数据科学大作业

——基于自然语言处理的隐私信息扫描研究

殷天逸、沈霁昀、董志昂

NJU

2022 年 1 月 22 日

OUTLINE

简介

背景

近年来，与数据滥用、隐私泄露及个人信息安全的相关问题在大数据时代背景下被不断激化。2018 年，“Facebook 剑桥分析数据丑闻”的爆发彻底引发了人们对于个人隐私数据安全性的关注与思考。自此以后，个人隐私数据泄露与滥用事件层出不穷。粗略的统计下，仅 2020 年一年，大型的隐私泄露信息事件就发生了 62 起。

简介

背景

在 2016 年 4 月 27 日，欧盟出版了《通用数据保护条例》以在法律法规层面保护个人隐私信息，该条例于 2018 年 5 月 25 日正式生效。2019 年 10 月，与之类似的《加州消费者隐私法》也在美国加州经过修正正式生效。中国在法律层面对于个人隐私信息的保护相对起步较晚。2021 年 11 月 1 日，正式施行的个人信息保护法为大众撑起了一把个人隐私信息法律层面的“保护伞”。

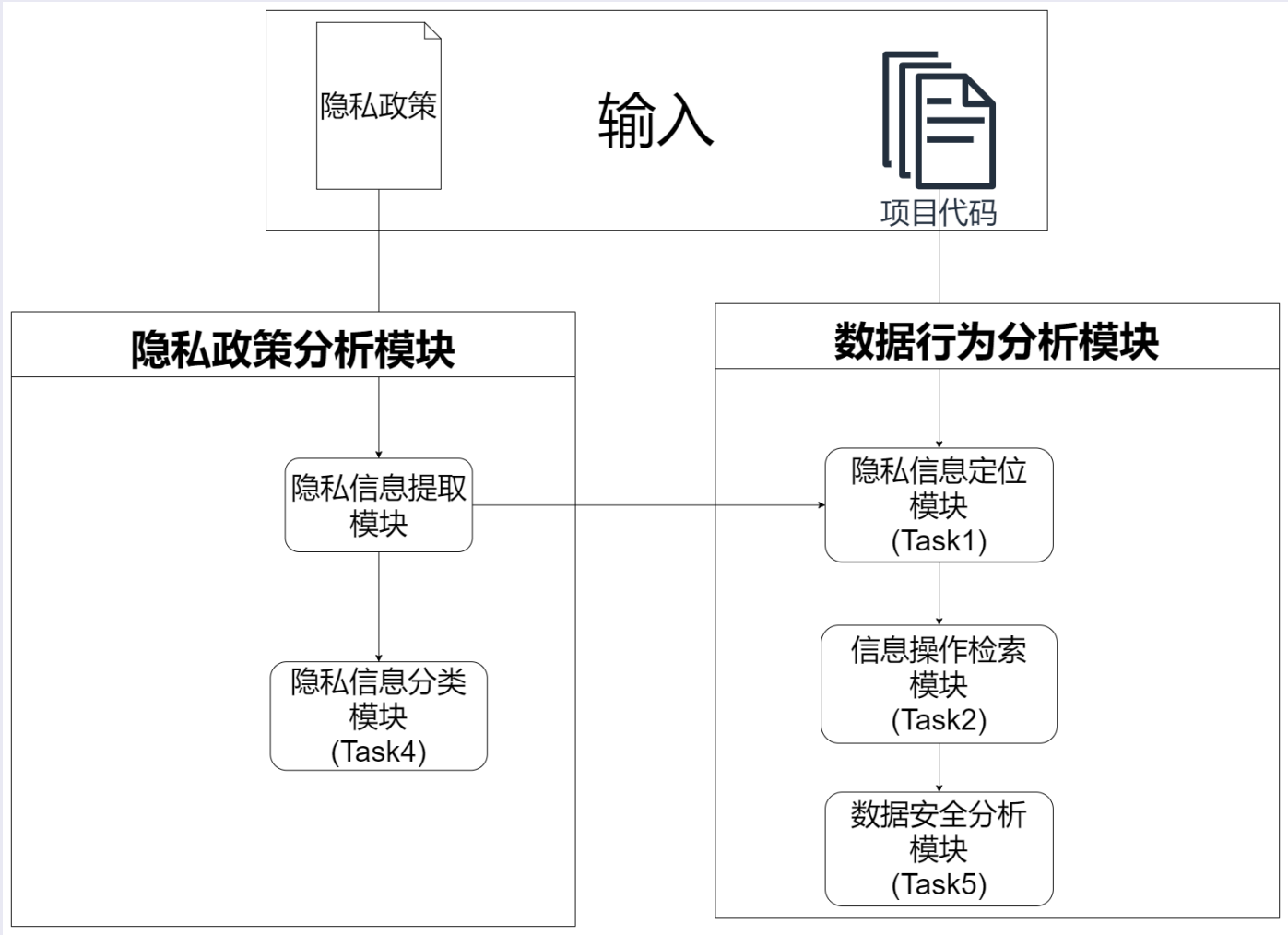
简介

背景

企业资源计划 (Enterprise Resource Planning 下称 ERP)，是指组织用于管理日常业务活动的一套软件，这些活动包括会计、采购、项目管理、风险管理和合规性、供应链运营等等。完整的 ERP 套件还包括企业绩效管理软件，用于帮助企业围绕财务业绩进行规划、预算、预测和报告。