基于胶囊网络的多任务少样本文本隐写分析

杨 雨 张梓葳 文 娟
(中国农业大学信息与电气工程学院 北京 100091)

摘 要 文本隐写分析是一种通过统计特征来区分载密文本和正常文本的技术.目前,最先进的文本隐写分析模型大多使用深度神经网络在单一任务上进行训练和测试.因此,现有模型在检测某种特定的隐写文本时有较好的性能.当待检测文本的领域、所使用的隐藏算法和嵌入容量发生变化时,模型的隐写分析性能会有一定程度的下降.为了增强文本隐写分析模型在不同检测任务上的快速自适应能力,并使模型能够处理少样本场景下的隐写分析任务,本文提出了一种基于胶囊网络的文本隐写分析方法.具体来说,使用带有自注意力的 Bi-LSTM(Bidirectional Long Short-Term Memory)作为通用任务提取器,从支持集和查询集中获取文本的句子表示;任务映射器作为元学习者主导元训练过程,在获取支持集的句子表示后,学习单个文本与任务间的非线性映射关系;然后,将映射结果和查询集的句子表示输入分类器,度量文本与任务之间的匹配程度;最后,基于均方误差 *MSE*(Mean Square Error)损失和 *KL* 散度(Kullback-Leibler Divergence)计算总预测损失.大量实验证明,我们的模型可以快速适应各种不同的任务,并在 1-shot、5-shot 和 10-shot 的检测任务中对三个域的平均检测精度分别达到了 85.11%、88.63%和 91.91%.

关键词 文本隐写分析;快速自适应;少样本学习;元学习;胶囊网络
中图法分类号 TP309 **DOI 号** 10.11897/SP.J.1016.2022.02592

Multi-Task Few-Shot Text Steganalysis Based on Capsule Network

YANG Yu ZHANG Zi-Wei WEN Juan
(*College of Information and Electrical Engineering, China Agricultural University, Beijing* 100091)

Abstract Text steganalysis is a technique to distinguish steganographic text from normal text using statistical features. Currently, the most advanced text steganalysis models are trained and tested on a single steganalysis task through deep neural structure and achieve excellent performance in detecting stego text generated by a specific steganography method with a certain embedding capacity in one kind of domain. However, when the target task changes (including the text domains, the steganographic algorithms used to generate the text, and the embedding capacity), the steganalysis performance of the model degrades to a certain extent. This paper proposes a capsule network-based approach for text steganalysis to enhance the model performance in different tasks, making the model achieve fast adaptation in few-shot scenarios. Specifically, we use a Bi-LSTM (Bidirectional Long Short-Term Memory) with a self-attention structure as a generic feature extractor to obtain sentence representations from the support set and query set. The task mapper guides the meta-training process as a meta-learner, learning a non-linear mapping relationship between a single text and a task after acquiring the sentence representations of the support set. After that, the mapping vector and the sentence representations of the query set are input to the classifier to obtain their matching degree. Finally, the total prediction loss composed of *MSE* and Kullback-Leibler Divergence losses is calculated. Extensive experiments demonstrate

收稿日期:2022-05-09;在线发布日期:2022-09-29.本课题得到国家自然科学基金项目(61802410)资助.**杨 雨**,硕士研究生,主要研究方向为信息安全、文本隐写分析. E-mail: S20203081453@cau.edu.cn. **张梓葳**,硕士研究生,主要研究方向为信息安全、文本隐写分析. **文 娟**(通信作者),博士,副教授,主要研究方向为自然语言处理、信息隐藏. E-mail: wenjuan@cau.edu.cn.

that our model can quickly adapt to various tasks and achieve the average detection accuracy of 85.11%, 88.63%, and 91.91% for the three domains under 1-shot, 5-shot, and 10-shot, respectively.

Keywords　text steganalysis; fast adaptation; few-shot learning; meta-learning; capsule network

1　引　言

加密系统、隐私系统和隐藏系统是网络空间中最基本的三种信息安全系统[1]. 隐藏系统不同于其他两种系统，它将机密信息嵌入到公共载体中，隐藏机密信息的存在，达到不容易被怀疑和攻击的目的[2-3]，在保证网络空间安全方面发挥着重要作用. 隐写术是构建隐藏系统的基本方法之一，旨在以不可察觉的方式将机密信息嵌入到载体中. 在网络空间中，可以用于信息隐藏的载体有视频[4]、图像[5-8]、音频[9-11]、文本[12-13]等. 文本是人们日常生活中最常用的信息媒介之一，对文本信息隐藏技术的研究具有极高的学术价值和实用价值. 因此，文本隐写术引起了研究者们的广泛关注. 目前，文本信息隐藏方法可以分为两大类. 一类是基于修改的方法，主要依据语法或语义对文本的内容进行修改实现秘密信息的嵌入，如同义词替换[14]、语法转换[15]等. 然而，由于文本信息的冗余度较小，这类方法的嵌入容量较低. 而且修改会或多或少地带来文本分布的变化，很容易被检测出来；另一类是基于生成的方法，通过统计语言模型，在秘密信息的驱动下自动生成与正常文本分布相近的隐写文本. 早期的研究使用 Markov 模型[16]和 N-gram 模型[17]引入语义特征；随着深度学习的发展，基于神经语言模型 Recurrent Neural Networks (RNN)[18]、Generative Adversarial Networks (GAN)[19]等的隐写方法通过对条件概率进行编码，根据秘密比特流从候选池中选择单词，生成更自然的文本[20-22]. 基于生成的文本信息隐藏方法具有较强的安全性和不可见性，在文本隐写任务中取得了最先进的性能[23].

在信息隐藏技术为人类安全做出贡献的同时，文本信息隐藏也可能被犯罪分子利用，对网络安全构成潜在威胁. 文本隐写分析作为文本隐藏的对抗技术，其目的是检测文本中是否存在秘密信息，有效防止文本隐写术的滥用. 传统的文本隐写分析方法大多是基于通用的机器学习框架提出[24-26]，首先手动抽取特征，再用分类器分出正常文本和载密文本. 然而，这些传统的方法都是针对特定隐写算法来设计特征的，难以适应不同类型的隐写算法. 为了应对逐渐成熟的文本隐写术带来的挑战，近年来，研究者们开始研究基于深度学习的文本隐写分析算法. Wen 等人[27]使用卷积神经网络(Convolutional Neural Network，CNN)来捕捉单词之间的局部相关性；Yang 等人[28]利用循环神经网络(Recurrent Neural Network，RNN)提取长距离语义特征；Niu 等人[29]将 CNN 和 Bi-LSTM 结合起来，获得局部和全局语义特征；Yang 等人[30]应用了一种基于特征金字塔的紧密连接的 LSTM 来融合额外的低层特征，以识别基于生成的文本隐写术；Peng 等人[31]通过一个微调的 BERT 模型建模了正常文本和隐写文本之间的差异；Yang 等人[23]使用转换器架构的语言模型作为语义提取器，利用图神经网络来保留句法特征；

