权限在隐私政策中有相应描述。一共进行 n 次检测，若 A 集合中的语句都能在 B 集合中找到相似语句，则证明了该 APP 的权限合规。

因为需要余弦相似度计算，所以需要获取表征句子语义的向量，本文使用了无监督 SimCSE [25] 进行句向量的获取。检测的整体流程如图 4.17所示。

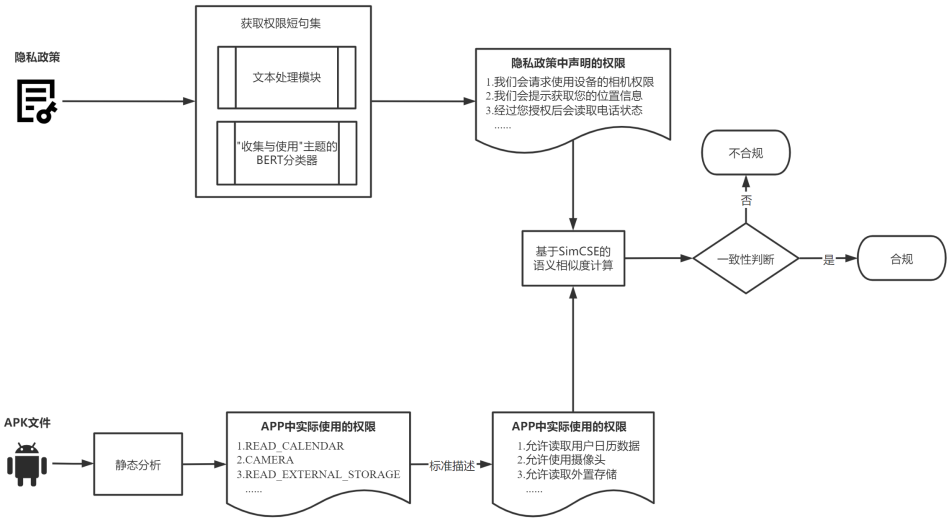

图 4.17: 权限合规性检测流程图

下面对 SimCSE 模型的构建进行讲解。

SimCSE 是一个基于对比学习的模型，对比学习需要构建正例，SimCSE 通过将一个句子通过两次不同的 dropout 得到两个句向量互相作为整正例，而同一个 batch 中的其他句向量做为负例。SimCSE 使用了 BERT 的编码器，在训练

过程中的 dropout mask 都是随机的，因此不需要做其他复杂的工作，只需要将 batch 中的每个句子复制两份即可。正反例的构建代码如图 4.18所示。

```
class CSECollator(object):
    def __init__(self, tokenizer, feature_list=("input_ids", "attention_mask", "token_type_ids")):
        self.tokenizer = tokenizer
        self.feature_list = feature_list

    def collate(self, batch):
        new_batch = []
        for example in batch:
            for i in range(2):
                new_batch.append({fea: example[feature] for feature in self.feature_list})
        new_batch = self.tokenizer.pad(
            new_batch,
            max_length=50,
            padding = 'max_length',
            return_tensors="pt"
        )
        return new_batch
```

图 4.18: 正反例构建代码

接下来需要构建 SimCSE 模型，这里的预处理模型选取了 chinese-BERT-wwm-ext 做为预处理模型，池化方式的选择使用了 encoder 的最后一层的 [CLS] 向量，dropout 的概率选取了 0.3，模型的构建代码如图 4.19所示。

```
class SimCSEModel(nn.Module):
    def __init__(self, pretrained="hfl/chinese-bert-wwm-ext", dropout_prob=0.3):
        super().__init__()
        conf = BertConfig.from_pretrained(pretrained)
        conf.attention_probs_dropout_prob = dropout_prob
        conf.hidden_dropout_prob = dropout_prob
        self.encoder = BertModel.from_pretrained(pretrained, config=conf)

    def forward(self, input_ids, attention_mask, token_type_ids):
        output = self.encoder(input_ids,
                              attention_mask=attention_mask,
                              token_type_ids=token_type_ids)
        output = output.last_hidden_state[:, 0]
        return output
```

图 4.19: SimCSE 模型构建代码

模型需要训练的 loss 函数为

$$\ell_i = -\log \frac{e^{\operatorname{sim}\left(\mathbf{h}_i^{z_i}, \mathbf{h}_i^{z_i'}\right)/\tau}}{\sum_{j=1}^{N} e^{\operatorname{sim}\left(\mathbf{h}_i^{z_i}, \mathbf{h}_j^{z_j'}\right)/\tau}} \tag{4.1}$$

该函数的意义在于让模型拉近正样本距离的同时能够远离负样本。loss 函数实现的代码如图 4.20所示，与论文中的公式保持一致，这里使用了 torch 中自带的 cosine_similarity 来快速实现句子相似度的计算，其中 tao 代表温度超参 τ，取值为 0.05。

```
def compute_loss(y_pred, tao=0.05, device="cuda"):
    idxs = torch.arange(0, y_pred.shape[0], device=device)
    # 生成正例的 idx
    y_true = idxs + 1 - idxs % 2 * 2
    # 计算各句子相似度，生成 similarities 矩阵
    similarities = F.cosine_similarity(y_pred.unsqueeze(1), y_pred.unsqueeze(0), dim=2)
    # 屏蔽对角矩阵
    similarities = similarities - torch.eye(y_pred.shape[0], device=device) * 1e12
    # 除以论文中提到的温度参数 τ
    similarities = similarities / tao
    loss = F.cross_entropy(similarities, y_true)
    return torch.mean(loss)
```

图 4.20: loss 函数构建代码

结合了设备性能以及多次实验测试的结果，选取了 SimCSE 模型参数的最优值如表 4.8所示。

表 4.8: SimCSE 模型最优参数

参数名称	参数含义	最优值
batch_size	批训练样本数	64
epochs	迭代次数	16
learning_rate	学习率	1e-5
Dropout	神经元随机失活概率	0.3

4.6.4 时序图

权限合规性检测任务的时序图如图 4.21所示，主进程先通过文本处理服务和主题分类器，获取主题为“收集与使用”的标题，然后再通过文本处理和“收集与使用”主题的分类器，获取所有分类结果为“使用的设备权限”的语句段，将这些语句段按逗号进行切割，获取“设备权限”短语集, 代表了隐私政策中声

明的权限。接着主进程再调用 APK 解析服务，通过静态分析的方法获取 APP 中实际使用的权限，并通过权限名称表将这些权限名转成标准描述。然后主进程调用分类服务中的 SimCSE 模型，进行语义相似度计算。