简而言之，ERP 是一套现代企业用于整合人员、流程和技术的系统，通过网格化、规格化的统筹安排来实现对一个公司所有员工的管理。在涉及对员工的管理方面，员工的个人隐私信息在某些场合便不可避免地需要被收集并利用。对于 ERP 项目的用户企业来说，有时也需要提供相关的隐私数据来方便 ERP 项目的管理与运行。在这个过程中，个人隐私信息的安全问题就值得我们去进一步关注。

技术框架

总路线图



TASK 1

找到代码中的个人信息

TASK 1

隐私信息定位

注意到在程序之中，个人信息往往是储存在变量之中，大部分情况下，变量名是有意义的。在此基础之上，我们希望能够扫描整个项目，找出源码中可能用来存储信息的变量。

TASK 1

隐私信息定位

考虑到一般来说变量的命名都较为规范，我们希望能够构建一个**隐私信息文本库**，对于前文找出的所有变量，我们只需要将其与文本库中内容进行比对，即可判断其是否为存有隐私信息的变量。

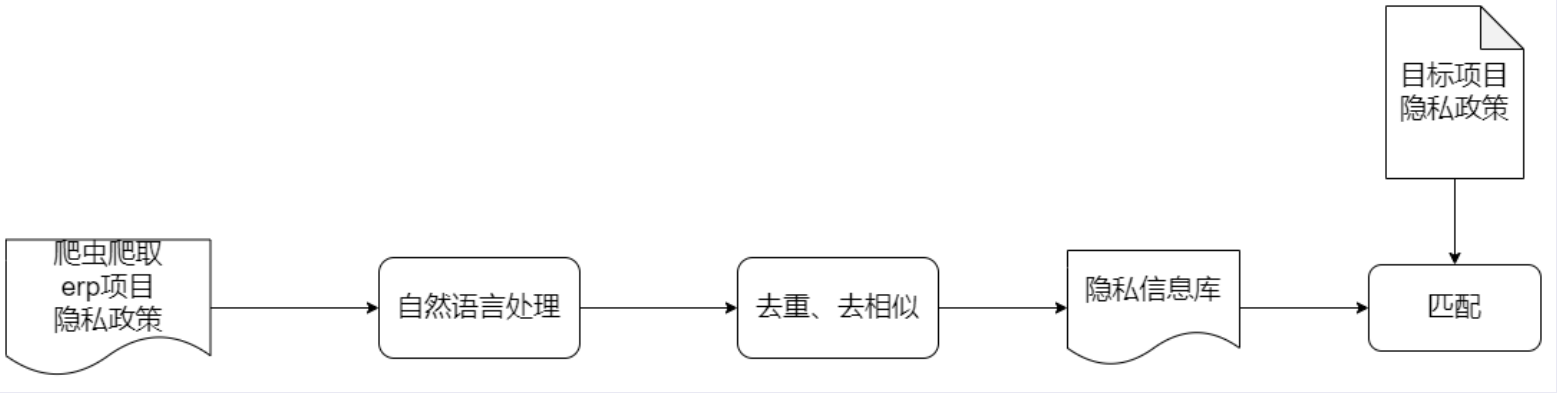
TASK 1

隐私信息定位

有关文本库的构建，我们利用爬虫，爬取了一些隐私政策。人工将其中涉及到的隐私信息提取出来，经过处理后构建了**隐私信息文本库**。

TASK 1

提取模块



TASK 1

隐私信息文本库部分可视化

TASK 1

隐私信息定位

我们利用了抽象语法树 (Abstract Syntax Tree，AST) 来分析源码。我们借助了 python 的 AST 包，支持我们找出代码中所有为变量类型的字符串。

