基于秘密共享的安全命名实体识别推理方法

佟 岩[1], 花忠云[1], 廖 清[1], 张玉书[2]

[1]哈尔滨工业大学(深圳) 计算机科学与技术学院, 深圳 中国 518055
[2]南京航空航天大学 计算机科学与技术学院, 南京 中国 210016

摘要 命名实体识别旨在识别一段文本中特定的实体, 这些识别出来的实体可以用来提升下游任务的性能, 因此命名实体识别已成为自然语言处理领域的基础任务之一。BiLSTM-CRF 模型以其结合了深度学习与统计机器学习的优点, 已成为处理该任务的基线方法, 然而现有的 BiLSTM-CRF 模型仅支持明文域下的推理, 从而在因资源不足而将计算外包的场景下会出现一定的隐私泄露问题。为了解决 BiLSTM-CRF 命名实体识别外包推理过程中的隐私泄露问题, 本文提出了一种基于秘密共享的安全命名实体识别推理方法 SecNER。SecNER 基于秘密共享技术对 BiLSTM-CRF 命名实体识别模型中涉及的非线性激活函数、最大值信息获取、数组访问等算子进行安全化设计, 并根据这些安全化算子构建了整个推理系统。SecNER 能够保证在 BiLSTM-CRF 命名实体识别外包推理过程中, 用户上传的待预测数据的安全性与被托管的模型参数的安全性。为了进一步优化安全词向量提取操作的性能, 本文对词向量嵌入层结构进行了调整, 采用了分桶的思想, 从而减少了桶中单词的个数, 进而减少了安全词向量提取过程中的通信开销。本文利用基于模拟的安全性分析方法对 SecNER 进行了安全性证明, 并设计实验证明了方案可行性。实验结果表明, 在三个数据集上与明文推理方法相比, 安全的命名实体识别推理方法的 F1 值最多下降 0.001, 且推理的时间开销在可接受范围内。

关键词 命名实体识别; 信息安全; 秘密共享; 隐私计算
中图法分类号 TP309.2 **DOI 号** 10.19363/J.cnki.cn10-1380/tn.2025.11.04

Secure Named Entity Recognition Inference Method Based on Secret Sharing

TONG Yan[1], HUA Zhongyun[1], LIAO Qing[1], ZHANG Yushu[2]

[1] School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen 518055, China
[2] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

Abstract Named entity recognition is designed to identify distinct entities within a given textual context. The entities identified through this process serve as valuable components for improving the efficacy of subsequent natural language processing tasks. Consequently, named entity recognition has evolved into a foundational task in the realm of natural language processing. The BiLSTM-CRF model, with its integration of the advantages of deep learning and statistical machine learning, has emerged as a baseline method for addressing this task. However, existing BiLSTM-CRF models only support inference in plaintext domains, leading to potential privacy leakage issues in scenarios where computations are outsourced due to insufficient resources. To address the privacy issues arising during the outsourcing inference process of BiLSTM-CRF named entity recognition, SecNER, a secure named entity recognition inference method based on secret sharing is proposed in this paper. SecNER employs secret sharing techniques to secure the operations involved in the BiLSTM-CRF named entity recognition model, such as nonlinear activation function, retrieving maximum value information from an array and array accessing. The entire inference system is constructed based on these secure operations. SecNER ensures the security of both the user's uploaded data for prediction and the hosted model parameters during the outsourcing inference. To further optimize the performance of secure word vector extraction operations, adjustments are made to the structure of the word vector embedding layer in this paper. The concept of bucketing is employed to reduce the number of words in each bucket, thereby minimizing the communication overhead during secure word vector extraction. A simulation-based security analysis is provided and extensive experiments have been designed to demonstrate the feasibility of SecNER. Results indicate that, compared to the plaintext model on three datasets, the F1 score of the proposed secure named entity recognition inference method decreases by at most 0.001. Moreover, the time overhead of the inference process remains within an acceptable range.

Key words named entity recognition; information security; secret sharing; privacy-preserving computation

通信作者: 花忠云, 博士, 副教授, Email: huazhongyun@hit.edu.cn。
本课题得到国家自然科学基金项目(No. 62071142)资助。
收稿日期: 2024-04-06; 修改日期: 2024-06-04; 定稿日期: 2025-10-17

1 引言

命名实体识别(named entity recognization, NER)旨在识别一段文本中的特定的实体如人名、地名、机构名等。这些识别出来的实体可以用来提升像机器翻译等下游任务的性能, 从而 NER 已成为自然语言处理(Natural Language Processing, NLP)领域的基础任务之一。BiLSTM-CRF 模型[1]以其结合了双向长短时记忆网络(Bidirectional Long Short-Term Memory, BiLSTM)和条件随机场(Conditional Random Field, CRF)的优点, 已经成为处理该任务的基线方法。然而现有的 BiLSTM-CRF 模型仅支持明文域下的推理, 从而在外包推理的场景下存在严重的隐私泄露问题。

考虑如下的场景, BiLSTM-CRF 模型的持有者想要向外提供命名实体识别推理服务。由于该模型可能是持有者通过安全外包训练获得的, 持有者本身并没有足够的算力。此外, 即使模型持有者拥有一定的算力, 考虑到推理任务需要时刻满足用户的计算需求, 即使在某些时刻没有请求量, 这些用于推理的算力也必须处于闲置状态, 以备不时之需, 这会导致计算资源的浪费。因此, 模型持有者倾向于将模型推理任务的计算外包给云服务商[2-3]。在传统模式下, 整个过程模型以明文的形式暴露给云服务商, 从而造成了模型隐私泄露的问题。除此之外, 使用该服务的用户, 需要将自己待预测的文本数据上传给云服务商以获取推理结果, 这些数据可能包含了重要的隐私信息, 将其以明文形式上传无疑损害了用户隐私。因此如何保护该场景下的用户数据安全和模型参数安全具有重要的研究意义。

隐私保护机器学习是解决机器学习中隐私泄露问题的有效手段, 近年来已引起了研究学者的广泛关注。按照保护的阶段可将其分为两类[4], 其中第一类关注于对机器学习训练过程中的隐私信息加以保护[5-8], 第二类则关注于对推理过程中的隐私信息加以保护[2,9-12], 这两种类型均能保证模型参数安全和用户数据安全。

随着隐私保护机器学习的兴起, 近期一些工作将关注点投放在了自然语言处理领域[13-17]。文献[13]利用安全多方计算技术构建了一个端到端的推理系统, 可以用于机器翻译任务之中。文献[15]和文献[16]针对文本恶意检测场景, 提出了隐私保护文本分类系统。以上工作均属于在机器学习推理阶段对隐私加以保护。文献[14]利用同态加密技术[18]对 Word2Vec 模型训练过程加以保护, 能够得到与明文下训练近乎相同质量的词向量。但是由于同态加密操作计算开销较大, 其训练时间较长, 从而无法支持大规模的数据集。文献[17]利用轻量级的秘密共享技术对 GloVe 模型训练过程加以保护, 由于秘密共享能高效进行密文计算的特点, 其可以支持更大规模的训练语料。

尽管近年来针对隐私保护机器学习的研究已经取得了显著的进展, 现有的工作尚未有针对 BiLSTM-CRF 命名实体识别外包推理场景下的隐私泄露问题加以解决的。已有的大部分研究[2, 19-21]更多关注于针对图像的卷积神经网络模型的隐私保护, 而 BiLSTM-CRF 命名实体识别推理属于自然语言处理任务, 且采用的是循环神经网络, 该网络的计算误差在迭代计算过程中会不断扩大, 从而针对该场景下的隐私保护方案要求有更高的精度。现有的针对自然语言处理领域的隐私保护工作[13-17], 要么仅关注于深度学习模型, 要么仅关注于传统统计机器学习模型, 而 BiLSTM-CRF 结合了二者的特点, 且其中的传统机器学习模型为条件随机场模型, 是这些工作所未涉及的, 从而这些工作无法直接用来解决命名实体识别外包推理场景下的隐私泄露问题。除此之外, 与图片可以直接转化为数字表示不同, 文本单词需要通过词向量嵌入层将单词映射为向量表示, 如何安全且高效地实现这种映射, 也是一个急需解决的难题。由此可见, 针对 BiLSTM-CRF 命名实体识别外包推理场景下的隐私保护, 是一项困难且有意义的工作。

为了解决上述隐私泄露问题, 本文基于秘密共享技术提出了一个安全的命名实体识别推理系统 SecNER。SecNER 可以实现 BiLSTM-CRF 命名实体识别外包推理场景下的模型参数安全与用户数据安全。为了计算的高效性, 本文采用了轻量级的秘密共享技术提供密码学上的安全性保障。SecNER 可分为三个大的子模块, 分别为安全词嵌入层、安全 BiLSTM 层和安全 CRF 层。在安全词嵌入层, 为了优化安全词向量获取操作, 本文对词嵌入层的结构进行了调整, 将单词分到了多个桶中, 从而降低了每个桶中单词的数量, 进而大大减少了计算时的通信开销。除此之外, 本文还利用相关 Beaver 三元组[22]对该过程中的乘法计算进一步优化, 减少了其造成的通信开销。本文利用秘密共享技术对安全 BiLSTM 层与安全 CRF 层中涉及非线性算子进行安全化, 采用迭代近似拟合的方式去实现 BiLSTM 层中的非线性函数 Sigmoid 与 Tanh 在密文域下的计算, 基于安全比较算法来实现 CRF 层中数组最大值函数 max 与数组最大值索引 arg max 的安全化, 采用盲化旋转偏