检测结束后，主进程会将检测的结果信息生成 PermissionResult 模型类，存储到 MySQL 中。

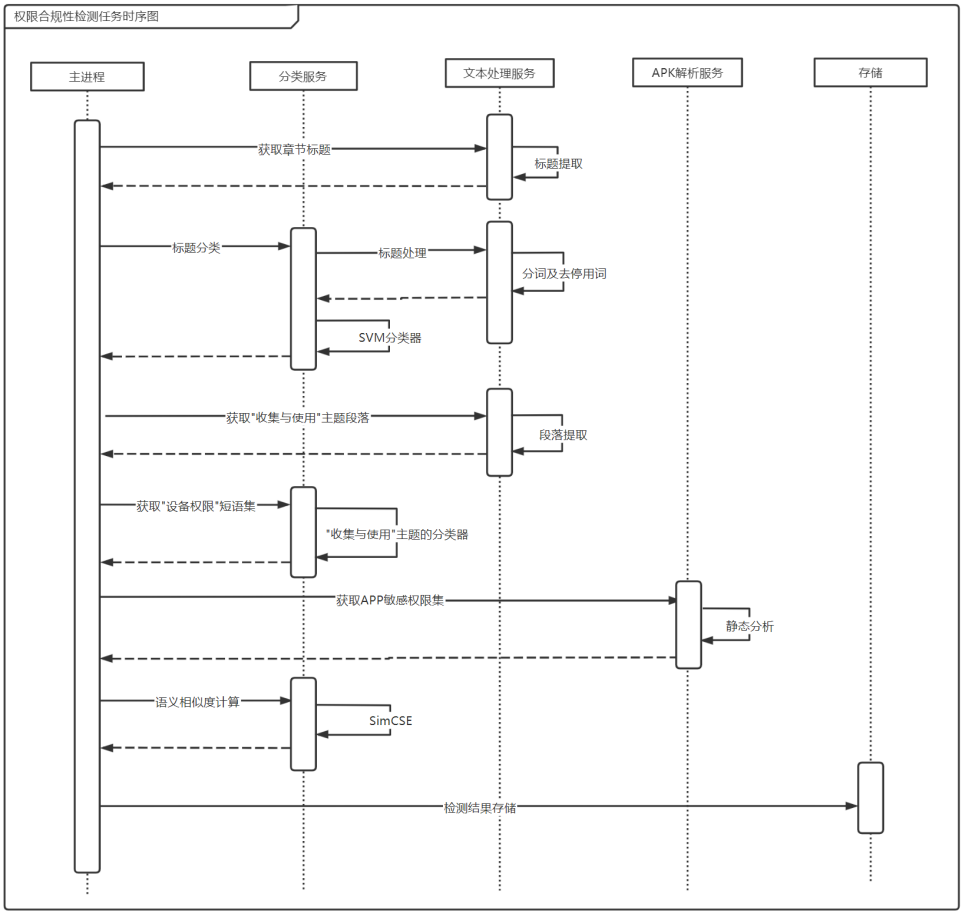

图 4.21: 文本合规性检测模块时序列图

4.7 结果生成模块

结果生成模块在用户进行任务“查看”操作的时候，会根据任务的类型，将任务的结果信息反馈给前端，同时还给文本合规性检测任务提供了检测报告供用户下载。

4.7.1 时序图

结果生成模块的时序图如图 4.22所示。具体功能通过以下方法进行实现

PrivacyResultShow 负责展示文本合规性检测任务的结果信息。PrivacyRe-

sultShow 从前端的 HttpRequest 请求中解析该任务的主键 id，根据这个 id 从数据库中获取对应 PrivacyResult 模型类的信息，再将获取的结果数据通过 JsonResponse 返回给前端。

PermissonResultShow 负责展示权限合规性检测任务的结果信息。PermissonResultShow 从前端的 HttpRequest 请求中解析该任务的主键 ID，根据这个 id 从数据库中获取对应 PermissonResult 模型类的外键，再将获取的结果数据再通过 JsonResponse 返回给前端。

ReportGenerate 可以为文本合规性检测任务生成检测报告。ReportGenerate 从前端的 HttpRequest 请求中解析任务的主键 id，根据这个 id 从数据库中获取对应 PrivacyResult 模型类的信息，之后调用 python 的 docxtpl 库并使用自定的结果报告模板，生成结果报告的 word 文件，然后将 word 转换成 PDF 格式，并通过 FileResponse 发送给前端进行下载。

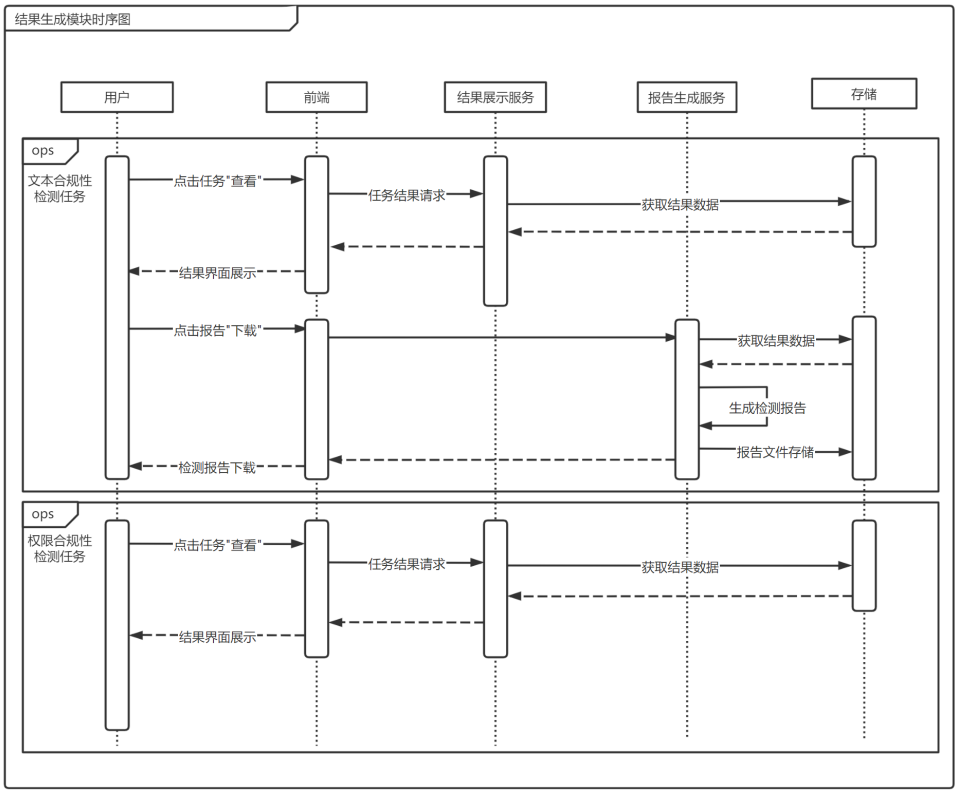

图 4.22: 结果生成模块时序图

4.7.2 界面展示

图 4.23展示了“东方财富”APP 文本合规性检测任务的结果界面，用户通过点击“查看”按钮进入任务的结果界面。

“任务信息”主要显示该任务的任务名称，APP 名称，执行耗时以及完成时间。“合规性检测”首先展示了该隐私政策的章节标题中没有描述的主题，从图中可以看出东方财富 APP 的隐私政策缺少了描述”联系方式”这一主题的标题。接着展示了隐私政策缺少哪些隐私条款，从图中可以看出东方财富 APP 的隐私政策文本中缺乏关于“公司的联系方式”，“如何删除个人信息，“个人信息安全保护措施”的描述，前面红字是该隐私条款所属的主题。最后提供了检测报告给用户下载，检测报告为 pdf 类型。金融专项检测首先展示了创建任务时，用户输入的 APP 类型，然后对该 APP 是否收集了非必要个人信息进行了说明，从图中可以看到该 APP 收集了非必要的个人信息。