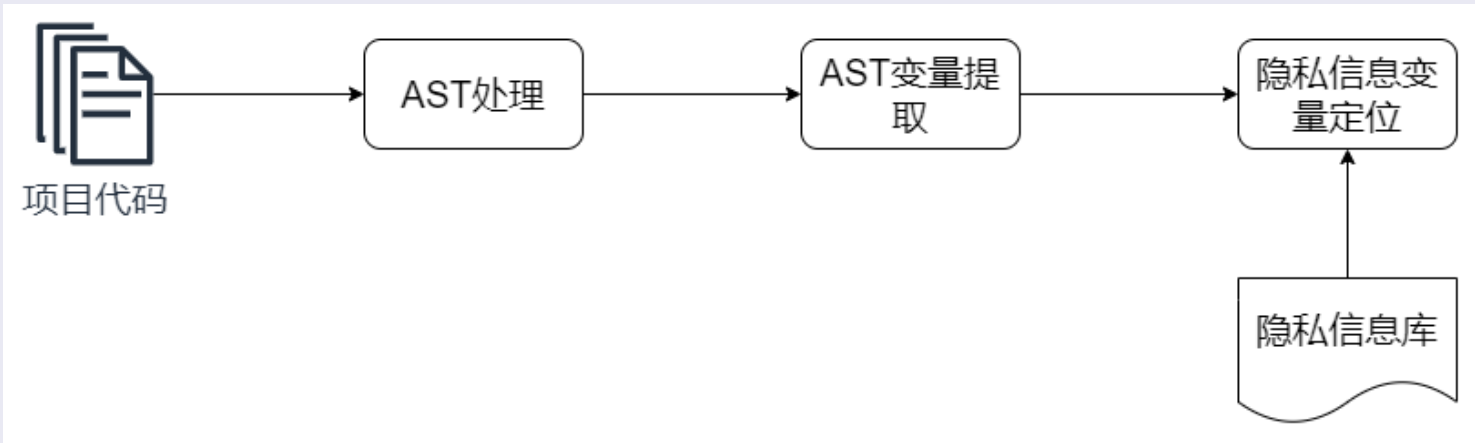
TASK 1

抽象语法树

```
            type_comment=None,
        ),
        Assign(
            targets=[Name(id='country_id', ctx=Store())],
            value=Call(
                func=Attribute(
                    value=Name(id='fields', ctx=Load()),
                    attr='Many2one',
                    ctx=Load(),
                ),
                args=[],
                keywords=[
                    keyword(
                        arg='string',
                        value=Constant(value='Country', kind=None),
                    ),
                    keyword(
                        arg='comodel_name',
                        value=Constant(value='res.country', kind=None),
                    ),
                    keyword(
                        arg='help',
                        value=Constant(value='Country for which this tag is available, when applied on taxes.', kind=None),
                    ),
                ],
            ),
            type_comment=None,
        ),
        FunctionDef(
            name='_get_tax_tags',
```