移量的方式来实现安全数组访问。本文采用基于模拟的安全分析方法对 SecNER 的安全性进行了证明, 并设计实现了所提出的系统, 在三个数据集上进行了性能测试。实验结果表明, 在三个数据集上与明文模型相比, SecNER 的推理 F1 值最多下降 0.001, 且推理的时间开销处于可接受范围。本文的贡献点概括总结如下。

(1) 本文提出了一个安全的命名实体识别推理系统 SecNER, 可以保证 BiLSTM-CRF 命名实体识别外包推理过程中的模型参数安全与用户数据安全。

(2) 本文将秘密共享技术与 BiLSTM-CRF 命名实体识别推理深度结合, 构建了一系列推理所需的安全算子, 设计了高效的安全词向量获取算法。

(3) 本文形式化地证明了 SecNER 的安全性, 在三个数据集上的实验结果表明, SecNER 与明文模型在精度上拥有近乎相同的推理性能, 且推理时间开销处于可接受范围内。

2 预备知识

2.1 密码学工具

2.1.1 加性秘密共享

加性秘密共享也称算数秘密共享是一种轻量级的加密方式, 可以支持密文下的加法与乘法计算。给定两个值的密文, 加性秘密共享可以支持计算两个值加法和乘法结果的密文。本文采用的是两方加性秘密共享。对于环 $\mathbb{Z}_{2^l}$ 中 l 比特的隐私数据 x, 加性秘密共享通过将其拆分为环中的两个元素 x_1 与 x_2 来实现对其加密。x_1 与 x_2 在环中是均匀随机的, 不会泄露 x 的任何信息, 且满足 $x_1+x_2\equiv x\ (\mathrm{mod}\,2^l)$, 由此可由 x_1 与 x_2 恢复原始值 x。具体而言, 对于值 x 的加密, 可先在环中随机选择一个值作为 x_1, 计算 $x-x_1$ 作为 x_2, 最后将 x_1 与 x_2 分别发送给两个参与方即可。向量或矩阵的加密也是类似, 不过需要将随机值 x_1 替换为随机向量或矩阵。尽管加密后的 x_1 与 x_2 单独来看已不具备原始值的有效特征, 但由于 x_1 与 x_2 结合在一起包含了 x 的所有信息量, 两个拥有 x_1 和 x_2 的参与方可以通过交互执行一些操作, 安全完成密文域下的计算[20]。本文将 x 的算数秘密共享用 $\langle x\rangle$ 表示, x 的两个秘密份额用 $\langle x\rangle^1$ 和 $\langle x\rangle^2$ 表示。特别地, 比特长度为 1 的算数秘密共享被称为布尔共享, 为了便于区分, 本文用 $[\![x]\!]$ 表示 x 的布尔共享, 用 $[\![x]\!]^1$ 和 $[\![x]\!]^2$ 表示 x 的两个布尔共享份额。加性秘密共享具有线性性, 进行秘密共享的加法与秘密共享和明文的乘法时, 不需要任何交互, 参与方本地即可完成计算。在进行秘密共享间的乘法时, 需要依靠 Beaver 三元组[23]辅助实现, 两个参与方需进行一轮通信传递中间信息完成计算。对两个 l 比特的秘密共享进行乘法操作的通信开销为 $4l$ 比特。特别地, 文献[20]指出, 在进行秘密共享矩阵乘法时, 可以使用 Beaver 三元组的矩阵形式辅助实现。

2.1.2 截断方式

秘密共享所支持的操作都是在整数环中进行的, 然而在机器学习任务的计算中, 涉及的都是实数域上的计算, 为此本文采用定点数的方式来表示小数。具体来说, 可以将实数 x 通过近似缩放的方式将其映射到整数环 $\mathbb{Z}_{2^l}$ 中相对应的整数值 $\bar{x}=\lfloor x\cdot 2^q\rfloor\ (\mathrm{mod}\,2^l)$, 在这里 q 是用来控制精度的缩放因子。考虑到两个放缩后的值进行相乘运算, 其结果相当于利用放缩因子 $2q$ 进行缩放, 为此, 在每次进行放缩后元素的乘法操作后, 要将结果的缩放因子降为 q。本文采用文献[20]中本地截断的方法, 每个参与方只需将结果最低 q 位直接截断即可, 该本地截断方式被证明在环足够大的情况下, 以极高的概率最多对结果产生 1 比特的误差, 可用性极高。

2.2 命名实体识别推理方法

命名实体识别旨在识别文本中特定的实体, 是自然语言处理领域的基础任务。BiLSTM-CRF 模型以其结合了神经网络和统计机器学习两种方法的优点, 识别精度较高, 是该任务的基线方法。该模型共分为两层, 第一层为一个双向循环神经网络 BiLSTM, 用来提取文本的特征, 并将特征传入到第二层, CRF 层。CRF 层经过计算会输出每个单词所属的标签类别, 通过这些标签类别可以组建实体, 从而完成实体识别任务。BiLSTM 是 LSTM 网络的双向版本, 能够捕获文本的正序与倒序信息。给定时序输入 $(x_1, x_2,\cdots, x_n)$, 针对每一时刻 t, LSTM 网络先计算输入门和输出门输出,

$$i_t=\mathrm{Sigmoid}(W_{xi}x_t+W_{hi}h_{t-1}+b_i), \tag{1}$$

$$o_t=\mathrm{Sigmoid}(W_{xo}x_t+W_{ho}h_{t-1}+b_o), \tag{2}$$

其中, h_{t-1} 为上一时刻的隐藏层输出, W 和 b 分别为每个控制门对应的参数, 后续不再赘述。Sigmoid 为激活函数, 定义为

$$\mathrm{Sigmoid}(x)=\frac{1}{1+\mathrm{e}^{-x}}, \tag{3}$$

之后计算遗忘门输出 f_t 和候选记忆细胞输出 g_t,

$$f_t = \text{Sigmoid}(W_{xf}x_t + W_{hf}h_{t-1} + b_f), \quad (4)$$

$$g_t = \text{Tanh}(W_{xg}x_t + W_{hg}h_{t-1} + b_g), \quad (5)$$

其中, Tanh 为激活函数, 定义为

$$\text{Tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \quad (6)$$

最后利用先前的输出值更新记忆细胞 c_t 与隐藏层状态 h_t, 其中符号 ⊙为哈达玛积。

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \quad (7)$$

$$h_t = o_t \odot \text{Tanh}(c_t) \quad (8)$$

至此当前时刻所有计算已完成, 其结果会传入下一时刻辅助计算, 然而这只是正向传递, BiLSTM 增加了另一套反向计算网络, 计算时需将文本倒序输入, 将其得到的隐藏层状态与正向网络的拼接即为捕获的语义特征, 这些语义信息随后被传入到 CRF 层。

CRF 模型是一种无向图模型, 常被用在标注和分析序列任务中。在利用 CRF 进行实体识别任务时有两个分数概念, 其中发射分数代表当前词标注为某一标签的概率, 转移分数则表示从某一个标签转变为另一个标签的概率。设标签空间大小为 N, 对于一个输入序列$(x_1, x_2, \cdots, x_n)$和一个标签序列$(T_1, T_2, \cdots, T_n)$, 该标签序列对于该输入序列的总分数 $S = \sum_{i=1}^{n} \text{Em}(x_i, T_i) + \text{Tr}(T_i, T_{i+1})$, 其中 $\text{Em}(x_i, T_i)$ 表示输入 x_i 发射为标签 T_i 的发射分数, $\text{Tr}(T_i, T_{i+1})$ 表示从标签 T_i 转移为标签 T_{i+1} 的转移分数, 这里标签 T_{n+1} 为人为添加的标签, 为 N 号, 表示 END, 意为句子结尾。利用 CRF 进行标签标注, 就是在所有可能的标签序列空间中找到对输入序列总得分最大的标签序列。算法 1 中的维特比算法是解决该问题的有效手段, 为了便于计算, 对输入序列首尾分别添加了两个标签, 1 号 START 表示句子开头, N 号 END 表示句子结尾。该算法结合动态规划思想利用数组 s_i 保存 i 时刻以每一类标签结束的标签序列的最大分数, 用标签转移路径数组 p_i 保存当前时刻以每类标签结束的最大分数是由上一时刻哪类标签转化而来的, 便于后续序列的构建, 具体的计算过程由算法 1 给出。