由于文本合规性检测任务涉及到的内容较多，结果展示页面仅仅提供了概括性的展示，用户可以通过下载检测报告查看详细情况，包括整改意见，具体标题对应的主题，每项隐私条款的符合情况等，检测报告如图 4.24所示。

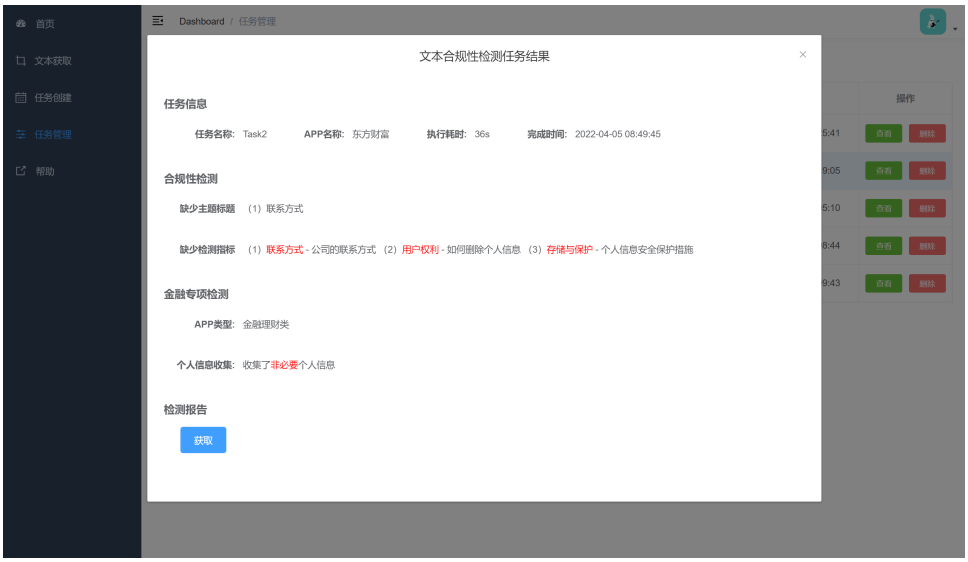

图 4.23: 文本合规性检测任务结果界面

图 4.25展示了“富途牛牛”APP 的权限合规性检测任务的结果展示界面，通过点击“查看”按钮进入任务的结果界面。

“任务信息”部分同文本合规性检测任务一样，显示了任务名称，APP 名称，执行耗时以及完成时间。“权限合规性检测”通过表格的形式进行了展示，“实际使用权限”是从 APK 文件中通过静态分析提取的权限，“政策声明权限”是从隐私政策中提取出权限，相关的权限放在了同一行，方便用户观看。最后进行了结果说明，总结了隐私政策没有提及却在 APP 中使用的权限。从图中可以看到，该 APP 一共使用了五项敏感权限，其中访问精准位置这一敏感权限未在隐私政

隐私政策检测报告

任务名称：task2　应用名称：东方财富

✧ 修改建议

[联系方式]主题在标题中未体现，未说明[提供公司的联系方式]；[存储与保护]主题撰写不全，未说明[发生安全事故处理方式]；[用户权利]主题撰写不全，未说明[如何注销账户]；隐私政策中收集了[非必要个人信息]。

✧ 标题主题

标题	对应主题
我们如何收集信息	收集与使用
我们如何使用 Cookie、Beacon、网络通道等技术	使用 cookie 等同类技术
我们如何存储和保护信息	存储与保护
我们如何使用信息	收集与使用
我们如何对外提供信息	共享与披露
您如何访问和管理自己的信息	用户权利
我们如何保护未成年人的信息	未成年人特殊保护
本指引的修订及更新	通知和修订
其他事宜	其他
无	联系方式

✧ 检测指标

检测指标	状态
运营商情况	符合
政策适用的产品或服务	符合
政策生效及更新时间	符合
收集和使用个人信息的目的	符合
收集和使用个人信息的方式	符合
收集的个人信息类型	符合
申请的设备权限	符合
个人信息存储地点和期限	符合
个人信息安全保护措施	不符合
发生安全事故处理方式	符合
共享个人信息的目的	符合

图 4.24: 检测报告

策中进行声明。

4.8 本章小节

本章主要对面向金融 APP 的隐私政策合规性检测系统进行了实现细节的详细描述，首先介绍了数据交互模块的时序图和界面截图，然后对文本检测模块的功能进行了讲解并提供了核心代码，接着对两个核心模块文本合规性检测模

图 4.25: 权限合规性检测任务结果界面

块以及权限合规性检测模块的整体设计、算法实现以及时序图进行了介绍，最后对结果生成模块进行了讲解，介绍了它的时序图和界面截图。

第五章　系统测试与案例分析

本章节先介绍了测试的目标和测试环境，然后依次对算法性能、真实案例、功能需求和非功能需求进行了测试。

5.1　测试准备

5.1.1　测试目标

本文对面向金融 APP 的隐私政策合规性检测系统主要进行四个方面的测试，首先是算法性能测试，测试了基于 SVM 的章节分类器的准确率；测试了三个主题下的 BERT 模型的分类准确率，并与其他几种分类模型进行了对比；测试了三种类型金融 APP 的 BERT 二分类模型的分类准确率，并与其它几种分类模型进行了对比；测试了无监督 SimCSE 模型在语义相似度检测上的准确度，并与 BERT-whitening 模型进行了对比。接着对实际案例进行了分析，共从三个类型中选取了 9 款金融 APP 进行文本合规性以及权限合规性的检测。最后对系统进行了功能测试和非功能测试，功能测试主要是针对第三章中描述的系统测试用例，非功能测试主要进行了响应时间的测试。

5.1.2　测试环境

本系统的测试环境如表 5.1所示。系统部署在 16G 内存的物理服务器上，带有显存为 16G 的 NVIDIA30 系显卡，为深度学习的运行提供计算资源。系统的后端采用 Django 框架，前端采用 Vue 框架。系统的数据库采用 MySQL。由于项目属于 Web 项目，使用 chrome 浏览器进行测试。