以上这些方法以端到端的方式自动提取特征，克服了传统手动特征提取方法的局限性，但是大多只在检测与训练样本独立同分布的的隐藏文本时能获得较好的性能. 当待检测文本的领域、所对应的隐藏算法和嵌入容量发生变化时，模型的隐写分析性能会有一定程度的下降. 为了解决以上问题，Xue 等人[32]设计了一个分布式的适应层，采用三个损失函数来实现领域自适应，使模型在语料失配的情况下具有更好的隐写分析性能；Wen 等人[33]提出了一个用于文本隐写分析的元学习框架——FS-Stega，实现了模型在文本域、隐藏算法和嵌入容量不同时的快速适应.

在本文中，我们将不同类别的隐写文本分析工作看作不同的任务. 待检测的文本类别主要取决于三个方面，即文本语料、隐藏算法和嵌入容量. 在实际应用中，同一个隐写分析模型通常会遇到多个不同的检测任务. 不同任务的训练样本具有较大的分布差异，且有的任务所能使用的训练样本可能非常稀疏. 因此，为了增强文本隐写分析模型在不同检测任务上的快速自适应能力，并使模型能处理少样本场景下的隐写分析任务，本文提出了一种基于胶囊网络的文本隐写分析方法——CN-Stega(Capsule Net-

work Steganalysis). 该模型利用活动向量(Activity Vector)特征状态的重要信息,稳健地学习部分与整体关系中的不变量,能更好的建模高层特征和低层特征之间的相对关系,并学习到单个语句和其所在任务之间的特定信息,继而区分出各语句所属的任务,达到有效区分正常文本和载密文本的目的.

CN-Stega 包含三个部分:通用特征提取器、任务映射器和分类器. 其中,任务映射器被视为元学习者,通过不同的支持集获取句子的元表示,主导元训练过程. 具体来说,为了解决模型跨领域检测性能低的问题,我们使用带有自注意力的 Bi-LSTM 作为通用任务提取器,从支持集中获取有标注文本的句子表示,并从查询集中获取待检测文本的句子表示. 接下来,将支持集的句子表示输入到任务映射器中,进行单一文本到整体任务的非线性映射. 然后,将映射结果和查询集的句子表示输入分类器,度量待检测文本与任务之间的匹配程度,并将抽象的匹配程度量化为适配分数. 最后,得到基于适配分数的均方误差损失和基于真实标签分布和预测概率分布的 KL 散度损失组成的总预测损失. 大量实验证明,CN-Stega 可以快速适应各种不同的任务,并在少样本学习的场景中达到先进的检测性能. 这证实了 CN-Stega 具有检测来自不同文本语料、隐写算法和嵌入容量的先进能力. 综上所述,本文的贡献如下:

(1) 我们提出了一种基于胶囊网络的多任务少样本文本隐写分析模型,能实现不同任务之间的模型快速自适应. 其中,任务映射器被视为元学习者,主导元训练过程,学习单一文本与任务间的非线性映射关系.

(2) 为了充分融合不同任务的预测结果,本文从两方面考察损失:一方面考察待检测文本与任务之间的 MSE 损失;另一方面考察真实标签分布和预测概率分布之间的 KL 散度损失. 该损失用于预测模型的多分类任务.

(3) 我们构建了三个元数据集开展隐写分析实验. 实验结果表明,与最先进的算法相比,CN-Stega 在训练样本较少的场景下对不同任务具有更高的检测准确率.

本文第 2 节介绍胶囊网络和元学习的相关工作;第 3 节详细说明 CN-Stega 的算法流程和评估指标;第 4 节给出实验设置和实验评价结果,并进行全面的讨论;最后,在第 5 节得出结论.

2 相关工作

2.1 胶囊网络

胶囊网络(Capsule Network)于 2017 年由 Sabour 等人[34]提出. 不同于 CNN 用标量记录局部信息[35],胶囊网络使用活动向量特征学习部分与整体关系中的不变量. 胶囊网络对输入向量的处理分为两个阶段:仿射变换(Affine Transformation)和动态路由(Dynamic Routing). 其结构图如图 1 所示.

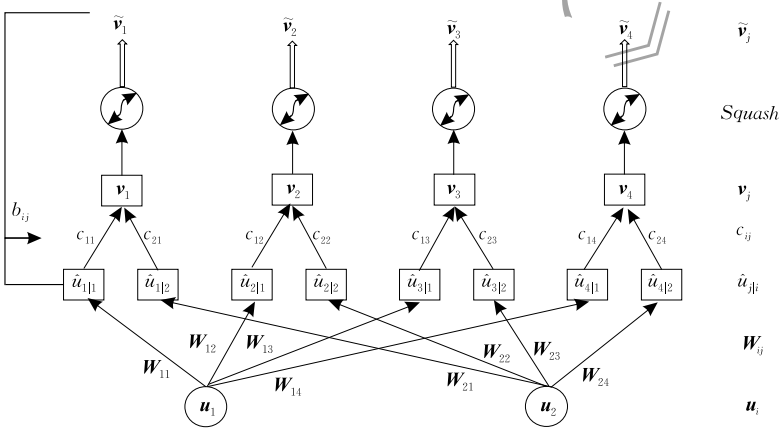

图 1 胶囊网络结构图

其中,$\boldsymbol{u}_i$ 是第 l 层的输出向量,由 l 层的胶囊 i 产生,$\boldsymbol{W}_{ij}$ 是权值矩阵,$\hat{u}_{j|i}$ 是仿射变换后的结果,用于输入到第 $l+1$ 层的胶囊 j 中:

$$\hat{u}_{j|i}=\boldsymbol{W}_{ij}\boldsymbol{u}_i \tag{1}$$

动态路由中,耦合系数(Coupling Coefficients) c_{ij} 指第 l 层的胶囊 i 和第 $l+1$ 层的胶囊 j 耦合的先验概率. 因此,胶囊 i 和下一层中所有胶囊的耦合系数和为 1. 每次迭代将动态修正胶囊间的连接强度并通过路由 softmax 选择确定耦合系数:

$$c_{ij}=\frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})} \tag{2}$$

其中，c_{ij} 满足条件：$\sum_j c_{ij}=1$. b_{ij} 是原始的权重，初始化为零，用来更新 c_{ij}，可以同时和其它权重有区别地进行学习.