TASK 1

信息定位模块



TASK 2

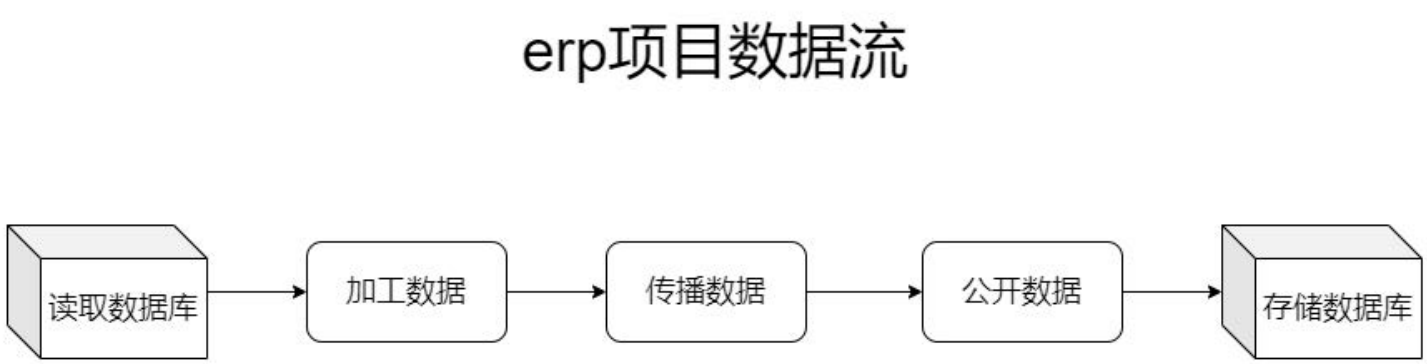
TASK 2

找到代码中对个人信息的处理操作。

TASK 2

找到代码中对个人信息的处理操作

我们考虑首先从程序与数据库的交互入手，检索对数据库操作的语句，判断程序对数据库的增删查改操作，便可识别定位程序对信息的操作。而在读取数据库之后或是存进数据库之前的操作，则是对数据的使用、加工或公开等，是下一步研究的工作。



TASK 2

找到代码中对个人信息的处理操作

我们发现一个大的项目，往往会有一些基础设施的建设，将 SQL 操作进行封装，就是其中很常见的一种，以我们研究的 Odoo 项目为例，将数据库操作封装为了 Environment 类，含有 create, delete, execute, unlink 等一系列封装了 SQL 语句的与数据库交互的函数。因此，经过一系列研究，我们认为对于数据库操作的检索，应该既要直接寻找 SQL 语句，也要先观察其是否有相关基础设施函数，是需要视具体情况而定的。

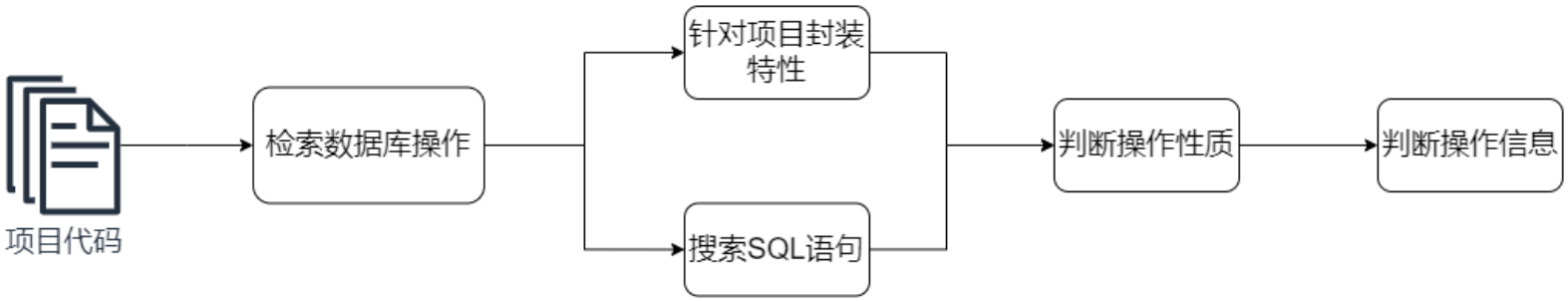
TASK 2

找到代码中对个人信息的处理操作

然而对于数据的处理不仅仅只包含与数据库进行交互这一项处理，还包含对信息的收集、存储、使用、加工、传输、提供、公开、删除等基本操作。由于函数名通常都有其实际含义，我们仍然尝试从函数名和注释入手，判断它的处理类型。

TASK 2

操作定位模块



TASK 4

TASK 4

对获取隐私政策中的个人信息进行自动分类。

TASK 4

隐私信息自动分类

在 TASK 1 中我们提到了**隐私信息文本库**，在 TASK 4 中，我们希望能够借助机器学习实现对隐私政策中涉及到的隐私信息进行分类。

TASK 4

隐私信息自动分类

文本分类的核心都是如何从文本中抽取出能够体现文本特点的关键特征，抓取特征到类别之间的映射。
传统的词袋模型例如 TF-IDF 等，虽然比较简单直观，但是仅将词语符号化，没有考虑词之间的语义联系。因此我们考虑采用词嵌入模型。最著名的词嵌入模型是 Google 的 Word2Vec(2013)，其他的还有斯坦福大学的 GloVe(2014) 和 Facebook 的 FastText(2016)。