表 5.1: 测试环境

类型	测试环境
服务器	16G 内存, NVIDIA Geforce GTX 3060 12G
操作系统	Windows 10
数据库	MySQL 5.8
Python 库	Python 3.8, Django 3.2.5, torch 1.8.0, Transformers4.1.5
浏览器	chrome
客户端	Node.js 16.14.2, Vue 2.6.10, Element Ui

5.2 算法性能测试

5.2.1 基于 SVM 的标题分类器性能测试

选取了 10 款金融 APP 共计 103 个标题做为测试集，成功预测了 100 个标题对应的主题，总体准确率为 97.1%，具体预测情况如表 5.2所示。

表 5.2: 标题分类器测试结果

类别	标题总数	预测成功数	准确率
收集与使用	17	17	100%
存储与保护	11	11	100%
共享与披露	13	12	92.3%
使用 cookie 等同类技术	9	9	100%
未成年人特殊保护	11	11	100%
用户权利	11	11	100%
通知和修订	12	11	91.7%
联系方式	11	11	100%
其他	8	7	87.5%
共计	103	100	97.1%

5.2.2 隐私条款检测中的 BERT 模型性能测试

1. 验证集

选取了 15 款金融 APP 的隐私政策文本，然后使用文本处理模块和章节标题分类器，给每一个标题确定了一个主题，之后按照主题汇总了对应标题下的段落内容。然后人工对“收集与使用”、“存储与保护”和“共享与披露”这三个主题下的语句段对应的隐私条款进行标注。考虑到每个主题的文本中除了阐述跟隐私条款相关的内容外，还有一些与隐私条款无关的内容，因此加入了一个“其他”的类别。

对于“收集和使用”主题的内容，标注了“收集和使用个人信息的目的”、“收集和使用个人信息的方式”、“收集的个人信息类型”、“申请的设备权限”、“信息的第三方获取”和“其他”六个类别。

对于“存储与保护”主题下的内容，标注了“个人信息存储地点和期限”、“个人信息安全保护措施”、“发生安全事故处理方式”和“其他”四个类别。

对于“共享与披露”主题下的内容，共包含了“共享个人信息的目的”、“共享的个人信息类型”、“接收方的类型或身份”、“征得授权同意的例外”、“个人信息跨境传输的说明”和“其他”六个类别。

三个主题下每个隐私条款的标注数量如表 5.3所示。

表 5.3: 隐私条款检测的验证集

数据集主题	分类类型	数量
收集与使用	收集和使用个人信息的目的	98
	收集和使用个人信息的方式	119
	收集的个人信息类型	163
	信息的第三方获取	172
	申请的设备权限	131
	其他	225
	共计	**913**
存储与保护	个人信息存储地点和期限	39
	个人信息安全保护措施	62
	发生安全事故处理方式	29
	其他	46
	共计	**176**
共享与披露	共享个人信息的目的	76
	共享的个人信息类型	24
	接收方的类型或身份	38
	征得授权同意的例外	56
	个人信息跨境传输的说明	13
	其他	87
	共计	**294**

2. 测试指标

对于多分类任务，最直接的评价指标为考虑全类别的预测准确率，准确率 = 预测正确的数量 / 总数量。

3. 测试结果

为了证明模型的效果，采用了 3 个模型进行对比实验：

fastText [35]: 由 Facebook 提出的一款机器学习训练工具，将文本的词及 n-gram 向量叠加平均得到文本向量, 然后使用该向量做 softmax 多分类。

TextCNN [36]: 将卷积神经网络 CNN 应用到文本分类任务。

TextRNN [37]: 利用循环神经网络 RNN [38] 或一些变种比如 LSTM [39]、GRU [40] 等解决文本分类问题。

表 5.4展示了合同分类算法的实验结果，从准确率来看，BERT 在“收集与使用”、“存储与保护”和“共享与披露”这三个主题下的分类表现均明显优于其他模型。

表 5.4: 隐私条款检测实验结果

准确率 主题 / 模型	收集和使用	存储与保护	共享与披露
fastText	0.783	0.794	0.782
TextCNN	0.865	0.879	0.877
TextRNN	0.874	0.868	0.872
BERT	**0.918**	**0.923**	**0.917**

5.2.3 金融专项检测中的 BERT 模型性能测试

1. 验证集选取了 20 款金融理财类、20 款手机银行类以及 20 款网络借贷类的隐私政策里面所有与个人信息收集相关语句作为测试集，参考《常见类型移动互联网应用程序必要个人信息范围规定》的要求对其进行了标注，具体如表 5.6所示。

表 5.5: 金融专项检测的验证集

APP 类型	必要个人信息语句	非必要个人信息语句	共计
金融理财类	124	83	207
手机银行类	142	117	259
网络借贷类	138	105	243

2. 评价指标

对收集的个人信息是否必要的分类可以是一个二分类问题，本文对隐私数据泄露风险检测效果的评价指标包括：准确率、精确率、召回率以及 F1-Score，下面对以上指标进行简要介绍。

TP: 把必要个人信息判断为必要的数目。

FN: 把必要个人信息判断为非必要的数目。

FP: 把非必要个人信息判断为必要的数目。

TN: 把非必要个人信息判断为非必要数目。

准确率：Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$ 。

精确率: Precision = $\frac{TP}{TP+FP}$ 。

召回率：Recall = $\frac{TP}{TP+FN}$ 。

F1-Score：$F1 = \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$ 。

3. 测试结果

为了证明模型的效果，采用了 TextCNN 和 TextRNN 两个模型进行对比实验，实验结果如表 5.6所示。从表中可以看到，使用 BERT 模型的分类效果在三

表 5.6: 金融专项检测实验结果

APP 类型	模型名称	准确率	精确率	召回率	F1
金融理财类	TextCNN	0.889	0.917	0.887	0.901
	TextRNN	0.908	0.941	0.895	0.917
	BERT	0.935	0.958	0.927	0.943
手机银行类	TextCNN	0.919	0.923	0.929	0.926
	TextRNN	0.896	0.908	0.901	0.904
	BERT	0.934	0.943	0.936	0.94
网络借贷类	TextCNN	0.914	0.927	0.92	0.924
	TextRNN	0.905	0.905	0.934	0.918
	BERT	0.93	0.929	0.95	0.939

个金融类别和四个评价指标上的表现均好于其他模型。

5.2.4 SimCSE 模型性能测试

1. 验证集

从隐私政策文本中，为 27 种敏感权限分别挑选了 10 条相关描述的句子做为验证集，共 270 条语句，每条语句都标注了它们对应的敏感权限类型。部分验证集的语句如表 5.7所示。

表 5.7: 权限合规性检测的验证集（部分）

政策中的敏感权限语句	敏感权限类型
经过您授权后会读取电话状态	READ_PHONE_STATE
我们会请求使用设备的短信发送功能	SEND_SMS
我们会向您请求面容解锁的权限	CAMERA
我们会请求使用设备的麦克风权限	READ_PHONE_STATE
我们会请求使用设备的相机权限	CAMERA
若您拒绝易淘金调用您的摄像头权限	CAMERA
我们会以弹窗形式请求获取你设备的日历读取权限	READ_CALENDAR
我们会提示获取您的位置信息	ACCESS_FINE_LOCATION
此环节需要获取您文件写入外部存储的权限	WRITE_EXTERNAL_STORAGE
.....	.....