活动向量 $\boldsymbol{v}_j$ 表示特定实体类型的实例化参数，如对象或部分对象.

$$\boldsymbol{v}_j=\sum_i c_{ij}\hat{\boldsymbol{u}}_{j|i} \tag{3}$$

活动向量的长度表示实体存在的概率，方向表示实例化参数，所以向量长度值必须在 0 到 1 之间. 为了实现这种压缩，并完成胶囊网络层级的激活功能，Sabour 等人使用了非线性函数 *Squash*. 该函数将向量 $\boldsymbol{v}_j$ 的长度缩放到 0 和 1 之间的同时，保持其方向不变：

$$\bar{\boldsymbol{v}}_j=\frac{\|\boldsymbol{v}_j\|^2}{1+\|\boldsymbol{v}_j\|^2}\frac{\boldsymbol{v}_j}{\|\boldsymbol{v}_j\|} \tag{4}$$

同一水平的活跃胶囊通过变换矩阵对更高级别的胶囊的实例化参数进行预测. 当多个预测一致时，更高级别的胶囊变得活跃.

近年来，胶囊网络在自然语言处理领域得到了广泛的应用. Zhao 等人[36]成功地将其应用于带有大标签数据集的完全监督文本分类问题；Xia 等人[37]扩展了基于胶囊的体系结构来计算目标意图和源意图之间的相似度；Geng 等人[38]在胶囊网络中动态路由的基础上创建了 Induction 网络，进行少样本文本分类的学习；Li 等人[39]运用胶囊网络进行文本隐写分析.

2.2　元学习

元学习，又被称为“学习如何学习”的方法，通过少数几个训练实例，快速学习新技能或适应新任务[40]. 每个任务可以表示为一个数据集 $\hat{D}$，数据集 $\hat{D}$ 通常被划分为两部分：一个是用于学习的支持集(Support Set) S；另一个是用于训练和测试的查询集(Query Set) Q，即 $\hat{D}=\langle S,Q\rangle$. N-way K-shot $\hat{Q}$-query 分类任务指支持集 S 中有 N 类数据，每类数据有 K 个带有标签的样本；查询集 Q 中有与支持集 S 相同的 N 个类，每类数据有 $\hat{Q}$ 个已标注的样本.

一般来说，元学习分为两大类. 其一，基于优化的方法. 在基于优化的元学习方法中，深度学习模型通过反向传播梯度进行学习. Munkhdalai 等人[41]提出了元网络(Meta Networks)；Mishra 等人[42]提出了简单神经专注学习者(Simple Neural AttenIve Learner，SNAIL)来提取任务间通用特征；Gu 等人[43]扩展了低资源神经机器翻译(Neural Machine Translation，NMT)的模型无关元学习算法(Model-Agnostic Meta-Learning，MAML)；Geng 等人[38]在动态路由的基础上提出了一种新的元学习网络：Induction 网络；Bao 等人[44]利用注意生成器来提高文本分类的准确性.

其二，基于度量的方法. 基于度量的元学习方法的核心思想类似于最近邻算法和核密度估计. 该方法在已知标签的支持集上预测出两个数据样本之间的相似概率. 因此，学到一个好的核函数对于基于度量的元学习模型至关重要. Koch 等人[45]使用暹罗神经网络对输入实例之间的相似性进行自然排序；Luong[46]通过多任务训练提高了模型效果；Snell 等人[47]提出了原型网络(Prototypical Networks)，通过计算欧氏距离衡量每一类的度量空间；Sung 等人[48]通过输入特征学习度量空间；为了提取一些可转移的知识，Ravi 和 Larochelle[49]利用两个元学习者学习优化算法；Nichol 等人[50]在每个任务的最优方向上依次下降，以逼近每个任务的最优位置；Yu 等人[51]将不同领域的任务聚类，每个聚类训练一个模型，然后在未知领域的少样本任务下对模型进行微调；Jiang 等人[52]训练了一个无偏初始模型来泛化其他各种任务.

在文本隐写分析领域，由于待检测的任务中，文本分布受到文本语料、隐藏算法和嵌入容量等诸多因素的影响而差异性较大，因此，现有模型的领域适应性不足. 为了处理不同的隐写文本检测任务，本文引入了元学习机制，基于胶囊网络完成样本到任务的映射，以实现模型在多任务少样本场景中的快速适应[38].

3　模　型

在本节中，我们将介绍所提出的 CN-Stega 文本隐写分析模型.

3.1　相关定义

文本隐写分析的目标是区分给定的待检测文本句子 $X=(x_1,x_2,x_3,\cdots,x_L)$ 是载体文本(cover text)还是载密文本(stego text)，其中，L 是句子的长度，x_i 表示句子中第 i 个单词. 将某个文本隐写分析任务 T_i 所检测的文本集 X 定义为：由某个特定的隐写算法 $f(\cdot)$ 在某个文本领域 D 上基于特定的嵌入容量 bpw(bits per bit)所生成的文本. 则待检测文本 X 由以下函数生成：

$$X=f(D,bpw) \tag{5}$$

其中嵌入容量 bpw 是指一比特载体中所嵌入的秘密

信息比特数.$bpw=0$ 表示未嵌入秘密信息的载体文本,而 $bpw>0$ 则表示载密文本.本模型使用多个隐写算法在不同文本领域的分布下基于不同的嵌入容量构建出多个任务,每个任务 T_i 就是一个特定的类(class),拥有特定的标签 y_i.任务集 T 被划分成三个不相交的子集:元训练集 M^{train},元验证集 M^{val} 以及元测试集 M^{test}.模型在任务集 T 上进行 N-way K-shot $\hat{Q}$-query 训练时,从 M^{train} 中随机选择 N 个任务,从每个任务中随机抽取 K 和 $\hat{Q}$ 个样本句子组成支持集 S 以及查询集 Q:

$$S=\{(X_{nk}^S,y_n^S)\mid k=1,2,3,\cdots,K;n=1,2,3,\cdots,N\} \tag{6}$$

$$Q=\{(X_{nk}^Q,y_n^Q)\mid k=K+1,K+2,K+3,\cdots,K+\hat{Q};\ n=1,2,3,\cdots,N\} \tag{7}$$

其中,X_{nk}^S 表示支持集 S 中第 n 个任务的第 k 个句子,y_n^S 表示支持集中句子对应的整数标签.

3.2 模型架构

本文提出的 CN-Stega 文本隐写分析模型的整体框架(以 3-way 1-shot 1-query 为例)如图 2 所示.该框架包含三部分:通用特征提取器、任务映射器和分类器.