算法 1. 维特比算法

输入: 待标签序列$(x_1, x_2, \cdots, x_n)$, 标签个数 N

输出: 输入对应的最优标签序列$(T_1, T_2, \cdots, T_n)$

1) FOR $j = 1$ TO N DO

2) 初始化数组 $s_0[j] = -\infty$

3) 初始化 START 标签分数, $s_0[1] = 0$

4) FOR $i = 1$ TO n DO

5) FOR $j = 1$ TO N DO

6) FOR $k = 1$ TO N DO

7) $\text{tmp}[k] = s_{i-1}[k] + \text{Em}(x_i, j) + \text{Tr}(k, j)$

8) 更新最大分数数组 $s_i[j] = \max(\text{tmp})$

9) 更新路径数组 $p_i[j] = \arg\max(\text{tmp})$

10) FOR $j = 1$ TO N DO

11) $s_{n+1}[j] = s_n[j] + Tr(j, N)$

12) $T_n = \arg\max(s_{n+1})$, path.add(T_n)

13) FOR $i = n - 1$ TO 1 DO

14) $T_i = p_{i+1}[T_{i+1}]$, path.add(T_i)

15) 反转 path, 输出最终结果$(T_1, T_2, \cdots, T_n)$

在 BiLSTM-CRF 模型中, CRF 层每一时刻的发射分数是由该时刻 BiLSTM 的隐藏层输出经由一个全连接层得到的。CRF 层在模型训练时, 可以自动学习到一个标签转移矩阵, 从而能够提供转移分数。由此, 实现安全的 BiLSTM-CRF 模型的关键在于如何进行 BiLSTM 神经网络的隐私计算和 CRF 维特比算法的隐私计算, 后续本文将针对这些问题给出相应的解决方案。

3 问题描述

BiLSTM-CRF 模型以其结合了深度学习和统计机器学习的优点, 已经成为当前命名实体识别领域的基线方法。当模型的持有方想要对外提供命名实体识别服务时, 出于计算资源短缺的考虑, 其可能将推理任务的计算外包给云。在这个过程中, 模型以明文形式暴露给云, 从而造成了模型隐私泄露问题。除此之外, 使用该命名实体识别服务的用户, 需要将自己的文本数据上传给云来获取推理结果, 将其以明文形式上传无疑损害了用户的个人隐私。为了解决上述的隐私泄露问题, 本文提出了第一个基于秘密共享的隐私保护 BiLSTM-CRF 模型, 简称 SecNER。SecNER 可以保证在上述外包推理场景下, 模型持有方的模型安全以及用户使用方的数据安全。具体而言, 用户在本地利用秘密共享对待识别数据 d 进行加密得到密文

$$\text{cip} = \text{Enc}(d), \quad (9)$$

模型持有者在本地利用秘密共享对模型 $\boldsymbol{M}$ 进行加密, 得到密文

$$\mathcal{M} = \text{Enc}(\boldsymbol{M}), \quad (10)$$

用户和模型持有者将密文上传给云服务商, 云服务商在密文域下完成推理计算, 得到结果密文

$$\text{cip\_ans} = f_{\mathcal{M}}(\text{cip}), \tag{11}$$

最终, 用户本地解密, 得到明文推理结果

$$\text{ans} = \text{Dec}(\text{cip\_ans}), \tag{12}$$

可以看到整个推理过程中, 模型、待识别数据、推理结果都是密文, 进而可以保证数据安全。

3.1 系统架构

如图 1 所示, SecNER 中共有三类实体, 分别为模型提供方、云服务器以及用户。各个实体的作用如下。

用户: 用户为命名实体识别模型的使用方, 用户在本地利用模型公开的字典, 将自己需要识别的文本中的每个单词转化为一个独热向量, 之后用户对这些向量利用秘密共享加密为两份, 分别发送给两个云服务器 CS_1 和 CS_2。

云服务器: SecNER 采用了三个云服务器 $CS_{\{1,2,3\}}$ 的架构, 这样的架构在很多安全系统中都被采用[24-26], 本文借鉴了他们的思想并紧随目前的研究趋势。其中 CS_1 和 CS_2 承担了安全计算的主要计算工作, CS_3 只在某些具体的协议中起到辅助计算作用。在完成计算后, CS_1 和 CS_2 将结果的秘密共享份额发送给用户, 用户可本地恢复结果。

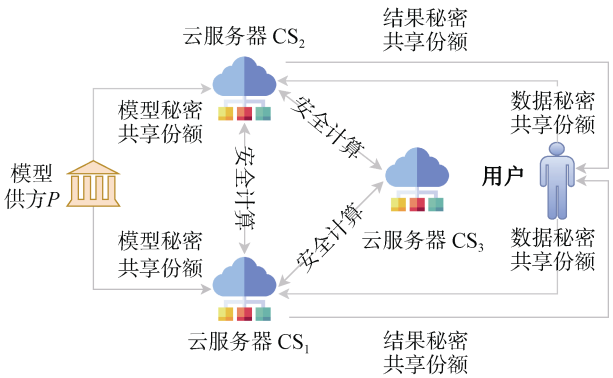

图 1 系统架构图

Figure 1 System architecture

模型提供者: 模型提供者代表拥有模型, 但不想承担计算任务的实体。其在本地将模型参数利用定点数表示, 采用秘密共享对其进行加密, 并将密文分别发送给 CS_1 和 CS_2 两个云服务器, 在完成模型分发过程后即可离线。

3.2 威胁模型

与先前的一些三方隐私保护机器学习模型相同[24,27-28], SecNER 采用了半诚实且不合谋的威胁模型, 即三个服务器会诚实地执行设计的协议, 且不会互相合谋, 但会在协议执行过程中尝试去学习到用户的隐私数据。这样的假设是合理的, 因为这些服务器可能来自不同的云厂商, 为了各自的名誉, 这些大品牌厂商不会轻易尝试与其他厂商合谋[21]。

3.3 设计目标

在上述威胁模型下, SecNER 旨在提供安全的 BiLSTM-CRF 命名实体识别推理服务, 其具体的设计目标如下。

安全性: 安全性包括模型安全与数据安全。模型安全指被托管的模型参数不被泄漏, 数据安全指待预测的用户数据不被泄漏。

高效性: 高效性指在安全推理过程中的通信开销与计算开销应尽可能小。

可用性: 可用性指在密文域下的推理结果的精度应与明文下的应尽可能接近。

3.4 方案应用模式

事实上, 本文设计的安全实体识别系统是一个独立的功能模块, 其输出结果可以应用于在很多自然语言处理任务中, 如机器翻译、信息检索。以信息检索系统为例, 用户可以利用该安全实体识别系统, 在不暴露自己文本情况下去识别出文本中的重要实体, 然后利用这些实体作为特征, 在自己的知识库中检索包含对应实体的知识, 从而缩小检索范围, 提升检索准确度。在这个过程中, 用户文本是加密的, 不会暴露给云服务商, 从而实现了安全检索。

4 SecNER 方案设计

4.1 设计概览

BiLSTM-CRF 模型分为词嵌入层、BiLSTM 层和 CRF 层, 要实现对 BiLSTM-CRF 模型的隐私保护, 即可拆分为对这三层分别进行隐私保护。由于明文下单词到词向量的词嵌入操作可以直接按照索引实现, 而在密文下这种转换是很难实现的, 为了能够更好地实现密文下单词到词向量的转换, 本文采用分桶的思想对模型的词嵌入层结构进行了调整, 从而能够加快这一过程。BiLSTM 层主要涉及矩阵乘法、哈达玛乘法, 以及激活函数 Sigmoid 和 Tanh 的计算。矩阵乘法和哈达玛乘法只涉及加法和乘法操作, 这些操作在秘密共享域中可以直接实现, 从而关键在于如何实现安全 Sigmoid 和 Tanh 的计算。为此, 本文采用了 Newton-Raphson 等近似拟合的方法, 将这些非线性函数转化为只涉及乘法和加法的操作, 进而完成了计算目标。在 CRF 层中, 主要涉及加法操作, arg max 和 max 操作, 其中加法是很容易实现的, 因此实现安全 CRF 层的关键在于如何实现安全 arg max 和安全 max 函数的计算。为此, 本文利用基

于并行前缀加法器的安全比较算法来实现这两个函数在秘密共享域下的计算。在 CRF 层的最后, 需要根据标签转移路径数组, 按照索引值不断去数组中检索数据, 依次构建出概率最大的标签序列。在明文域下, 这个操作是十分简单即可实现的, 但在秘密共享域中, 由于索引是秘密共享的, 数组同样也是秘密共享的, 所以无法按照明文下的操作直接进行检索。为了解决这个安全数组访问问题, 本文引入了一个辅助服务器 CS_3 来协助 CS_1 和 CS_2 完成计算。下面将给出本文的方案详细介绍。