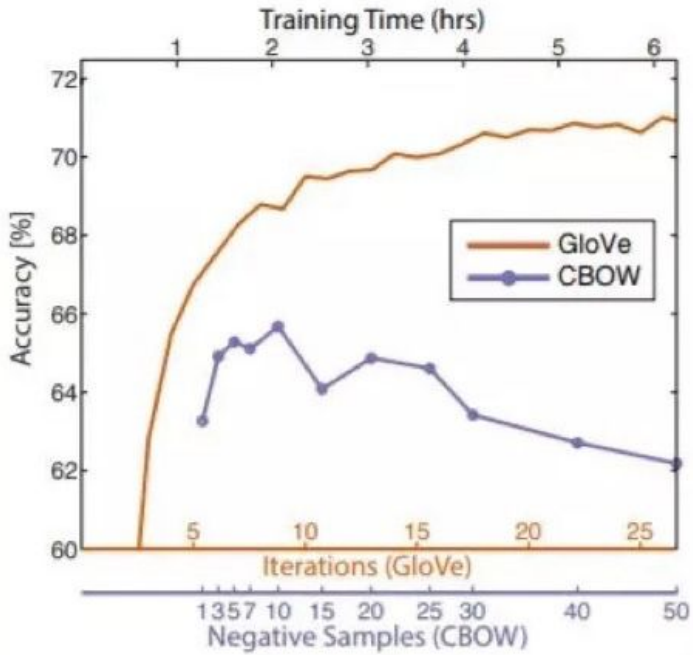
TASK 4

隐私信息自动分类

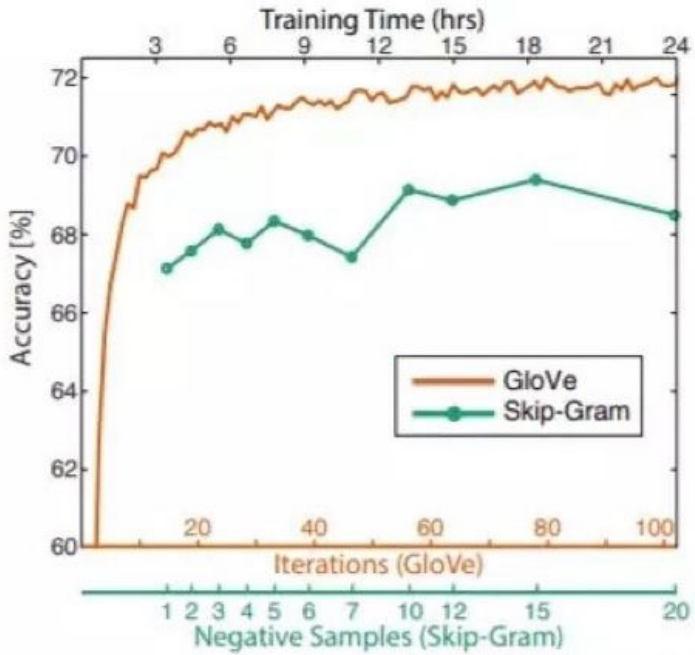
我们使用了 spacy 预训练的 GloVe 模型，将对于一个文本，求与每一类词的相似度。Word2Vec 只考虑到了词的局部信息，没有考虑到词与局部窗口外词的联系，GloVe 利用共现矩阵，同时考虑了局部信息和整体的信息，在语义相似度上表现更加优秀。