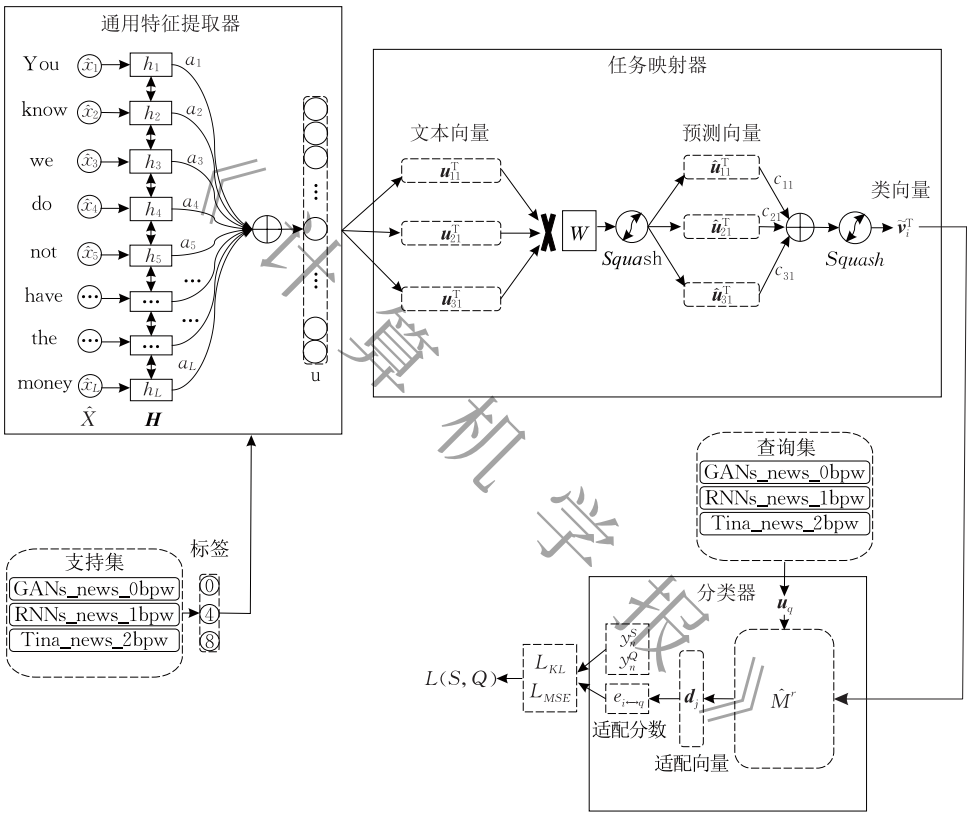

图 2 CN-Stega 文本隐写分析模型整体结构图(注:支持集和查询集中,Tina_news_2bpw 表示类中的文本是由隐写算法 Tina 基于 News 语料库以 2bpw 的嵌入容量生成的.为了更美观的图示,任务映射器中的向量均以转置的形式画出)

3.2.1 通用特征提取器

为了从不同的任务中提取通用任务特征,本文使用具有自注意的 Bi-LSTM[53] 来获取不同分布文本的通用信息.给定一条长度为 L 的输入文本 X,用词向量模型获取其词嵌入表示 $\hat{X}=(\hat{x}_1,\hat{x}_2,\hat{x}_3,\cdots,\hat{x}_L)$,并将其输入到 Bi-LSTM 模型中获取双向隐层特征:

$$\overrightarrow{h_t}=\overrightarrow{LSTM}(\hat{x}_t,h_{t-1}) \tag{8}$$

$$\overleftarrow{h_t}=\overleftarrow{LSTM}(\hat{x}_t,h_{t+1}) \tag{9}$$

我们连接 $\overrightarrow{h_t}$ 和 $\overleftarrow{h_t}$,形成隐层状态 h_t:

$$h_t=[\overrightarrow{h_t},\overleftarrow{h_t}] \tag{10}$$

因此,Bi-LSTM 最终的输出为 $\boldsymbol{H}=(h_1,h_2,h_3,\cdots,h_L)\in R^{L\times 2m}$,其中单向 LSTM 的隐层长度是 m.为了将可变长度的文本 X 编码成固定大小的嵌入,我们在 $\boldsymbol{H}$ 中选择一个隐层向量的线性组合,并计算注意力向量:

$$\boldsymbol{a}=\text{softmax}(\overline{\boldsymbol{W}}_{a2}\tanh(\overline{\boldsymbol{W}}_{a1}\boldsymbol{H}^{\mathrm{T}})) \tag{11}$$

其中,$\boldsymbol{a}\in R^{1\times L}$ 是注意力向量,$\overline{\boldsymbol{W}}_{a1}\in R^{d_a\times 2m}$ 和 $\overline{\boldsymbol{W}}_{a2}\in R^{1\times d_a}$ 是权值矩阵,d_a 是超参数.最终,句子的通用特征表示为 $\boldsymbol{u}$:

$$\boldsymbol{u}=\sum_{i=1}^{L} a_i \cdot h_i \tag{12}$$

3.2.2　任务映射器

在本节，我们采用了胶囊网络中的动态路由算法[34,38]. 任务映射器的主要目的是实现一个从文本向量 $\boldsymbol{u}_{ij}^{S}$ 到任务向量 $\tilde{\boldsymbol{v}}_i$ 的非线性映射：

$$\{\boldsymbol{u}_{ij}^{S}\}_{i=1,\cdots,N;j=1,\cdots,K} \rightarrow \{\tilde{\boldsymbol{v}}_i\}_{i=1,\cdots,N} \tag{13}$$

其中，$\boldsymbol{u}_{ij}^{S} \in R^{2m\times 1}$，表示支持集 S 中第 i 类的第 j 个句子经由通用特征提取器得到的文本向量.

为了让模型能够处理任意规模(*any*-way *any*-shot)的任务，我们对支持集 S 上的所有文本向量 $\boldsymbol{u}_{ij}^{S}$ 进行了权重共享转换. 所有文本向量 $\boldsymbol{u}_{ij}^{S}$ 共享相同的变换权重 $\boldsymbol{W}\in R^{2m\times 2m}$ 和偏移量 B. 每条文本的预测向量 $\hat{\boldsymbol{u}}_{ij}^{S}$ 为

$$\hat{\boldsymbol{u}}_{ij}^{S}=squash(\boldsymbol{W}\cdot\boldsymbol{u}_{ij}^{S}+B) \tag{14}$$

从而，本模型可以处理任何规模的文本任务，具有较高的灵活性. 非线性映射捕获了低级样本特征和高级类别特征之间的重要的不变的语义关系[54]. 为了确保类向量 $\tilde{\boldsymbol{v}}_i$ 自动封装该类中的文本特征向量 $\boldsymbol{u}_{ij}^{S}$，本模型迭代使用动态路由. 在每次迭代中，动态选择并确定耦合系数 c_{ij}：

$$c_{ij}=\mathrm{softmax}(b_{ij}) \tag{15}$$

其中，$\sum_{j=1}^{K} c_{ij}=1$. b_{ij} 是权重，我们在首次迭代中将其初始化为 0.