2. 测试指标

该实验类似于一个信息检索问题，输入某个敏感权限的标准语句和验证集，

通过 SimCSE 模型计算句向量并从验证集中找出所有与标准语句达到余弦相似性阈值的句子，这里使用了查准率与查全率以及 F1 做为评价指标。

查准率 P ＝正确找到的句子 / 满足余弦相似度阈值的句子

查全率 R ＝正确找到的句子 / 验证集中所有该标签的句子

$F1 = \frac{2 \times P \times R}{P+R}$

3. 测试结果

参照实验选取了 BERT-whitening 模型。因为有 27 个敏感权限，所以进行了 27 次实验，记录了三个指标的平均值作为最终结果。选取不同的余弦相似度阈值也会影响查重率和查准率，阈值越高，查准率越高，查全率越低，这里通过对选取多个不同的阈值在验证集上测试，为每个模型选取了 F1 值最高时的阈值，其中 BERT-whitening 模型表现最好时余弦相似度阈值取 0.87，SimCSE 模型表现最好时阈值取 0.71。

实验结果如表 5.8所示，SimCSE 在查准率，查全率和 F1 三项评价指标的表现上都好于 BERT-whitening。

表 5.8: 权限合规性检测实验结果

模型	查准率	查全率	F1
BERT-whitening	0.857	0.928	0.891
SimCSE	0.924	0.961	0.942

5.3 真实案例分析

本节从金融理财类、手机银行类以及网络借贷类 3 个类别下选取了 3 个 APP，共计 9 个 APP 进行真实案例的分析，这 9 个 APP 的具体情况如表 5.9所示。首先获取它们的隐私政策文本和 APK 文件，然后通过系统分别对他们执行了文本合规性检测任务和权限合规性检测任务，并对每个 APP 的检测结果进行了总结和展示。

5.3.1 隐私条款检测的测试

隐私条款检测主要检测隐私政策中每个章节的标题是否覆盖了所有主题，以及隐私政策是否满足每个隐私条款。这里根据每个 APP 检测任务生成的报告将所有检测结果进行了整理，汇总后如表 5.10所示。可以看到，该系统检测出了 5 款 APP 的章节标题不完整，缺少的对应主题；8 款 APP 的隐私条款不完整。注意，如果 APP 缺少某个主题，并不表明它该主题下的隐私条款全部缺失，有

表 5.9: 待测 APP 的基本情况

类型	APP 名称	APK 大小	隐私政策更新时间
金融理财	长江 e 号	120MB	2021 年 7 月 28 日
	蜻蜓点金	136MB	2021 年 11 月 01 日
	东吴秀财	142MB	无
手机银行	重庆银行	101MB	2021 年 8 月 26 日
	浦大喜奔	145MB	2021 年 12 月 28 日
	兴业银行	201MB	2021 年 12 月 30 日
网络借贷	万达贷	28.8MB	2021 年 8 月 31 日
	太平通	108MB	2021 年 10 月 25 日
	借财童子	20.6MB	2021 年 11 月 30 日

可能会在“其他”类别里找到。

表 5.10: 隐私条款检测的测试结果

APP 名称	缺少主题	缺少隐私条款
长江 e 号	(1) 共享与披露	(1) 共享个人信息的目的 (2) 共享的个人信息类型 (3) 个人信息存储地点和期限 (4) 发生安全事故处理方式
蜻蜓点金	无	无
东吴秀财	(1) 使用 cookie 等技术 (2) 未成年特殊保护 (3) 用户权利 (4) 通知与修订 (5) 联系方式	(1) 运营商情况 (2) 政策时效 (3) 使用 cookie 等技术的目的 (4) 停用 cookie 等技术的方式 (5) 如何访问个人信息 (6) 如何修改个人信息 (7) 如何删除个人信息 (8) 改变授权同意的范围 (9) 如何注销账号
重庆银行	(1) 共享与披露 (2) 未成年人特殊保护	(1) 运营商情况 (2) 发生安全事故处理方式 (3) 个人信息存储地点和期限 (4) 个人信息跨境传输的说明 (5) 征得授权同意的例外
浦大喜奔	无	(1) 运营商情况
兴业银行	无	(1) 个人信息存储的地点和期限
万达贷	(1) 未成年人特殊保护	(1) 未成年人使用说明 (2) 运营商情况
太平通	(1) 使用 cookie 等技术	(1) 使用 cookie 等技术的目的 (2) 停用 cookie 等技术的方式 (3) 发生安全事件的处理方式
借财童子	无	(1) 个人信息跨境传输的说明 (2) 征得授权同意的例外

5.3.2 金融专项检测的测试

金融专项检测结果如表 5.11所示，根据结果可以看到，本系统共从 9 款 APP 中检测出了 8 款 APP 收集了非必要的个人敏感信息。

表 5.11: 金融专项检测的测试结果

类型	APP 名称	个人信息收集情况
金融理财	长江 e 号	收集了非必要的个人信息
	蜻蜓点金	收集了非必要的个人信息
	东吴秀财	只收集了必要的个人信息
手机银行	重庆银行	收集了非必要的个人信息
	浦大喜奔	收集了非必要的个人信息
	兴业银行	收集了非必要的个人信息
网络借贷	万达贷	收集了非必要的个人信息
	太平通	收集了非必要的个人信息
	借财童子	收集了非必要的个人信息

5.3.3 权限合规性检测的测试

这里根据每个 APP 权限合规性检测任务的结果进行了汇总，详细情况如表 5.12所示，本系统的权限合规性测试一共检测出了四款金融 APP 的权限不合规行为。

表 5.12: 权限合规性检测结果的测试结果

APP 名称	声明权限是否一致
长江 e 号	一致
蜻蜓点金	缺少 (1)ACCESS_COARSE_LOCATION 相关描述
东吴秀财	缺少 (1)CALL_PHONE (2)RECORD_AUDIO 相关描述
重庆银行	缺少 (1)WRITE_EXTERNAL_STORAGE (2)WRITE_CONTACTS 相关描述
浦大喜奔	一致
兴业银行	一致
万达贷	缺少 (1)GET_ACCOUNTS
太平通	一致
借财童子	一致

5.4 功能测试

本节根据第三章的功能需求进行分析，对系统用例进行功能测试设计，主要做法是按照用例描述的步骤对指定用例进行操作执行，以验证系统是否符合预期，是否出现功能上的异常，从而判断系统功能是否完善。