TASK 4

隐私信息自动分类



GloVe vs CBOW



GloVe vs Skip-Gram

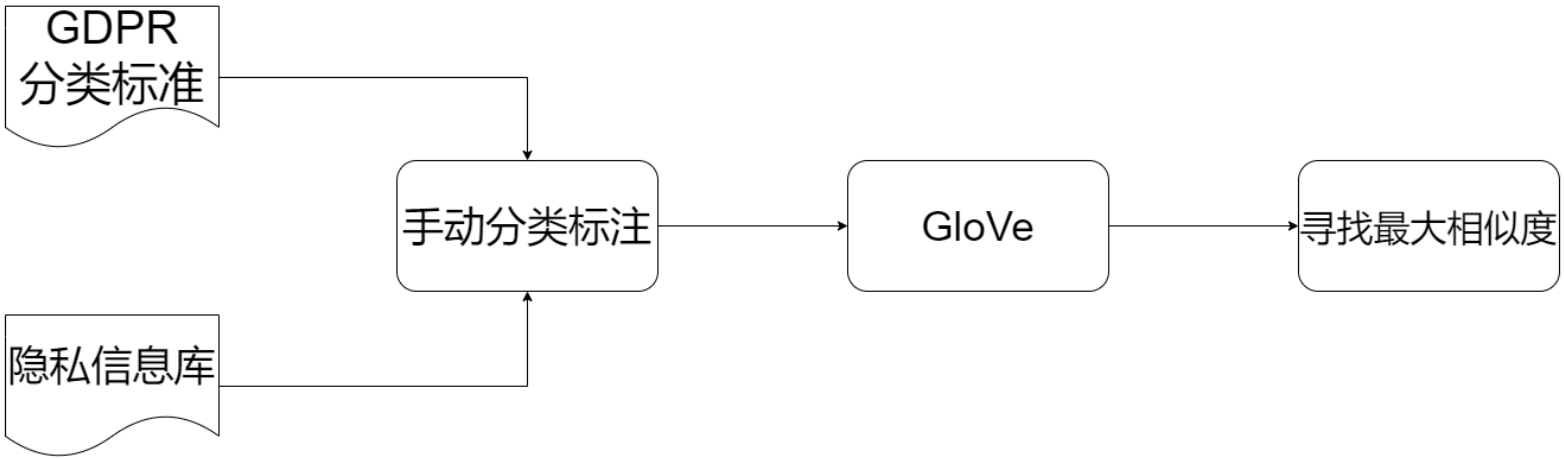
TASK 4

隐私信息自动分类

我们参考了一篇文献 (https://arxiv.org/abs/2101.11574)，文中对于分类使用的是求与每组相似度的算术平均值，但在实践后，我们发现效果并不是很好，于是我们调整了策略，采用求相似度最大值的方法进行分类，结果证明分类效果较为显著。