给定每条文本的预测向量 $\hat{\boldsymbol{u}}_{ij}^{S}$，则每个任务的候选向量 $\boldsymbol{v}_i$ 是第 i 个任务中所有文本的预测向量 $\hat{\boldsymbol{u}}_{ij}^{S}$ 的加权和：

$$\boldsymbol{v}_i=\sum_{j=1}^{K} c_{ij}\cdot\hat{\boldsymbol{u}}_{ij}^{S} \tag{16}$$

然后，使用 *squash* 函数得到路由过程中的输出向量——任务向量 $\tilde{\boldsymbol{v}}_i$：

$$\tilde{\boldsymbol{v}}_i=squash(\boldsymbol{v}_i) \tag{17}$$

在每次迭代的最后使用协议路由法调整 b_{ij}：

$$b_{ij}=b_{ij}+\hat{\boldsymbol{u}}_{ij}^{S}\cdot\tilde{\boldsymbol{v}}_i \tag{18}$$

如果某一个文本的预测向量 $\hat{\boldsymbol{u}}_{ij}^{S}$ 生成的任务候选向量 $\boldsymbol{v}_i$ 的长度较大，将在动态路由迭代中产生自顶向下的反馈. 该反馈在增加该样本耦合系数的同时降低其他样本的耦合系数，这种类型的调整不需要恢复任何参数. 因此，本文的少样本学习网络具有较高的有效性和鲁棒性.

3.2.3　分类器

在分类器模块，我们将度量待检测文本与任务类之间的匹配程度，并将匹配程度量化为具体的数值——匹配分数 $e_{i\leftrightarrow q}$. 分类器有两个输入，一个是文本向量 $\boldsymbol{u}_{ij}^{S}$ 经由任务映射器最终映射成的任务向量 $\tilde{\boldsymbol{v}}_i$，另一个是查询集 Q 中的每个待检测文本 X 经由通用特征提取器后获得的文本向量 $\{\boldsymbol{u}_q\}_{q=1}^{\hat{Q}}$. 分类器的输出即为适配分数 $e_{i\leftrightarrow q}\in[0,1]$，是一个标量，表示支持集中标注文本所在的第 i 类与查询集中第 q 个待检测文本之间的匹配程度. 具体来说，我们引入了神经张量层[55]，它擅长对两个向量之间的匹配程度进行建模[56-57]. 神经张量层输出如下的适配向量 $\boldsymbol{d}_i$：

$$d(\tilde{\boldsymbol{v}}_i,\boldsymbol{u}_q)=f(\tilde{\boldsymbol{v}}_i^{\mathrm{T}}\cdot\hat{M}^{[1:z]}\cdot\boldsymbol{u}_q) \tag{19}$$

其中，$\hat{M}^{r}\in R^{2m\times 2m\times z}$ 是张量，$r\in[1,\cdots,z]$ 是张量的 z 个切片，f 是非线性激活函数 RELU[58]. 支持集中的第 i 类和查询集中第 q 个待检测文本之间的最终适配分数 $e_{i\leftrightarrow q}$ 由 sigmoid 激活函数计算出：

$$e_{i\leftrightarrow q}=\mathrm{sigmoid}(\hat{W}\cdot d(\tilde{\boldsymbol{v}}_i,\boldsymbol{u}_q)+\hat{B}) \tag{20}$$

其中，$\hat{W}$ 是权重，$\hat{B}$ 是偏移量.

3.2.4　目标函数

我们使用 *MSE* 损失和 *KL* 散度损失来共同训练 CN-Stega 模型：

$$L(S,Q)=L_{MSE}+\gamma L_{KL} \tag{21}$$

其中，γ 是超参数，本文取值 0.5；L_{MSE} 是多分类任务中适配分数 $e_{i\leftrightarrow q}$ 与查询集的真实标签 y_n^Q 之间的均方误差损失. 当 $e_{i\leftrightarrow q}$ 与 y_n^Q 成功匹配时相似度为 1，不匹配时相似度为 0：

$$L_{MSE}=\sum_{i=1}^{N}\sum_{n=1}^{\hat{Q}}(e_{i\leftrightarrow q}-1(y_n^Q==i))^2 \tag{22}$$

KL 散度主要用来衡量两个概率分布之间的相似性，两个概率分布越相似，*KL* 散度值越小. 本文中，我们用 *KL* 散度衡量多分类任务中真实标签分布和预测概率分布之间的分布差异：

$$L_{KL}=\sum_{i=1}^{N\times\hat{Q}}[t_i\log t_i-t_i\log p_i] \tag{23}$$

其中，t_i 为真实的标签分布，p_i 为预测的概率分布. 当两分布的相似度不足时，训练产生的 *KL* 值过大，经由模型训练使得真实标签分布和预测概率分布逐渐接近，*KL* 散度值也随之减小，模型得到有效训练.

本模型三个部分的所有参数均由自适应梯度(Adaptive Gradient，Adagrad)[59]和反向传播联合训练，有着良好的泛化特性. 随着训练次数的增加，模型逐渐积累归纳和比较的能力. 模型的元训练的细节见算法 1.

算法 1. CN-Stega 元训练算法.