表 5.13展示了用户登录测试用例。本用例主要关注在不同条件下系统能否处理用户输入返回正确结果并显示提示信息。

表 5.13: 用户登录测试用例

测试 ID	UC1
测试名称	用户登录测试用例
测试功能	用户可以使用已授权的账号进行登录
测试步骤	1. 输入已授权用户名和正确密码进行登录； 2. 输入已授权用户名和错误密码进行登录； 3. 输入未授权用户名和任意密码进行登录
预期结果	1. 登录成功，系统跳转至主页面； 2. 登录失败，弹窗显示登录失败并提示密码错误； 3. 登录失败，弹窗显示登录失败并提示用户账号未授权。
实际结果	与预期相符合

表 5.14展示了隐私政策文本获取的测试用例。本用例主要关注系统是否能根据用户输入的网址链接成功下载隐私政策的文本，是否能够自动识别无效链接和错误链接等。

表 5.14: 隐私政策文本获取测试用例

测试 ID	UC2
测试名称	隐私政策文本获取测试用例
测试功能	用户通过 APP 的隐私政策链接下载隐私政策文本
测试步骤	1. 进入文本获取界面，输入一个可用的隐私政策链接，点击获取按钮 2. 进入文本获取界面，输入一个错误格式的链接，点击获取按钮 3. 点击下载好的文本，查看内容
预期结果	1. 弹窗显示文本获取成功，文本被自动下载到本地 2. 弹窗显示隐私政策链接无效，请输入正确链接 3. 文本能成功打开，内容与网站一致
实际结果	与预期相符合

表 5.16展示了隐私政策上传的测试用例。本用例主要关注隐私政策能否被上传，文件数量超过 1 时是否会提示，当文件类型不为 txt 时是否会被自动识别，文件上传的进度条显示等。

表 5.16展示了 APK 文件上传的测试用例。本用例主要关注 APK 文件能否被上传，文件数量超过 1 时是否会提示，当文件类型不为 APK 时是否会被自动识别，上传数据库中已经记录的 APK 文件时是否会提示，文件上传的进度条是否有显示等。

表 5.17展示了用户创建任务的测试用例，主要关注没有填写任务信息时是否会提示，文件未上传时是否会提示，任务能否被创建成功，任务创建成功后能

表 5.15: 隐私政策上传测试用例

测试 ID	UC3
测试名称	隐私政策上传测试用例
测试功能	用户上传隐私政策文件
测试步骤	1. 点击选取文件，选择一个 txt 文件，点击上传文件 2. 点击选取文件，选择一个非 txt 类型的文件，点击上传文件 3. 点击选取文件，选择一个 txt 类型的文件，再次点击选取文件，选择一个 txt 类型的文件
预期结果	1. 界面出现文件上传进度条，进度条 100% 后弹窗显示文件上传成功 2. 弹窗显示文件类型必须为 txt 类型 3. 弹窗显示只能选取一个文件上传
实际结果	与预期相符合

表 5.16: APK 文件上传测试用例

测试 ID	UC4
测试名称	APK 文件上传测试用例
测试功能	用户上传 APK 文件
测试步骤	1. 点击选取文件，选择一个 APK 文件，点击上传文件 2. 点击选取文件，选择一个非 APK 类型的文件，点击上传文件 3. 点击选取文件，选择一个 APK 类型的文件，再次点击选取文件，选择一个 APK 类型的文件 4. 点击选取文件，选择一个已经上传过的 APK 文件，点击上传文件
预期结果	1. 界面出现文件上传进度条，进度条 100% 后弹窗显示文件上传成功 2. 弹窗显示文件类型必须为 APK 类型 3. 弹窗显示只能选取一个文件上传 4. 弹窗显示 APK 文件已存在，无需上传
实际结果	与预期相符合

否在任务列表中正确显示。

表 5.18展示了查看任务列表的测试用例，主要关注系统能否正确展示地任务信息，并正确地完成任务的查看以及删除操作以及搜索功能是否有效。

表 5.19展示了用户查看任务详情的测试过程与预期结果，主要关注系统能否正确展示地任务详细情况，文本合规性检测任务是否能够下载并查看任务检测报告。

5.5 非功能性测试

本节使用测试工具对系统各功能的响应时间进行测试，包含用户登录，上传隐私政策文件，上传 APK 文件，新建隐私文本合规性检测任务，新建权限合

表 5.17: 用户创建任务测试用例

测试 ID	UC5
测试名称	用户创建任务测试用例
测试功能	用户能够完成检测任务的创建
测试步骤	1. 不填写任务名称，点击创建任务 2. 填写任务名称，不填写 APP 名称，点击创建任务 3. 填写任务名称和 APP 名称，不选择任务类型，点击创建任务 4. 填写任务名称和 APP 名称，选择任务类型，不上传相应的文件，点击创建任务 5. 填写任务名称和 APP 名称，选择任务类型，上传相应的文件，点击创建任务
预期结果	1. 任务创建失败，弹窗显示请输入任务名称 2. 任务创建失败，弹窗显示请输入 APP 名称 3. 任务创建失败，弹窗显示请选择任务类型 4. 任务创建失败，弹窗显示请上传文件 5. 任务创建成功，弹窗显示任务创建成功
实际结果	与预期相符合

表 5.18: 查看任务列表测试用例

测试 ID	UC6
测试名称	查看任务列表测试用例
测试功能	用户能够查看任务列表，完成搜索任务，查看任务详情以及删除任务的功能
测试步骤	1. 点击左侧功能栏查看任务，用户可以查看自己所有创建的任务 2. 选择一项已经完成的任务，在操作栏中点击查看按钮 3. 选择一项运行中的任务，在操作栏中点击查看详情 4. 选择一项任务，点击删除按钮 5. 选择要搜索的任务类型和状态，输入任务名称，点击搜索按钮
预期结果	1. 系统进入任务列表界面，展示所有已被创建任务的基本信息； 2. 跳转进入任务详情界面； 3. 查看详情失败，弹窗显示待任务运行结束后再查看； 4. 弹窗显示任务删除成功，页面自动刷新，已删除的任务消失； 5. 页面显示所有被搜索到的任务。
实际结果	与预期相符合

规性检测检测任务，任务列表查看，任务详情查看以及检测报告下载。对以上功能各进行 100 次操作来获取每次的响应时间，并计算平均响应时间。

从表中的数据可以得出，满足系统的时间特性非功能性需求。用户创建文本合规性检测任务和权限合规性检测任务时，表中记录的是创建任务消耗的时间，此时的任务状态为运行中。考虑用户的使用体验，当任务创建完成后，任务