4.2 安全的词向量获取

在利用 BiLSTM 网络进行计算之前, 首先需要做的是将单词转化为对应的词向量, 实现词嵌入操作。在明文域下, 该操作可利用索引直接实现。然而在秘密共享域下, 由于词向量矩阵此时已经被秘密共享加密分为了两份, 且用户不能直接将索引暴露给云服务商, 否则云服务商可根据索引值反推出用户单词, 实现该操作变得十分困难。一种直接的实现方式是, 模型的提供者将自己的字典排好序, 使得每个单词对应一个序号, 并将该字典公开发布给用户。用户利用该字典, 将自己的每一个单词转化为一个独热向量, 该独热向量的维度与字典中单词个数相同, 且在该向量中只有该单词对应的序号位置的值是 1, 其余位置的值都是 0。随后, 用户将该独热向量利用秘密共享技术加密分成两份, 分发给 CS_1 和 CS_2。CS_1 和 CS_2 交互进行秘密共享域下的矩阵乘法, 将独热向量和词向量矩阵做内积, 从而完成对应单词的词向量安全获取。这种方法的弊端在于字典中单词的个数非常多, 一般为几十万的级别, 且单词维度较大一般为 100 维以上, 这就导致词向量矩阵的规模很大, 对其进行秘密共享域下的乘法会造成巨大的通信开销, 方法可用性差。

针对上述问题, 本文对模型的词向量矩阵结构进行了调整, 设计了如算法 2 所示的安全词向量获取算法。算法采用分桶思想, 将原词向量矩阵中单词均匀分到 B 个桶中, 桶中单词依旧保持有序。模型拥有者将这些小的词向量矩阵利用秘密共享加密并分发给云, 公开字典。用户上传数据时, 对每个单词找到对应的桶, 并根据桶中序号构造独热向量, 将其加密后与桶号一起分发给云。此后, CS_1 和 CS_2 按桶号找到对应小的词向量矩阵的秘密份额, 交互进行独热向量和词向量矩阵在秘密共享域下的矩阵乘法。可以看到由于桶中字典大小变为了原来的 $1/B$, 则独热向量和词向量矩阵大小都变为了 $1/B$, 从而大大减少了通信开销。除此之外, 如果多个单词同属一个桶, 则可以利用相关 Beaver 三元组[22]来辅助计算乘法, 对于相同的词向量矩阵可只掩码一次进一步减少通信开销。

算法 2. 安全词向量获取

输入: 总词向量矩阵 $\boldsymbol{V}$, 用户的输入单词 w

输出: w 对应词向量的秘密共享

1) 模型持有者将 $\boldsymbol{V}$ 均匀分到 B 个桶中, 利用秘密共享加密为 $\{\langle \boldsymbol{V}_1\rangle, \langle \boldsymbol{V}_2\rangle, \cdots, \langle \boldsymbol{V}_B\rangle\}$

2) 为每个桶构建字典, 生成 $\{D_1, D_2, \cdots, D_B\}$

3) 模型持有者分发词向量矩阵秘密共享份额, 并公开字典, 完成初始化

4) 用户按字典将单词转化为独热向量, 利用秘密共享加密为 $\langle O_w\rangle$, 与桶号 idx 一起分发给云

5) CS_1 和 CS_2 交互计算秘密共享下矩阵乘法 $\langle O_w\rangle \cdot \langle \boldsymbol{V}_{\text{idx}}\rangle$ 完成提取

4.3 安全的 BiLSTM

安全的 BiLSTM 将词向量的秘密共享作为输入, 在秘密共享域计算每一单词的隐藏层状态。这些隐藏层状态经由全连接层映射到标签类别空间, 可作为发射分数传入下一层。如前所述, 实现安全的 BiLSTM 的关键在于如何安全地计算激活函数 Sigmoid 和 Tanh。本文用 SecSigmoid 与 SecTanh 表示这两个函数在秘密共享域下的计算。对于 SecSigmoid, 本文借鉴了文献[29]的思想, 按照先近似 e^{-x}, 再用 Newton-Raphson 方法拟合倒数的方式完成计算。详细地, 激活函数 $\mathrm{Sigmoid}(x)=1/(1+\mathrm{e}^{-x})$ 的计算可以分两步完成, 首先计算 $1+\mathrm{e}^{-x}$, 其次计算 $1+\mathrm{e}^{-x}$ 的倒数。$1+\mathrm{e}^{-x}$ 可以采用如下公式进行近似

$$1+\mathrm{e}^{-x}=1+\lim_{n\to\infty}\left(1+\frac{-x}{2^n}\right)^{2^n}, \tag{13}$$

公式中只涉及加法和乘法运算, 因而可以在秘密共享域下直接计算。下面需要计算 $1+\mathrm{e}^{-x}$ 的倒数。一般地, 对于一个正数输入 u, u 的倒数可以采用 Newton-Raphson[30]方法迭代拟合, 首先选定一个初始值 y_0, 并按照如下公式迭代计算

$$y_{n+1}=y_n(2-uy_n), \tag{14}$$

则极限 $\lim_{n\to\infty} y_n=1/u$, 需要注意的是该极限收敛的条件是 $0<y_0<2/u$。由此, 如何设置 y_0 是拟合 $1+\mathrm{e}^{-x}$ 倒数的关键。当 $x\geq 0$ 时, $1+\mathrm{e}^{-x}\leq 2$, 则 $2/(1+\mathrm{e}^{-x})\geq 1$, 从而当 $x\geq 0$ 时, 可设置 $y_0=0.75$。当

$x<0$ 时, 很难由 $1+\mathrm{e}^{-x}$ 估计 y_0 的值, 此时无法直接拟合 $1+\mathrm{e}^{-x}$ 的倒数, 也就无法计算 $\mathrm{Sigmoid}(x)$。为了解决这个问题, 可以利用转换等式 $\mathrm{Sigmoid}(x)=1-\mathrm{Sigmoid}(-x)$ 将其转化为输入为正数 $-x$ 的场景, 由于正数是可以拟合的, 从而可以实现对输入为负数场景的计算。

定义 $\mathrm{sign}(x)$ 表示 x 的正负值, 如果 $x\geq 0$, $\mathrm{sign}(x)=1$, 否则 $\mathrm{sign}(x)=-1$。定义 $\mathrm{msb}(x)$ 为 x 的最高有效位, $x\geq 0$, $\mathrm{msb}(x)=0$, 否则 $\mathrm{msb}(x)=1$。则有 $\mathrm{sign}(x)=1-2\mathrm{msb}(x)$。根据上述思想, 计算任意输入的 $\mathrm{Sigmoid}(x)$ 可先计算 $|x|=\mathrm{sign}(x)\cdot x$, 考虑到如下等式

$$\begin{aligned}\mathrm{Sigmoid}(x)=&(1-\mathrm{msb}(x))\cdot\mathrm{Sigmoid}(|x|)\\&+\mathrm{msb}(x)\cdot(1-\mathrm{Sigmoid}(|x|),\end{aligned}\tag{15}$$

进而有

$$\begin{aligned}\mathrm{Sigmoid}(x)=&\mathrm{Sigmoid}(|x|)+\mathrm{msb}(x)\\&-2\mathrm{msb}(x)\cdot\mathrm{Sigmoid}(|x|),\end{aligned}\tag{16}$$

从而可将 $\mathrm{Sigmoid}(x)$ 转化为 $\mathrm{Sigmoid}(|x|)$ 的计算, 解决了输入为负数的问题。上述计算过程只涉及了加法乘法, 以及最高有效位提取操作, 其中加法乘法可在秘密共享域下直接实现。最高有效位提取操作在秘密共享域下的实现, 可先用文献[19]提出的安全并行前缀加法器获取输入 x 最高有效位的布尔共享 $[\![\mathrm{msb}(x)]\!]$, 再利用文献[21]提出的 B2A 函数将布尔共享转化为算数共享 $\langle\mathrm{msb}(x)\rangle$。由此可以实现在秘密共享域下计算 $\mathrm{Sigmoid}(x)$ 函数。算法 3 给出了 SecSigmoid 整个计算流程。

算法 3. SecSigmoid

输入: 秘密共享 $\langle x\rangle$

输出: 秘密共享 $\langle\mathrm{Sigmoid}(x)\rangle$

1) CS_1 和 CS_2 利用安全并行前缀加法器计算 $[\![\mathrm{msb}(x)]\!]$, 并利用 B2A 函数将其转化为加性秘密共享 $\langle\mathrm{msb}(x)\rangle$

2) CS_1 和 CS_2 计算 $\langle\mathrm{sign}(x)\rangle=1-2\langle\mathrm{msb}(x)\rangle$

3) CS_1 和 CS_2 计算 $\langle|x|\rangle=\langle\mathrm{sign}(x)\rangle\cdot\langle x\rangle$

4) CS_1 和 CS_2 计算 $\langle t\rangle=1-\dfrac{\langle|x|\rangle}{2^{n_1}}$

5) FOR $i=1$ TO n_1 DO

6) CS_1 和 CS_2 计算 $\langle t\rangle=\langle t\rangle\cdot\langle t\rangle$

7) CS_1 和 CS_2 计算 $\langle t\rangle=1+\langle t\rangle$, $\langle y_0\rangle=\langle 0.75\rangle$

8) FOR $i=1$ TO n_2 DO

9) 迭代计算 $\langle y_i\rangle=\langle y_{i-1}\rangle\cdot(2-\langle t\rangle\cdot\langle y_{i-1}\rangle)$