输入:元训练集 M^{train}、*Learning rate* ζ

输出:适配分数 $e_{i\leftrightarrow q}$

1. 随机初始化
2. REPEAT
3. 从 M^{train} 中随机采样 N 个任务(包含嵌入容量为 0 的载体任务)
4. FOR 任务 T_i 的所有文本 DO
5. 从 T_i 中随机采样 $K+\hat{Q}$ 个样本组成支持集 S_i 和查询集 Q_i
6. 利用通用特征提取器获取 S_i 的文本向量 $\boldsymbol{u}_{ij}^{S}$、Q_i 的文本向量 $\boldsymbol{u}_q$
7. 根据式(14),计算 S_i 的预测向量 $\hat{\boldsymbol{u}}_{ij}^{S}$
8. 初始化 $b_{ij}=0$
9. FOR *iter* DO
10. 根据式(15)~(17),计算得到类向量 $\tilde{\boldsymbol{v}}_i$
11. 根据式(18),更新 b_{ij}
12. END FOR
13. 根据式(20),计算 S_i 中的第 i 类与 Q_i 中第 q 个待检测文本之间的适配分数 $e_{i\leftrightarrow q}$
14. 根据式(21)~(23),计算总损失 $L(S,Q)$
15. 根据总损失,通过 Adagrad 更新 $e_{i\leftrightarrow q}$
16. END FOR
17. UNTIL *episode*=3000 or *patience*=500

4 实 验

4.1 数据集与实验设置

本文使用的多任务数据集由我们自己创建.为了使模型的泛化性能更强,在创建数据集时采用了不同的文本隐写算法、嵌入容量以及语料库.不同的隐写算法带来的分布差异主要体现在两方面:隐写文本生成器以及嵌入方式.从生成器的角度考虑,我们选择了目前最好的生成式模型,即基于 GAN、RNN 和 LSTM 的生成式隐写算法.从嵌入方式的角度考虑,我们选择的这三种算法中有基于采样(sampling-based)的方法和基于块分组(block-based)的方法.基于上述考虑,我们选择了这三种代表性隐写算法:GAN-TStega (GANs)[19]、RNN-Stega (RNNs)[18] 和 Tina[21],用于生成嵌入容量分别为 0 和 1、2、3 的载体文本以及载密文本.我们在四个文本隐写分析常用的语料库上评估提出的模型:Twitter[60]、COCO[61]、News① 和 Movie Review(Movie)[62].

因此,我们基于四个语料库,使用三种隐写算法生成的四种文本构建了元训练集 M^{train} 和元测试集 M^{test}.为了测试模型在不同领域的泛化能力,我们设定,元训练集 M^{train} 中的任务来自同一个文本语料(例如,Movie),而元测试集 M^{test} 中的任务来自另外三个语料(例如,Twitter、COCO 和 News).最终,我们构建了三个不同的元数据集,每个元数据集用于不同的独立实验(详见表 1~表 3),以考察模型在新的文本语料上的自适应隐写分析能力.同时,我们规定嵌入容量为零($bpw=0$)的任务为载体任务并分配任务标签 0,嵌入容量非零($bpw>0$)的任务为载密任务,任务标签是 1~12.从表 1~表 3 可以看出,

表 1 元数据集 1——以 Movie 为元训练域

	Movie				
	任务标签	隐写算法	训练样本数量	文本领域	隐藏容量 bpw
元训练集	0	GANs	500	Movie	0
		RNNs	500		
		Tina	500		
	1	GANs	3500	Movie	1
	2	GANs	3000	Movie	2
	3	GANs	3500	Movie	3
	4	RNNs	3500	Movie	1
	5	RNNs	3500	Movie	2
	6	RNNs	3500	Movie	3
	7	Tina	3500	Movie	1
	8	Tina	3500	Movie	2
	9	Tina	3500	Movie	3
元测试集	10	Tina	50	COCO	1
	11	GANs	50	Twitter	2
	12	RNNs	50	News	3

注:训练样本数量指不同类别下文本句的数量,如 1-GANs-3500-Movie-1 表示类 1 是由隐写算法 GANs 基于 Movie 语料库以 1bpw 的嵌入容量生成的 3500 条语句构成的.

表 2 元数据集 2——以 Twitter 为元训练域

	Twitter				
	任务标签	隐写算法	训练样本数量	文本领域	隐藏容量 bpw
元训练集	0	GANs	500	Twitter	0
		RNNs	500		
		Tina	500		
	1	GANs	3500	Twitter	1
	2	GANs	3000	Twitter	2
	3	GANs	3500	Twitter	3
	4	RNNs	3500	Twitter	1
	5	RNNs	3500	Twitter	2
	6	RNNs	3500	Twitter	3
	7	Tina	3500	Twitter	1
	8	Tina	3500	Twitter	2
	9	Tina	3500	Twitter	3
元测试集	10	Tina	50	COCO	1
	11	GANs	50	Movie	2
	12	RNNs	50	News	3

① Thompson A. Snapcrack/all-the-news/data. https://www.kaggle.com/

表 3　元数据集 3——以 News 为元训练域

	News				
	任务标签	隐写算法	训练样本数量	文本领域	隐藏容量 bpw
元训练集	0	GANs	500	News	0
		RNNs	500		
		Tina	500		
	1	GANs	3500	News	1
	2	GANs	3000	News	2
	3	GANs	3500	News	3
	4	RNNs	3500	News	1
	5	RNNs	3500	News	2
	6	RNNs	3500	News	3
	7	Tina	3500	News	1
	8	Tina	3500	News	2
	9	Tina	3500	News	3
元测试集	10	Tina	50	COCO	1
	11	GANs	50	Movie	2
	12	RNNs	50	Twitter	3

我们在构建载体任务时混合了三种隐写算法生成的语料，而载密类别的任务只有单一的语料. 这是考虑到隐写算法造成文本的分布差异的两个方面. 第一是不同算法所用的语言模型不同，带来了生成文本的分布差异；第二是秘密信息的嵌入方式，如编码方式和候选池的构建方式不同带来的隐写文本的分布差异. 而后者导致的分布差异更为明显. 载体类中的语句均为嵌入容量为零的生成文本，即没有进行秘密信息的嵌入，因此，语料的分布仅受到前者语言模型的影响，没有受到后者嵌入方式的影响. 而载密类中的语句分别由不同的嵌入方式和不同的隐藏容量生成，分布特征差别较大. 因此，我们将混合了三种隐写算法生成的载体类看成同一个任务，而将载密类看成不同任务. 在现实场景中，我们遇到的载体类也通常是多而杂的，而待检测的载密类往往是相对较少的. 这样做也有助于弱化载体类别的分布差异，把关注点放在不同隐写算法导致的载密文本分布差异上.

我们在三个元数据集上使用了维度为 300 的 Glove 嵌入[63]，即 $emb\_dim=300$；设置单向 LSTM 的隐层状态大小 $m=128$，注意力维度 $d_a=64$. 任务映射器中，动态路由算法使用的迭代次数 $iter=3$. 适配器中是一个 $z=100$ 的神经张量层，后面接一个由 sigmoid 激活的全连接层. 我们在三个元数据集上建立了 3-way K-shot 25-query 的模型（$N=3$，$K=1$、5、10，$\hat{Q}=25$），即随机选出 3 个样本类，每一类有 K 个标注文本和 25 个待检测文本（标注文本和待检测文本不相同）. 模型的学习率 $learning\ rate=$ 1e-4，随机种子 $seed=42$，$episode=3000$. 实验设置及参数如表 4 所示.