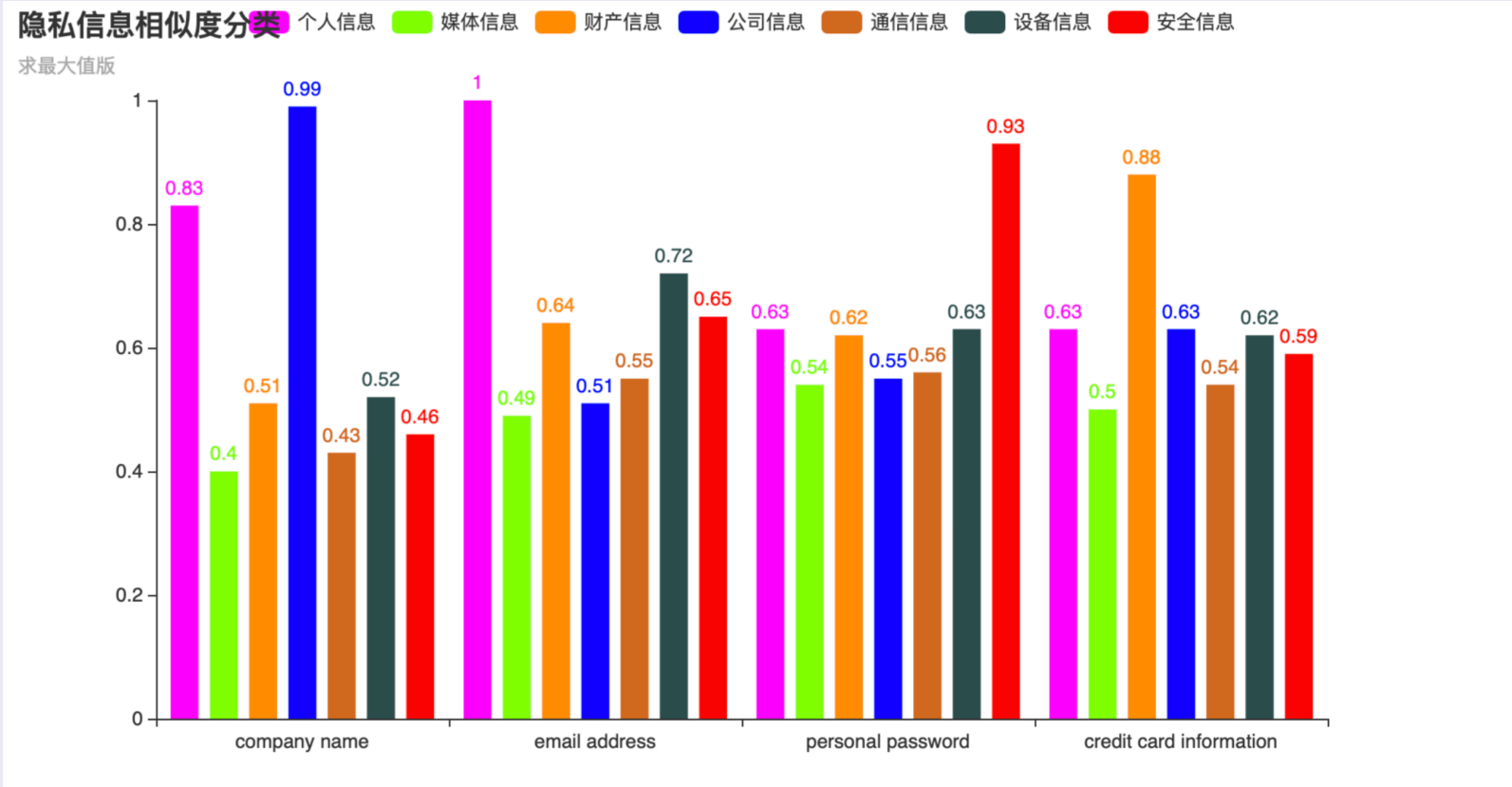
TASK 4

分类模块



TASK 4

分类效果


隐私信息相似度分类
个人信息
媒体信息
财产信息
公司信息
通信信息
设备信息
安全信息
求最大值版
1
0.8
0.6
0.4
0.2
0
0.83
0.4
0.51
0.99
0.43
0.52
0.46
1
0.49
0.64
0.51
0.55
0.72
0.65
0.63
0.54
0.62
0.55
0.56
0.63
0.93
0.63
0.5
0.88
0.63
0.54
0.62
0.59
company name
email address
personal password
credit card information

TASK 5

判断用户数据在处理过程中是否进行过安全处理。

TASK 5

判断用户数据在处理过程中是否进行过安全处理

对于加密的检测，因为常用的加密方式并不多 (hashlib,hmac 等等)，如果用 python 自带的加密方法需要引入相应的包，我们希望能够直接通过检查所用到的包以及所用到的加密函数来找到加密操作。

实验对象

实验对象

我们以开源的 ERP 类型 WEB 项目 Odoo(https://github.com/odoo/odoo) 为研究对象，分析了其源码中与隐私信息有关的内容。Odoo 主要是基于 python 实现的。

实验对象

实验对象

Odoo 不仅仅是一个简单的开源 ERP，Odoo 更像一个框架 + 市场的平台，不但包含了 ERP、CRM、PLM、HR 等核心企业应用，还有电子商城、智能建站、社区、POS、门店管理、物流等行业应用。目前，各种应用的数量已经达到惊人的 15850 个，它的流动性和完全整合可满足甚至是最复杂公司的需求。Odoo 的灵活性在于这些应用可按照公司的发展进行添加，随着需求的变化和客户群的发展逐一添加应用。

但也正因如此，Odoo 的用户隐私信息安全就显得愈发重要，关系到诸多公司及其用户，为了研究这一问题，我们希望通过数据科学分析等方法的帮助，实现对系统隐私信息的检测。

总结

优点

任务 1 和任务 4 具有较高的扩展性。
任务 1 采用了基于 AST 的隐私信息定位方法，有较好的可移植性。
任务 4 的基于语义相似度的分类方法效果较为显著。

总结

不足及可改进之处

任务 2 和任务 5 的扩展性较差。因为 odoo 项目较大，内部架构较为复杂，很多地方难以溯源。这导致了任务 2 采用的方法可移植性较差，而任务 5 不能完全找到对所有隐私信息的加密操作，可能会有遗漏。

总结

不足及可改进之处

任务 1 和任务 4 较为依赖隐私信息文本库，文本库的完整度一定程度上影响了任务 1 和任务 4 的完成效果。任务 4 中采取了单一的词嵌入方法，缺乏实验去比较不同方法 (Word2Vec 和 GloVe) 的效果。

任务分配

- 殷天逸: 任务 1 和任务 5、提出隐私信息分类算法
- 沈霁昀: nlp 部分、隐私信息分类的代码实现、提出任务 2 的算法
- 董志昂: 爬取隐私政策、构建隐私信息文本库、任务 2 代码实现

成果展示