10) 完成计算 $\langle\mathrm{Sigmoid}(x)\rangle=\langle y_{n_2}\rangle+\langle\mathrm{msb}(x)\rangle-2\cdot\langle\mathrm{msb}(x)\rangle\cdot\langle y_{n_2}\rangle$

由于存在公式 $\mathrm{Tanh}(x)=2\mathrm{Sigmoid}(2x)-1$, 所以安全 SecTanh 的计算可以转化为 SecSigmoid 的计算, 本文不再赘述。且除激活函数外, BiLSTM 中其余操作都是线性的, 其在秘密共享域下可以直接计算, 至此, 安全 BiLSTM 中所有的算子都已具备, 安全 BiLSTM 设计已经完成。

在完成 BiLSTM 隐藏层的安全计算后, 需要将隐藏层的输出输入到一个全连接层, 将其映射到标签类别空间。由于全连接层的操作只涉及矩阵乘法和加法, 因此可以在秘密共享域中直接实现, 全连接层的输出可以看作每一时刻对应的发射分数的秘密共享, 该分数被传入 CRF 层进行后续的计算。

4.4 安全的 CRF

4.4.1 安全的数组最大值信息获取

安全的 CRF 利用收到的发射分数与模型参数中的标签转移矩阵在秘密共享域下完成对输入序列的最优标注。在 CRF 层中, 主要运行维特比算法来构建最优标签序列, 如算法 1 所示, 维特比算法可分为前向计算最大分数和反向构建最大序列两阶段。第一阶段主要涉及加法、乘法、数组最大值 max 和数组最大值索引 arg max 的计算。因此, 如何在秘密共享域下实现安全的 max 和 arg max 是问题解决的关键。本文先实现两个元素的安全最大值算法, 再将其扩展到数组上。用 p 表示两个数 a 和 b 的大小关系, $a\geq b$, $p=1$, 否则有 $p=0$。则有

$$\max(a,b)=p\cdot a+\neg p\cdot b,\tag{17}$$

其中, 公式中 p 的计算可利用文献[19]中的安全比较函数在秘密共享域下进行。该安全比较函数针对输入的秘密共享 $\langle a\rangle$ 和 $\langle b\rangle$ 可计算出布尔共享 $[\![p]\!]$ 与 $[\![\neg p]\!]$, 接着利用 B2A 函数将其转化为算数共享。由此公式(17)可在秘密共享域下计算, 进而可实现两个元素的安全最大值。安全 arg max 的计算也是类似, 只不过是将公式(17)中的 a 和 b 的值替换为对应的索引值, 不再赘述, 且其与安全 max 可同时计算, 本文将二者合并到一个协议中, 称为安全最大值信息获取。

在将上述计算扩展到数组时, 一种直接的方式是依次对数组中的元素逐个进行比较, 可以看到比较的轮数为 $L-1$, 其中 L 为数组长度, 由于安全比

较操作需要云服务器通信交互完成, 当数组较长时, 比较的轮数较大, 进而通信轮数较大。为此, 本文采用构造比较树的思想对其进行优化, 令数组中的元素先两两比较, 胜者进入下一轮, 在下一轮中再两两比较, 依次不断进行下去, 直到只剩一个元素。一个简单的比较示例如图 2 所示, 其中数对第二维为索引值。可以看到通过这种方式, 比较的轮数降为 log(*L*), 大大提升了效率。

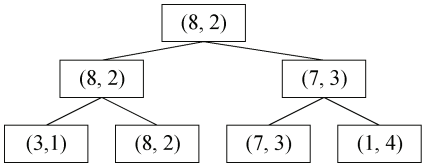

图 2　简单的比较示例

Figure 2　A simple comparison example

4.4.2　安全的数组访问

在维特比算法的最后(算法 1 第 14 行), 需利用索引去数组中检索对应的值, 并将获取到的值当作新的索引去下一个数组中继续检索, 以此构建出最优的序列。然而在秘密共享域下, 由于数组是秘密共享的, 索引也是秘密共享的, 直接利用索引去检索元素是不可实现的。为了解决这个问题, 本文借鉴了文献[28]的方法, 引入第三个服务器 CS_3 来辅助计算, 从而实现安全数组访问 SecAccess。给定秘密共享的数组 $\langle \boldsymbol{a} \rangle$ 和秘密共享的索引 $\langle y \rangle$, SecAccess 可在不泄露 y 的前提下输出秘密共享值 $\langle \boldsymbol{a}[y] \rangle$。

初始化时, 三个云服务器 $\mathrm{CS}_i (i \in \{1,2,3\})$, 分别生成密钥 k_i, 并将 k_i 发送给 CS_{i+1}, 在这里定义 $\mathrm{CS}_{3+1} = \mathrm{CS}_1$, 则可构造 CS_i 的相关随机值数组 c^i,数组中第 j 个元素为 $c^i[j] = F_{k_i}(j) - F_{k_{i-1}}(j)$, 其中 F 为伪随机函数, $k_{1-1} = k_3$。任意两个云服务器还可根据其共有的密钥构造一致随机值 r^i, 定义服务器 CS_i 和 CS_{i+1} 的第 j 个一致随机值 $r_j^i = F_{k_i}(j) \bmod \ell$, 其中 ℓ 表示秘密共享数组的长度。接下来三个服务器, 按如下步骤完成计算。

(1) CS_1 本地完成下述操作。对其拥有的秘密共享数组份额 $\langle \boldsymbol{a} \rangle^1$ 顺时针旋转 r^1 个位置, 从而得到 $\langle \boldsymbol{a} \rangle^1 = \langle \boldsymbol{a} \rangle^1 \circlearrowright r^1$。利用相关随机值数组 c^1 对其掩码, 得到 $\langle \boldsymbol{a'} \rangle^1 = \langle \boldsymbol{a} \rangle^1 + c^1$, 再对 $\langle \boldsymbol{a'} \rangle^1$ 顺时针旋转 r^3 个位置得到 $\langle \boldsymbol{a''} \rangle^2 = \langle \boldsymbol{a'} \rangle^1 \circlearrowright r^3$, 随后利用 r^1 与 r^3 对索引进行掩码有 $\langle t \rangle^1 = \langle y \rangle^1 + r^1 + r^3$。最后将 $\langle t \rangle^1$ 与 $\langle \boldsymbol{a''} \rangle^2$ 发送给 CS_2。

(2) CS_2 本地完成下述操作。恢复索引值 t, 计算 $t = (\langle t \rangle^1 + \langle y \rangle^2) \bmod \ell$, 用 t 检索 $\mathrm{target} = \langle \boldsymbol{a''}[t] \rangle^2$, 对秘密共享数组份额 $\langle \boldsymbol{a} \rangle^2$ 顺时针旋转 r^1 位置, $\langle \boldsymbol{a} \rangle^2 = \langle \boldsymbol{a} \rangle^2 \circlearrowright r^1$, 用 c^2 掩码得到 $\langle \boldsymbol{a'} \rangle^2 = \langle \boldsymbol{a} \rangle^2 + c^2$。最后将 $\langle \boldsymbol{a'} \rangle^2$ 和 t 发送给 CS_3。

(3) CS_3 本地完成下述操作。用 c^3 对 $\langle \boldsymbol{a'} \rangle^2$ 掩码, 得到 $\langle \boldsymbol{a''} \rangle^3 = \langle \boldsymbol{a'} \rangle^2 + c^3$, 对 $\langle \boldsymbol{a''} \rangle^3$ 顺时针旋转 r^3 位置, 利用 t 检索 $\mathrm{aim} = \langle \boldsymbol{a''}[t] \rangle^3$, 随机生成 s, 发送 s 给 CS_1, 发送 $\mathrm{aim} - s$ 给 CS_2。则 s 与 $\mathrm{aim} - s + \mathrm{target}$ 即为 $\boldsymbol{a}[y]$ 的秘密份额。

至此, 维特比算法安全计算的所有安全算子都已具备, 只需按照算法 1 中的步骤在秘密共享域下依次计算即可, 本文不再赘述具体的执行细节。

5　安全性证明

目前, 在隐私保护机器学习领域没有具体的指标来衡量系统的安全性。因此, 本文基于模拟的安全分析方法对 SecNER 的安全性进行证明, 该分析方法被广泛用于证明隐私保护机器学习模型的安全性[2,20,31-32]。首先, 形式化地定义一个安全利用 BiLSTM-CRF 模型来进行命名实体识别的理想功能函数 $\mathcal{F}$ 如下。

输入: 用户向理想函数 F 输入自己单词对应的桶号及独热向量 $\{(b_1, o_1), \cdots, (b_n, o_n)\}$, 模型提供者输入 BiLSTM-CRF 模型 $\boldsymbol{M}$

计算: 收到 $\{(b_1, o_1), \cdots, (b_n, o_n)\}$ 和模型 $\boldsymbol{M}$ 后, F 利用模型进行命名实体识别推理