表 4　实验参数表

名称	数量	名称	数量
emb_dim	300	$seed$	42
m	128	$learning\ rate$	1e-4
d_a	64	$episode$	3000
$iter$	3	$\hat{Q}$	25
z	100		

4.2　评估指标

我们使用的评估指标包括准确性（accuracy，acc）、精度（precision，pre）和召回率（$recall$），这三个指标在文本隐写分析中被广泛用于检测模型性能：

$$acc=\frac{TP+TN}{TP+FP+TN+FN} \tag{24}$$

$$pre=\frac{TP}{TP+FP} \tag{25}$$

$$recall=\frac{TP}{TP+TN} \tag{26}$$

其中，TP（True Positive）表示将正例正确预测为正例的数量，TN（True Negative）表示将负例正确预测为负例的数量，FN（False Negative）表示将正例错误预测为负例的数量，FP（False Positive）表示将负例错误预测为正例的数量.

模型的元测试过程利用训练好的模型，输出在元测试集上的评估结果. 细节见算法 2.

算法 2.　CN-Stega 元测试算法.

输入：元测试集 M^{test}

输出：评估指标 acc、pre、$recall$

1. 随机初始化
2. 从 M^{test} 中随机采样 N 个任务
3. FOR 任务 T_i 的所有文本 DO
4. 　从 T_i 中随机采样 $K+\hat{Q}$ 个样本组成支持集 S_i 和查询集 Q_i
5. 　计算得到 S_i 的文本向量 $\boldsymbol{u}_{ij}^{S}$、Q_i 的文本向量 $\boldsymbol{u}_q$
6. 　根据式(14)，计算 S_i 的预测向量 $\hat{\boldsymbol{u}}_{ij}^{S}$
7. 　初始化 $b_{ij}=0$
8. 　FOR $iter$ DO
9. 　　根据式(15)～(17)，计算得到类向量 $\tilde{\boldsymbol{v}}_i$
10. 　　根据式(18)，更新 b_{ij}
11. 　　END FOR
12. 　　根据式(20)，计算适配分数 $e_{i\leftrightarrow q}$
13. 　　根据式(24)～(26)，计算评估指标 acc、pre 和 $recall$
14. END FOR

4.3　与经典元学习方法的比较

为了验证本文提出的模型在元学习范围内的有效性，我们选择了三种元学习算法 PROTO、MAML 和 FS-Stega 进行文本隐写分析实验的比较.

PROTO在机器视觉领域取得了良好的分类效果[47],它通过最小化中心点和每个任务实例之间的欧几里德距离来学习度量空间,以此来训练模型.我们使用激活函数RELU,使用MLP作为隐藏层,*dropout*=0.1,隐藏层和输出层的大小都设置为300.

MAML通过使用先验模型的参数达到快速适应新任务的目的[64].在MAML的快速适应阶段,每次训练中每0.1步长更新10次;外循环的学习率设置为0.001,梯度更新参数取10个采样任务的平均值.我们同样使用MLP和RELU,设置同PROTO.

而FS-Stega提出了一个用于文本隐写分析的元学习框架,以确保模型在任务之间的快速适应.它使用BERT提取任务间的通用特征,使用基于注意力的Bi-LSTM的元学习器学习任务特定特征[33].本文采用其效果更好的岭回归分类器来训练模型、更新元学习器.

表5给出了PROTO、MAML、FS-Stega和本模型在三个元数据集下分别进行3-way *K*-shot(*K*=1,5,10)的实验结果.每个数据集包含的隐写算法、文本领域和隐藏容量详见表1～表3.从表5中可以看出,与PROTO和MAML相比,本模型的准确率在3-way 1-shot场景下提高了10.36%～35.04%,平均值至少提高了13.47%;在3-way 5-shot场景下提高了3.69%～21.98%,平均值提高了6.65%以上;在3-way 10-shot场景下提高了6.79%～18.43%,平均值至少提高了8.6%.在Movie和News训练域下,本模型比FS-Stega的准确率高出0.06%～16.11%.平均来看,本模型性能较好.

表5 不同元训练域下的元学习方法的检测性能(3-way)

	方法	Movie			News			Twitter			平均值		
		acc	*recall*	*pre*	*acc*	*recall*	*pre*	*acc*	*recall*	*pre*	*acc*	*recall*	*pre*
1 shot	MAML	0.6996	0.6047	0.6971	0.5295	0.4919	0.6007	0.5325	0.5065	0.5113	0.5872	0.5344	0.6030
	PROTO	0.7421	0.6479	0.7152	0.7306	0.7020	0.7108	0.6764	0.6854	0.5892	0.7164	0.6784	0.6717
	FS-Stega[33]	0.8361	0.7910	0.7902	0.7488	0.6703	0.7027	**0.8241**	0.8212	0.8202	0.7930	0.7608	0.7710
	Our(w/o *KL*)	0.7733	0.6900	0.7781	0.7200	0.7800	0.7566	0.6900	0.7700	0.7604	0.7278	0.7467	0.7650
	Our	**0.8933**	**0.8799**	**0.8709**	**0.8799**	**0.8999**	**0.8634**	0.7800	**0.8600**	**0.8551**	**0.8511**	**0.8799**	**0.8631**
5 shot	MAML	0.7564	0.7289	0.7292	0.6602	0.5161	0.6174	0.6491	0.5299	0.6240	0.6886	0.5916	0.6569
	PROTO	0.8149	0.7637	0.8172	0.8431	0.8131	0.8077	0.8013	0.8152	0.7385	0.8198	0.7973	0.7878
	FS-Stega[33]	0.8748	0.8636	0.8682	0.8712	0.8456	0.8663	**0.8923**	**0.8698**	**0.8760**	0.8794	0.8597	0.8702
	Our(w/o *KL*)	0.8266	0.8299	0.8055	0.7599	0.8200	0.7907	0.7666	0.7087	0.7524	0.7844	0.7862	0.7829
	Our	**0.9200**	**0.9399**	**0.9032**	**0.8800**	**0.9099**	**0.8676**	0.8588	0.8602	0.8472	**0.8863**	**0.9033**	**0.8727**
10 shot	MAML	0.7623	0.7322	0.7368	0.7451	0.7080	0.7061	0.7611	0.6247	0.7750	0.7562	0.6883	0.7393
	PROTO	0.8336	0.7881	0.8363	0.8320	0.8296	0.8267	0.8836	0.7876	0.8358	0.8331	0.8018	0.8329
	FS-Stega[33]	0.8860	0.8721	0.8700	0.8993	0.8764	**0.8967**	**0.9165**	0.8708	0.8717	0.9006	0.8731	0.8795
	Our(w/o *KL*)	0.8800	0.8600	0.8676	0.8000	0.8300	0.7938	0.8606	0.8610	0.8399	0.8469	0.8503	0.8338
	Our	**0.9466**	**0.9400**	**0.9336**	**0.8999**	**0.8800**	0.8634	0.9107	**0.9000**	**0.8977**	**0.9191**	**0.9067**	**0.8982**

注:Our指本文所提出的模型;Our(w/o *KL*)表示仅使用了*MSE*损失,没有使用*KL*损失.