表 5.19: 查看任务详情测试用例

测试 ID	UC7
测试名称	查看任务详情测试用例
测试功能	用户查看不同类型任务的任务详情，并下载任务检测报告。
测试步骤	1. 选择任务类型为文本合规性检测的任务，点击详情按钮 2. 查看文本合规性检测任务的详情界面，点击检测报告并下载 3. 选择任务类型为权限合规性检测的任务，点击详情按钮
预期结果	1. 页面直接跳转至任务详情页，该页面显示了该文本合规性检测任务的基本信息 2. 下载的报告为 pdf 格式，是本次检测的详细信息； 3. 页面直接跳转至任务详情页，该页面显示了该权限合规性检测任务的基本信息；
实际结果	与预期相符合

异步执行，用户无需等待任务运行完毕。测试结果如表 5.20所示。

表 5.20: 响应时间测试

操作	平均响应时间	最大响应时间	最小响应时间	测试结果
用户登录	705ms	1108ms	513ms	通过
上传隐私政策文件	1032ms	1908ms	613ms	通过
上传 APK 文件	4012ms	17874ms	452ms	通过
新建文本合规性检测任务	834 ms	1252ms	656ms	通过
新建权限合规性检测任务	853 ms	1373ms	685ms	通过
任务列表查看	715ms	1308ms	311ms	通过
任务结果查看	778ms	1243ms	502ms	通过
检测结果报告下载	1102ms	1854ms	813ms	通过

第六章　总结与展望

6.1　总结

本文分析了当前金融 APP 面临的隐私政策审核难题，设计并实现了一个面向金融 APP 政策合规性检测系统。本文的主要工作如下：

（1）为了完成系统的检测功能，先结合法律条款和金融领域的行业规定，设计了一个金融 APP 的隐私政策检测指标，该指标一共包含了 9 个主题，每个主题下都有多个隐私条款，一共共包含了 25 个隐私条款做为检测指标。

（2）实现了文本合规性检测。文本合规性检测分为隐私条款检测和金融专项检测。隐私条款检测先通过文本处理模块，获取隐私政策文本中每个章节的标题，然后调用基于 SVM 的分类器对标题进行分类。确认好标题的主题后，再通过文本处理模块，对同一主题的内容进行汇总。每个主题会根据自身的特点，选择使用关键词匹配或基于 BERT 模型的分类器对该主题下的语句段进行分类，最后根据分类结果判断隐私政策是否符合检测指标。金融专项检测根据隐私条款检测中分类为“收集的个人信息类型”的内容，并根据金融 APP 的类型，使用对应的 BERT 分类器对个人信息内容进行一个二分类，判断收集的个人信息是不是必要的。该检测能够帮助用户快速发现隐私政策中的缺少项与违规项。

（3）实现了权限合规性检测。先建立了一个敏感权限名称表，表中有每一种敏感权限的标准描述。接着使用静态分析从 APP 中提取了实际使用的敏感权限，并结合敏感权限表获取将这些提取的敏感权限转化成标准描述。然后使用文本处理模块和标题分类器获取“收集与使用”主题下的内容，再使用面向“收集与使用”主题的 BERT 分类器，获取分类结果为“申请的设备权限”的内容。然后使用 SimCSE 模型对 APP 中使用的敏感权限和隐私政策声明的权限进行语义相似度比较，如果 APP 使用的敏感权限都能在隐私政策中被找到，则证明该 APP 的权限描述合规。该检测能够帮助用户快速发现隐私政策中未声明却在 APP 中使用的敏感权限。

（4）为了让用户能够简单快捷地使用完成检测任务，本文开发了一个的 Web 应用，用户可以通过 Web 应用对系统进行操作。Web 应用后端采用了 Django 框架，设计成 B/S 架构。文本合规性检测和权限合规性检测两项功能都集成到了系统中，用户可以上传隐私政策文本和 APK 文件到系统中，然后创建检测任务并通过任务列表查看任务的完成状态，并对已完成状态的任务查看检测结果并

获取检测报告。

6.2 展望

本文实现了面向金融 APP 的隐私政策合规性检测系统，实现了文本合规性检测和权限合规性检测两项功能，但还有很多可以改进的地方，未来计划就以下几个方面进行改进：

（1）整理一份质量良好且规模更大的数据集，由于由于人力问题，目前的数据集规模较小，未来可以考虑使用众包的方式对更大规模的隐私政策文本集进行标注，这样的数据集能够更好的训练模型。

（2）由于检测的是纯文本，网页内容一些标注和加粗的文字无法提取，相关法规中规定了需要对敏感个人信息进行标注，纯文本的情况无法做此方面的检查，未来计划结合网页中的标签信息，挖掘更多的文字表达特征。

（3）隐私政策中的一些情况需要在运行 APP 的情况才能进行检查，本系统只通过静态分析进行了隐私权限一致性的检测操作，未来准备结合动态分析对隐私政策弹窗提示方面的合规性进行检测。

（4）目前模型的运行速度和训练速度较慢，而隐私政策的时效性较短，会涉及到比较频繁的模型训练，未来考虑使用一些轻量级的模型，加快模型的训练和运行速度。

参考文献

[1] 谈咏梅, 钱小平. 我国网站隐私保护政策完善之建议 [J]. 现代情报, 2006, 26(1) : 215 – 217.

[2] 孙佑海. 网络安全法: 保障网络安全的根本举措——学习贯彻《中华人民共和国网络安全法》[J]. 中国信息安全, 2016(12) : 30 – 33.

[3] 敖颜思文, 王旭. 金融消费者个人信息立法保护研究 [J]. 现代商业, 2019(27) : 120 – 121.

[4] 陈彦达, 王玉凤, 张强. 我国金融科技监管挑战及应对 [J]. 金融理论与实践, 2020(1) : 49 – 56.

[5] 卢智睿. 从 100 款银行 APP 测评结果看金融消费者信息的保护和利用——写在《个人信息保护法 (草案)》征求意见之际 [J]. 征信, 2021 : 38 – 46.

[6] AMMAR W, WILSON S, SADEH N, et al. Automatic categorization of privacy policies: A pilot study[J], 2012.

[7] LIN J, AMINI S, HONG J I, et al. Expectation and purpose: understanding users' mental models of mobile app privacy through crowdsourcing[C] // Proceedings of the 2012 ACM conference on ubiquitous computing. 2012 : 501 – 510.

[8] LIU F, RAMANATH R, SADEH N, et al. A step towards usable privacy policy: Automatic alignment of privacy statements[C] // Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. 2014 : 884 – 894.

[9] RAMANATH R, SCHAUB F, WILSON S, et al. Identifying Relevant Text Fragments to Help Crowdsource Privacy Policy Annotations[C] // Proceedings of the AAAI Conference on Human Computation and Crowdsourcing : Vol 2. 2014 : 54 – 55.

[10] BHATIA J, BREAUX T D, REIDENBERG J R, et al. A Theory of Vagueness and Privacy Risk Perception[C] // Requirements Engineering Conference. 2016 : 26 – 35.

[11] OLTRAMARI A, PIRAVIPERUMAL D, SCHAUB F, et al. PrivOnto: a Semantic Framework for the Analysis of Privacy Policies[J]. Semantic Web, 2017, 9(6) : 1 – 19.