输出: F 将文本对应的标签序列返回用户

令 Π 表示实现理想功能 F 的协议, 则 Π 的安全性可由定义 1 所定义, 为了简化证明过程, 采用引理 1[32] 、引理 2[20]和引理 3[22]来辅助证明。

定义 1.令 view_b^{Π} 表示在执行协议 Π 的过程中每个 $\mathrm{CS}_b (b \in \{1,2,3\})$ 的视角, 则可以说协议 Π 在三方半诚实不合谋场景下是安全的, 如果对于每个 CS_b 存在一个概率多项式时间的模拟器, 使得 $S_b^{\Pi} \triangleq \mathrm{view}_b^{\Pi}$, 其中 S_b^{Π} 是由模拟器生成的 CS_b 的模拟视角, $\triangleq$ 表示不可区分。

引理 1.如果一个系统的所有子模块都是可模拟的, 那么整个系统也可模拟。

引理 2.在半诚实且不合谋的场景下, 秘密共享

乘法 SecMul 和秘密共享加法 SecAdd 是可模拟的。

引理 3.在半诚实且不合谋的场景下, 用相关 Beaver 三元组进行秘密共享乘法是可模拟的。

定理 1.在半诚实且不合谋的场景下, SecNER 能够按照定义 1 中的安全性实现理想功能函数 F。

证明. 可以看到, SecNER 共涉及了五个子模块, 即安全词向量获取(SWE)、安全 Sigmoid (SecSigmoid)、安全 Tanh (SecTanh)、安全最大值信息获取(Secmax)、安全数组访问(SecAccess)。根据引理 1, 如果每个子模块的模拟器都存在, 则 SecNER 是安全的。定义符号 Sim^{b}_{Π} 为在子模块 Π 的执行过程中负责生成 CS_b 视角的模拟器。由于除 SecAccess 外, CS_1 和 CS_2 在其余协议中是对称的, 且 CS_3 并不参与这些协议, 因此对于除 SecAccess 外的其他协议, 只需证明 CS_1 的模拟器存在即可。以下将给出这些子模块的模拟器存在性分析。

(1) 模拟器 $\mathrm{Sim}^{1}_{\mathrm{SWE}}$ 的存在性。可以看到, 在 SWE 协议中, CS_1 只利用相关三元组进行秘密共享域下的矩阵乘法操作, 根据引理 3 与引理 1, 模拟器 $\mathrm{Sim}^{1}_{\mathrm{SWE}}$ 存在。

(2) 模拟器 $\mathrm{Sim}^{1}_{\mathrm{SecSigmoid}}$ 的存在性。可以看到, 在 SecSigmoid 协议中, CS_1 利用了安全并行前缀加法器提取了布尔秘密共享$[\![\mathrm{msb}(x)]\!]$, 并利用 B2A 函数将其转化为加性秘密共享。安全并行前缀加法器和 B2A 函数在文献[19]与[21]中分别被证明是可模拟的。SecSigmoid 中其他的中间结果, 都是利用 SecMul 和 SecAdd 计算得到的, 根据引理 2, 这些中间结果也是可模拟的, 进而模拟器 $\mathrm{Sim}^{1}_{\mathrm{SecSigmoid}}$ 存在。

(3) 模拟器 $\mathrm{Sim}^{1}_{\mathrm{SecTanh}}$ 的存在性。由于在进行计算 SecTanh 时, 本文用公式 $\mathrm{Tanh}(x)=2\mathrm{Sigmoid}(2x)-1$ 将其转化为了 SecSigmoid 的计算, 由于模拟器 $\mathrm{Sim}^{1}_{\mathrm{SecSigmoid}}$ 存在, 进而模拟器 $\mathrm{Sim}^{1}_{\mathrm{SecTanh}}$ 存在。

(4) 模拟器 $\mathrm{Sim}^{1}_{\mathrm{Secmax}}$ 的存在性。在计算 Secmax 时, 主要利用到了安全比较函数, 该函数在文献[19]被证明为可模拟的, Secmax 中的其他中间结果都是利用 SecMul 和 SecAdd 计算得到的, 根据引理 2, 这些中间结果也是可模拟的, 进而模拟器 $\mathrm{Sim}^{1}_{\mathrm{Secmax}}$ 存在。

(5) 模拟器 $\mathrm{Sim}^{1}_{\mathrm{SecAccess}}$ 的存在性。CS_1 在协议的执行过程中收到了来自 CS_3 的 s, 由于 s 是由 CS_3 随机生成的, 进而其在 CS_1 视角是均匀随机分布的, 可以被模拟, 进而模拟器 $\mathrm{Sim}^{1}_{\mathrm{SecAccess}}$ 存在。

(6) 模拟器 $\mathrm{Sim}^{2}_{\mathrm{SecAccess}}$ 的存在性。在协议执行开始前, CS_2 拥有秘密共享 $\langle \boldsymbol{a}\rangle^2$ 和 $\langle y\rangle^2$ 以及密钥 k_1 和 k_2。之后 CS_2 从 CS_1 接收到了 $\langle t\rangle^1$ 和 $\langle \boldsymbol{a}''\rangle^2$。由于 $\langle \boldsymbol{a}'\rangle^2=(\langle \boldsymbol{a}\rangle^1 \circlearrowleft r^1+c^1)\circlearrowleft r^3$, 且掩码数组中元素 $c^1[j]=F_{k_1}(j)-F_{k_3}(j)(j\in[1,\ell])$, 尽管 CS_2 拥有 k_1 其没有 k_3, 所以 $F_{k_3}(j)$ 在其视角是均匀随机分布的, 考虑到 $F_{k_1}(j)$ 与 $F_{k_3}(j)$ 是独立生成的, 进而 $c^1[j]$ 在其视角下也是均匀随机分布的。又由于 $\langle \boldsymbol{a}\rangle^1$ 与 c^1 是独立的, 则 $\langle \boldsymbol{a}''\rangle^2$ 在 CS_2 视角也是均匀随机分布的, 从而 $\langle \boldsymbol{a}''\rangle^2$ 可以被模拟。同时, $\langle t\rangle^1=\langle y\rangle^1+r^1+r^3$, 由于 r^3 是利用密钥 k_3 所生成的, CS_2 没有 k_3, 进而 r^3 在其视角是均匀随机分布的, 又由于 r^3 与 r^1 和 $\langle y\rangle^1$ 独立, 从而 $\langle t\rangle^1$ 在其视角是均匀随机分布的, 可以被模拟。最后 CS_2 从 CS_3 接收到 $\mathrm{aim}-s$, 其中 s CS_3 随机生成的, 且与 aim 无关, 则 $\mathrm{aim}-s$ 在 CS_2 视角是均匀随机的, 可以被模拟。综上, 模拟器 $\mathrm{Sim}^{2}_{\mathrm{SecAccess}}$ 存在。

(7) 模拟器 $\mathrm{Sim}^{3}_{\mathrm{SecAccess}}$ 的存在性。可以看到, 在执行过程中 CS_3 从 CS_2 处得到 $\langle \boldsymbol{a}'\rangle^2$ 与 t。与之前的证明相似, 由于 CS_3 没有密钥 k_1, 从而 $\langle \boldsymbol{a}'\rangle^2$ 与 t 在 CS_3 视角下是均匀随机的, 可以被模拟, 进而模拟器 $\mathrm{Sim}^{3}_{\mathrm{SecAccess}}$ 存在。

至此完成了对定理 1 的证明, 从而证明了 SecNER 的安全性。

6 实验评估

6.1 实验设置

本文使用 Python 3.8.13 编程实现了 SecNER 模型, 所有实验在一台拥有 16 核 Intel(R) Xeon(R) Gold 6130 CPU @2.10GHzCPU、256GB RAM 和 Ubuntu 16.04 操作系统的机器上运行。与现有的相关工作一致[19-20], SecNER 中云服务器的网络环境设置为局域网, 为了更接近真实结果, 本文的实验结果已经将三个不合谋云服务提供商之间的网络延迟考虑在内。具体参照文献[20]的设置, 将其网络带宽和延迟分别设置为 1 GB/s 和 0.17 ms, 实验中使用 Linux 文件系统来模拟局域网环境下的通信。在统计运行时间时, 本文将云服务商之间的通信轮数乘以通信时延, 作为网络延迟的开销, 并将其加到最终的运行耗时中。

秘密共享域的大小设置为 $\mathbb{Z}_{2^{64}}$, 并将缩放因子 q

设置为 16。在计算 SecSigmoid 和 SecTanh 时, 公式(13)和公式(14)的迭代次数分别为 6 和 3。本文利用 Pytorch 1.12.0 进行 BiLSTM-CRF 模型的训练, 分桶时词向量的桶数为 100, BiLSTM 网络的结构参数和训练细节如表 1 所示。

表 1 BiLSTM 网络的结构参数和训练细节

Table 1 The parameters and training details of the BiLSTM network

隐藏层大小	学习率	字典数	词向量维度	网络层数
64	0.001	400000	100	1

实验采用的数据集为命名实体识别任务中常用的三个数据集, 分别为 Financial NER 数据集简称 FIN, MIT-Restaurant 数据集简称 RES, 以及 Conll2003 数据集, 简称 CON。表 2 给出了每个数据集的具体信息。

表 2 数据集信息

Table 2 The information of the datasets

数据集	训练样本数	预测样本数	实体类别
FIN	1164	303	4
RES	6900	1521	8
CON	14041	3453	4

6.2 子模块性能

考虑到本文采用了近似的方式去拟合 Sigmoid 和 Tanh 函数, 为了探究这些近似对精度的影响, 本文绘制了在[–20,20]区间内 SecSigmoid 和 SecTanh 的函数图像, 并将其与明文下的函数图像进行了对比, 其结果如图 3-4 所示。可以看到无论 SecSigmoid 还是 SecTanh, 其图像都与明文下的图像近似重合, 可见本文的近似计算方法的精确度是很高的。除此之外, 本文还测试了 SecSigmoid 和 SecTanh 在不同执行次数下的平均耗时和平均绝对误差, 其结果如图 5 和图 6 所示。可以看到 SecSigmoid 和 SecTanh 的单词平均执行时间近乎相同, 均为 7ms 左右, 由于本文是将 SecTanh 转化为 SecSigmoid 来进行计算的, 这符合预期。此外, SecSigmoid 的平均绝对误差约为 6.8×10^{-4}, SecTanh 的平均绝对误差约为 7.0×10^{-4}, SecTanh 的精度相比 SecSigmoid 稍差一点, 但总体而言安全激活函数的精度是很高的。

6.3 SecNER 性能

由于并没有先前的工作对 BiLSTM-CRF 进行外包推理场景下的隐私保护, 本文没有直接对比的方案, 因此实验主要评估本文方案对模型推理开销及模型推理精度上的影响。实验首先评估了 SecNER 在三个数据集上对每个样本预测所需的时间和通信开销, 其结果如图 7 和图 8 所示。可以看到, 在三个数据集上每个样本的平均执行时间大约在 4s, 为一个可接受的范围内。

在通信开销方面, FIN 数据集的通信开销最大, 每个样本约 60MB, 其余两个数据集通信开销约为 30MB。造成这种现象的原因可能是 FIN 样本更长, 且其中单词对应的桶的分布更加分散, 从而需要掩码更多的词向量矩阵。但是总体来看, 对于局域网场景下, 通信开销负担很小, 可见本文对词向量矩阵结构的调整效果显著。

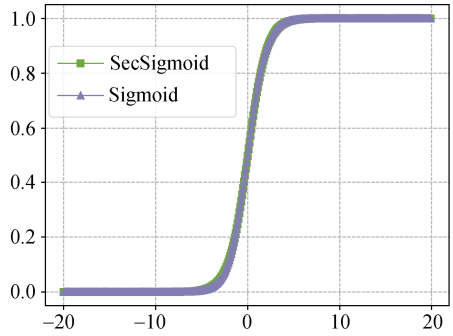

图 3 SecSigmoid 的函数图像

Figure 3 Graph for SecSigmoid function

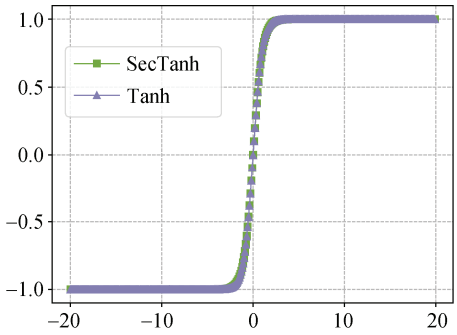

图 4 SecTanh 的函数图像

Figure 4 Graph for SecTanh function

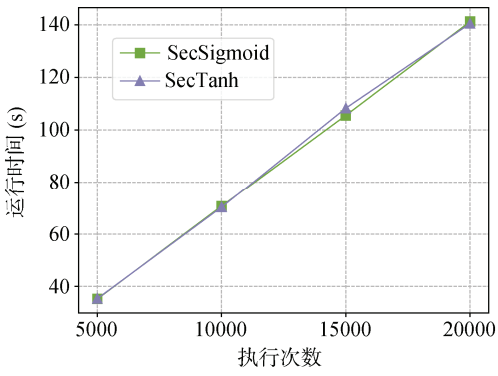

图 5 安全激活函数在不同执行次数下的运行时间

Figure 5 Running time of secure activation functions at different execution numbers

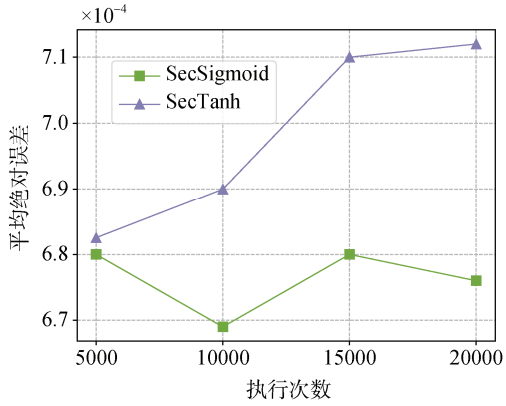

图 6 安全激活函数在不同执行次数下与明文结果的平均绝对误差

Figure 6 Average absolute error of secure activation functions compared to plaintext results at different numbers of executions

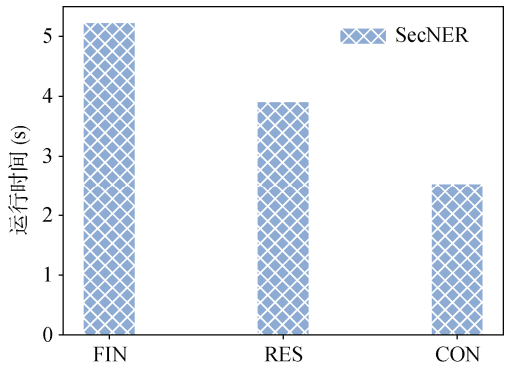

图 7 SecNER 在三个数据集上平均每个样本的执行时间

Figure 7 The average execution time of SecNER on three datasets per sample

除此之外, 本文对 SecNER 在三个数据集上的精度进行了测试, 并与明文下的模型精度进行了比较。表 3、表 4、表 5 分别给出了 SecNER 在 RES、CON、FIN 数据集上的精度性能表现。表中列出了每一类实体类别和总体的准确率、召回率和 F1 值。可以看到在 CON 和 FIN 数据集上, SecNER 的总体 F1 值与明文下持平, 并没有造成推理精度损失。在 RES 数据集上, 对设施类别实体的识别性能有轻微的下降, 但是总体 F1 值只有 0.001 的下降, 表明本文采用的隐私保护手段对模型精度影响较小, 不会影响模型的正常使用。

7 总结

本文提出了第一个基于秘密共享技术的安全 BiLSTM-CRF 命名实体识别推理模型 SecNER。SecNER 能够保证推理过程中模型参数安全与用户数据安全, 从而解决了命名实体识别外包推理中的隐私泄露问题。SecNER 对词嵌入层的结构进行了调整, 大大减少了安全词向量获取过程中的通信开销,

表 3 SecNER 在 RES 数据集上的性能表现

Table 3 Performance of SecNER on the RES dataset

实体类别	SecNER			BiLSTM-CRF		
	准确率	召回率	F1	准确率	召回率	F1
设施	0.698	0.632	0.663	0.711	0.636	0.671
菜肴	0.816	0.803	0.810	0.821	0.803	0.812
餐具	0.723	0.778	0.749	0.716	0.778	0.745
时间	0.715	0.675	0.694	0.711	0.675	0.693
地点	0.803	0.802	0.802	0.803	0.801	0.802
价格	0.822	0.784	0.802	0.822	0.784	0.802
评分	0.751	0.826	0.787	0.751	0.826	0.787
饭店名	0.764	0.781	0.772	0.763	0.779	0.771
总计	0.767	0.760	0.764	0.769	0.760	0.765

表 4 SecNER 在 CON 数据集上的性能表现

Table 4 Performance of SecNER on the CON dataset

实体类别	SecNER			BiLSTM-CRF		
	准确率	召回率	F1	准确率	召回率	F1
地名	0.871	0.892	0.881	0.871	0.892	0.881
杂项	0.751	0.701	0.725	0.752	0.697	0.723
机构名	0.808	0.775	0.791	0.813	0.774	0.793
人名	0.932	0.909	0.920	0.932	0.906	0.919
总计	0.856	0.838	0.847	0.858	0.837	0.847

表 5 SecNER 在 FIN 数据集上的性能表现

Table 5 Performance of SecNER on the FIN dataset

实体类别	SecNER			BiLSTM-CRF		
	准确率	召回率	F1	准确率	召回率	F1
地名	0.286	0.205	0.239	0.286	0.205	0.239
杂项	0.000	0.000	0.000	0.000	0.000	0.000
机构名	0.567	0.304	0.395	0.567	0.304	0.395
人名	0.943	0.912	0.927	0.943	0.912	0.927
总计	0.831	0.698	0.759	0.831	0.698	0.759