[12] HARKOUS H, FAWAZ K, LEBRET R, et al. Polisis: Automated analysis and presentation of privacy policies using deep learning[C] // 27th USENIX Security Symposium (USENIX Security 18). 2018 : 531 – 548.

[13] CONTISSA G, DOCTER K, LAGIOIA F, et al. Claudette Meets GDPR: Automating the Evaluation of Privacy Policies Using Artificial Intelligence[J]. Social Science Electronic Publishing, 2018.

[14] LIU S, ZHAO B, GUO R, et al. Have You been Properly Notified? Automatic Compliance Analysis of Privacy Policy Text with GDPR Article 13[C] // Proceedings of the Web Conference 2021. 2021 : 2154 – 2164.

[15] PANDITA R, XIAO X, YANG W, et al. WHYPER: Towards Automating Risk Assessment of Mobile Applications[J], 2013 : 527 – 542.

[16] SLAVIN R, WANG X, HOSSEINI M B, et al. Toward a framework for detecting privacy policy violations in android application code.[C] // IEEE. 2016 : 25 – 36.

[17] YU L, LUO X, QIAN C, et al. Enhancing the Description-to-Behavior Fidelity in Android Apps with Privacy Policy[J]. IEEE Transactions on Software Engineering, 2017, 44(9) : 834 – 854.

[18] 赵京胜, 宋梦雪, 高祥. 自然语言处理发展及应用综述 [J]. 信息技术与信息化, 2019(7) : 143–146.

[19] ZHANG M, DENG Z, CHE W, et al. Combining Statistical Model and Dictionary for Domain Adaption of Chinese Word Segmentation[J]. Journal of Chinese Information Processing, 2012, 26(2) : 8 – 12.

[20] SALTON G, BUCKLEY C. Term-weighting approaches in automatic text retrieval[J]. Information processing & management, 1988, 24(5) : 513 – 523.

[21] DEVLIN J, CHANG M-W, LEE K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.

[22] VASWANI A, SHAZEER N, PARMAR N, et al. Attention Is All You Need[J]. Advances in neural information processing systems, 2017, 30.

[23] SUTSKEVER I, VINYALS O, LE Q V. Sequence to sequence learning with neural networks[C] // Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 2. 2014 : 3104 – 3112.

[24] BAHDANAU D, CHO K, BENGIO Y. Neural machine translation by jointly learning to align and translate[J]. arXiv preprint arXiv:1409.0473, 2014.

[25] GAO T, YAO X, CHEN D. SimCSE: Simple Contrastive Learning of Sentence Embeddings[C] // Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. 2021 : 6894 – 6910.

[26] GAO J, HE D, TAN X, et al. Representation degeneration problem in training natural language generation models[J]. arXiv preprint arXiv:1907.12009, 2019.

[27] LI B, ZHOU H, HE J, et al. On the Sentence Embeddings from Pre-trained Language Models[C] // Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2020 : 9119 – 9130.

[28] SU J, CAO J, LIU W, et al. Whitening sentence representations for better semantics and faster retrieval[J]. arXiv preprint arXiv:2103.15316, 2021.

[29] WICHMANN B A, CANNING A, CLUTTERBUCK D, et al. Industrial perspective on static analysis[J]. Software Engineering Journal, 1995, 10(2) : 69 – 75.

[30] WILSON S, SCHAUB F, DARA A A, et al. The creation and analysis of a website privacy policy corpus[C] // Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2016 : 1330 – 1340.

[31] ZIMMECK S, STORY P, SMULLEN D, et al. MAPS: Scaling Privacy Compliance Analysis to a Million Apps[J]. Proceedings on Privacy Enhancing Technologies, 2019, 2019(3) : 66 – 86.

[32] SRINATH M, WILSON S, GILES C L. Privacy at scale: Introducing the privaseer corpus of web privacy policies[J]. arXiv preprint arXiv:2004.11131, 2020.

[33] CUI Y, CHE W, LIU T, et al. Pre-training with whole word masking for chinese bert[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2021, 29 : 3504 – 3514.

[34] SUN C, QIU X, XU Y, et al. How to fine-tune bert for text classification?[C] // China national conference on Chinese computational linguistics. 2019 : 194 – 206.

[35] JOULIN A, GRAVE E, BOJANOWSKI P, et al. Bag of tricks for efficient text classification[J]. arXiv preprint arXiv:1607.01759, 2016.

[36] KIM Y. Convolutional Neural Networks for Sentence Classification[J]. Eprint Arxiv, 2014.

[37] LIU P, QIU X, HUANG X. Recurrent neural network for text classification with multi-task learning[C] // Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence. 2016 : 2873 – 2879.

[38] LIPTON Z C, BERKOWITZ J, ELKAN C. A critical review of recurrent neural networks for sequence learning[J]. arXiv preprint arXiv:1506.00019, 2015.

[39] HOCHREITER S, SCHMIDHUBER J. Long Short-Term Memory[J]. Neural Computation, 1997, 9(8) : 1735 – 1780.

[40] CHO K, MERRIENBOER B V, GULCEHRE C, et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation[C] // Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2014 : 1724 – 1734.

简历与科研成果

基本情况 吴青衡，男，汉族，1997 年 4 月出生，重庆市忠县人。

教育背景

2020.09 ~ 2022.06 南京大学软件学院 硕士

2016.09 ~ 2020.06 重庆大学计算机学院 本科

致　　谢

在本篇论文完成之际，我想向硕士就读生涯中所有帮助和关心过我的老师、同学、朋友、亲人致以衷心的感谢。

首先，我要感谢我的导师陈振宇老师，陈老师是一名治学严谨、勤奋乐观的学者，在我的整个研究生生涯中给予了我很大帮助，教会了我如何成为一名优秀的研究生。在我毕业论文的书写过程中，陈老师一直严格监督我的写作进度，当我论文内容出现偏差时，陈老师会及时提出了修改意见，让我能够高效准确地完成论文写作。

其次，我要感谢所有帮助过我的学长、同学和朋友。感谢实验室的赵源学长，从论文最初的选题、写作过程中的各种修改再到论文最终的完成，都离不开赵源学长的帮助。感谢实验室里面的各位同学，研究生期间大家互帮互助，共同度过了愉快的两年。感谢身边的朋友王擎宇，王舒遥等人，无论是课程学习、实习期间还是最后的毕业设计，大家都经常交流讨论，让我受益颇丰。感谢我的本科同学蒋豪、彭浩等人，他们在我遇到挫折时经常鼓励开导我。

最后，我要感谢我的父母，父母一直是我的坚强后盾和精神支柱，使我在安心学习的同时能有一片平静祥和的精神世界。感谢母校南京大学对我学术能力以及精神品格上的培养，让我在今后人生中，能够时刻牢记“诚朴雄伟, 励学敦行”的校训，走好人生的每一步。