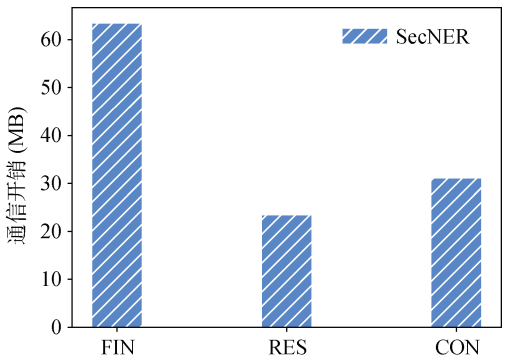

图 8 SecNER 在三个数据集上平均每个样本的通信开销

Figure 8 The average communication overhead of SecNER on three datasets per sample

采用了近似拟合的方式去计算非线性激活函数, 并采用比较树的形式优化了安全获取数组中最大元素信息所需的比较轮数。最后 SecNER 引入辅助服务器来完成 CRF 层中涉及的安全数组访问。本文对 SecNER 的安全性进行了形式化证明, 并设计了大量实验验证方案的可行性。实验结果表明, SecNER 在三个数据集上总体 F1 值最多下降 0.001, 且推理开销在可接受范围。

下一步工作中, 一是尝试对目前火热的大模型进行隐私保护, 进一步提高安全命名实体识别的精度; 二是探索设计在更加严格的恶意攻击威胁模型下的系统方案, 进一步提升系统的安全性。

参考文献

[1] Huang Z H, Xu W, Yu K. Bidirectional LSTM-CRF Models for Sequence Tagging[EB/OL]. 2015: arXiv: 1508.01991. https://arxiv.org/abs/1508.01991.

[2] Wang J, He D B, Castiglione A, et al. PCNNCEC: Efficient and Privacy-Preserving Convolutional Neural Network Inference Based on Cloud-Edge-Client Collaboration[J]. *IEEE Transactions on Network Science and Engineering*, 2023, 10(5): 2906-2923.

[3] Zheng Y F, Duan H Y, Tang X T, et al. Denoising in the Dark: Privacy-Preserving Deep Neural Network-Based Image Denoising[J]. *IEEE Transactions on Dependable and Secure Computing*, 2021, 18(3): 1261-1275.

[4] Tan Z W, Zhang L F. Survey on Privacy Preserving Techniques for Machine Learning[J]. *Journal of Software*, 2020, 31(7): 2127-2156. (谭作文, 张连福. 机器学习隐私保护研究综述[J]. *软件学报*, 2020, 31(7): 2127-2156.)

[5] Mohassel P, Rindal P. ABY^3: A Mixed Protocol Framework for Machine Learning[C]. *The 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018: 35-52.

[6] Agrawal N, Shahin Shamsabadi A, Kusner M J, et al. QUOTIENT: Two-Party Secure Neural Network Training and Prediction[C]. *The 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019: 1231-1247.

[7] Watson J-L, Wagh S, Popa R A. Piranha: A GPU platform for secure computation[C]. *USENIX Security Symposium*, 2022: 827-844.

[8] Al Badawi A, Hoang L, Mun C F, et al. PrivFT: Private and Fast Text Classification with Homomorphic Encryption[J]. *IEEE Access*, 2020, 8: 226544-226556.

[9] Kumar N, Rathee M, Chandran N, et al. CrypTFlow: Secure TensorFlow Inference[C]. *2020 IEEE Symposium on Security and Privacy*, 2020: 336-353.

[10] Xie B, Xiang T, Liao X F, et al. Achieving Privacy-Preserving Online Diagnosis with Outsourced SVM in Internet of Medical Things Environment[J]. *IEEE Transactions on Dependable and Secure Computing*, 2022, 19(6): 4113-4126.

[11] Niu C Y, Wu F, Tang S J, et al. Toward Verifiable and Privacy Preserving Machine Learning Prediction[J]. *IEEE Transactions on Dependable and Secure Computing*, 2022, 19(3): 1703-1721.

[12] Liu J, Juuti M, Lu Y, et al. Oblivious Neural Network Predictions *via* MiniONN Transformations[C]. *The 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017: 619-631.

[13] Feng Q, He D B, Liu Z, et al. SecureNLP: A System for Multi-Party Privacy-Preserving Natural Language Processing[J]. *IEEE Transactions on Information Forensics and Security*, 2020, 15: 3709-3721.

[14] Wang Q, Du M X, Chen X Y, et al. Privacy-Preserving Collaborative Model Learning: The Case of Word Vector Training[J]. *IEEE Transactions on Knowledge and Data Engineering*, 2018, 30(12): 2381-2393.

[15] Resende A, Railsback D, Dowsley R, et al. Fast Privacy-Preserving Text Classification Based on Secure Multiparty Computation[J]. *IEEE Transactions on Information Forensics and Security*, 2022, 17: 428-442.

[16] Reich D, Todoki A, Dowsley R, et al. Privacy-preserving classification of personal text messages with secure multi-party computation[J]. *Advances in Neural Information Processing Systems*, 2019, 32.

[17] Hua Z Y, Tong Y, Zheng Y F, et al. PPGloVe: Privacy-Preserving GloVe for Training Word Vectors in the Dark[J]. *IEEE Transactions on Information Forensics and Security*, 2024, 19: 3644-3658.

[18] Paillier P. Public-key cryptosystems based on composite degree residuosity classes[C]. *International conference on the theory and applications of cryptographic techniques*, 1999: 223-238.

[19] Liu X, Zheng Y, Yuan X, et al. Towards Secure and Lightweight Deep Learning as a Medical Diagnostic Service[C]. *European Symposium on Research in Computer Security*, 2021: 519-541.

[20] Mohassel P, Zhang Y P. SecureML: A System for Scalable Privacy-Preserving Machine Learning[C]. *2017 IEEE Symposium on Security and Privacy*, 2017: 19-38.

[21] Zheng Y, Duan H, Wang C. Towards secure and efficient outsourcing of machine learning classification[C]. *European Symposium on Research in Computer Security*, 2019: 22-40.

[22] Kelkar M, Le P H, Raykova M, et al. Secure poisson regression[C]. *USENIX Security Symposium*, 2022: 791-808.

[23] Beaver D. Efficient multiparty protocols using circuit randomization[C]. *Cryptology*, 1992: 420-432.

[24] Tan S J, Knott B, Tian Y, et al. CryptGPU: Fast Privacy-Preserving Machine Learning on the GPU[C]. *2021 IEEE Symposium on Security and Privacy*, 2021: 1021-1038.

[25] Araki T, Furukawa J, Ohara K, et al. Secure Graph Analysis at Scale[C]. *The 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021: 610-629.

[26] Boneh D, Boyle E, Corrigan-Gibbs H, et al. Lightweight Techniques for Private Heavy Hitters[C]. *2021 IEEE Symposium on Security and Privacy*, 2021: 762-776.

[27] Wagh S, Gupta D, Chandran N. SecureNN: 3-Party Secure Computation for Neural Network Training[J]. *Proceedings on Privacy Enhancing Technologies*, 2019, 2019(3): 26-49.

[28] Wang S L, Zheng Y F, Jia X H. SecGNN: Privacy-Preserving Graph Neural Network Training and Inference as a Cloud Ser-

vice[J]. *IEEE Transactions on Services Computing*, 2023, 16(4): 2923-2938.

[29] Knott B, Venkataraman S, Hannun A, et al. CrypTen: Secure Multi-Party Computation Meets Machine Learning[EB/OL]. 2021: arXiv: 2109.00984. https://arxiv.org/abs/2109.00984.

[30] Akram S, Ann Q U. Newton raphson method[J]. *International Journal of Scientific & Engineering Research*, 2015, 6(7): 1748-1752.

[31] Liu X N, Zheng Y F, Yuan X L, et al. Securely Outsourcing Neural Network Inference to the Cloud with Lightweight Techniques[J]. *IEEE Transactions on Dependable and Secure Computing*, 2023, 20(1): 620-636.

[32] Bogdanov D, Laur S, Willemson J. Sharemind: A framework for fast privacy-preserving computations[C]. *European Symposium on Research in Computer Security*, 2008: 192-206.

佟岩 于 2021 年在哈尔滨工业大学计算机科学与技术专业获得学士学位。现在哈尔滨工业大学(深圳)计算机技术专业攻读硕士学位。研究领域为隐私保护机器学习。Email: 594tongyan@gmail.com

花忠云 CCF 会员, 于 2016 年在澳门大学大学软件工程专业获得博士学位。现任哈尔滨工业大学(深圳)副教授。研究领域为云计算安全、多媒体信息安全、非线性系统理论。Email: huazhongyun@hit.edu.cn

廖清 CCF 会员, 于 2016 年在香港科技大学计算机科学与技术专业获得博士学位。现任哈尔滨工业大学(深圳)教授。研究领域为信息安全、人工智能。Email: liaoqing@hit.edu.cn

张玉书 CCF 会员, 于 2015 年在于重庆大学获得博士学位。南京航空航天大学教授, 博士生导师, 研究领域为多媒体安全、区块链等。Email: yushu@nuaa.edu.cn