实验证明,我们所提出的模型在不同领域任务间具有较好的泛化能力.另外,我们可以看到,就本模型而言,从两方面考察模型损失(Our)得到的模型性能要优于单方面考察的损失(Our(w/o *KL*)).这是因为在训练过程中,*KL*散度损失衡量真实标签分布和预测概率分布之间的相似性.在训练的初期,二者的相似性不高,随着训练的进行,两个概率分布之间的相似性逐步增大,损失函数逐渐减小,进而提高模型的检测性能.

4.4 与经典文本隐写分析方法的比较

我们将本文提出的CN-Stega模型与LS-CNN[27]、TS-RNN[28]、Dense-BiLSTM[30]和R-BiLSTM-C[29]四种最先进的文本隐写分析模型进行比较,以验证本模型的性能.特别需要说明的是,我们的模型是基于胶囊网络的在少样本上进行元训练的方法,而其他四种文本隐写分析模型无法在少数样本上训练.于是,我们将元训练数据集中的所有样本投入到其他四种模型中进行训练.

从表6可以看出,在三个训练域下,LS-CNN模型比我们的模型CN-Stega具有更高的精度,然而,LS-CNN模型的准确率和召回率远低于本模型.这说明LS-CNN模型在一个相对较小的训练数据集上并没有很好地利用所有的样本.此外,CN-Stega比其他隐写分析模型准确率提升了21.07%以上.另外,从两方面考察本模型损失(Our)得到的模型性能要优于单方面考察的损失(Our(w/o *KL*)).为了更直观的表明各模型对待检测句子的判断能力,我们考察了各模型对标签不同的两个例句的判断情况,如表7所示.以上结果表明本模型能够更好地从小样本数据集中提取特征,并很好地适应跨领域任务.

表 6　不同隐写分析模型的检测性能

方法	Movie			News			Twitter		
	acc	*recall*	*pre*	*acc*	*recall*	*pre*	*acc*	*recall*	*pre*
LS-CNN	0.5633	0.1267	**1.0000**	0.5933	0.2133	**0.8889**	0.6467	0.3000	**0.9783**
TS-BiRNN	0.6300	0.7200	0.6667	0.6500	0.4467	0.8701	0.7000	0.4667	0.8750
BiLSTM-Dence	0.6567	0.4333	0.7831	0.6467	0.3933	0.7972	0.6400	0.6467	0.6382
R-BiLSTM-C	0.6400	0.2933	0.9565	0.5933	0.2004	0.8181	0.6200	0.2870	0.8600
CN-Stega(w/o *KL*)	0.8800	0.8600	0.8676	0.8000	0.8300	0.7938	0.8606	0.8610	0.8399
CN-Stega	**0.9466**	**0.9400**	0.9336	**0.8999**	**0.8800**	0.8634	**0.9107**	**0.9000**	0.8977

注：CN-Stega 通过 3-way 10-shot 策略训练，其他模型使用元训练数据集中所有样本进行训练. CN-Stega(w/o *KL*)表示仅使用了 *MSE* 损失，没有使用 *KL* 损失.

表 7　不同隐写分析模型的例句判断情况

例句	标签	隐写分析方法	是否判断正确
but if you are taking medication there is absolutely no reason why you should not know whether it is on the banned list or not	0	LS-CNN	否
		TS-BiRNN	是
		BiLSTM-Dence	是
		R-BiLSTM-C	否
		CN-Stega	是
she will be a very good person and the president of the united states of america	1	LS-CNN	是
		TS-BiRNN	是
		BiLSTM-Dence	否
		R-BiLSTM-C	否
		CN-Stega	是

注：两个例句来自以 News 为元训练域的元数据集 3. CN-Stega 通过 3-way 10-shot 策略训练，其他模型使用元训练数据集 3 中所有样本进行训练.

4.5　特征提取消融实验

为了验证本文所使用的通用任务编码器的有效性，本节通过对任务通用性结构的消融来研究验证. 我们使用常用的 CNN[35] 方法来代替我们的通用任务提取器，二者均在 3-way 10-shot 25-query 策略下进行训练. CNN 对输入的文本单词进行一维卷积，通过 Max-over-Time Pooling 得到句子表示. 结果如表 8 所示.

从表 8 中可以看出，CNN 在本文框架下的表现不如 Bi-LSTM，这证明了 Bi-LSTM 作为通用任务特征提取器在提取元训练数据集特征方面更有优势.

表 8　不同特征提取方法的检测性能

提取方法	Movie			News			Twitter		
	acc	*recall*	*pre*	*acc*	*recall*	*pre*	*acc*	*recall*	*pre*
CNN	0.8266	0.7400	0.8968	0.7466	0.8100	0.7841	0.6800	0.7599	0.7551
Bi-LSTM	**0.9466**	**0.9400**	**0.9336**	**0.8999**	**0.8800**	**0.8634**	**0.9107**	**0.9000**	**0.8977**

注：Bi-LSTM 表示本模型所采用的通用任务提取器；两者均通过 3-way 10-shot 25-query 策略训练.

5　结　论

在本文，我们提出了基于胶囊网络的多任务少样本文本隐写分析模型 CN-Stega. 其中，任务映射器作为元学习者主导元训练过程，将通用特征提取器获取到的支持集的句子表示非线性的映射到任务. 然后，将查询集中的待检测文本输入本模型，度量其与任务之间的匹配程度，并将匹配程度量化为更直观的适配分数. 最后，模型得到基于适配分数的均方误差损失以及基于真实标签分布和预测概率分布差异的 *KL* 散度损失构成的总预测损失. 实验证明，与目前最先进的隐写分析模型和其他元学习方法相比，我们的模型具有更好的领域自适应能力，并显著提高了少样本条件下的文本检测性能. 在未来工作中，我们将基于更多的隐写算法来研究多任务隐写分析模型.

参　考　文　献

[1] Shannon C E. Communication theory of secrecy systems. Bell System Technical Journal, 1949, 28(4): 656-715

[2] Petitcolas F A, Anderson R J, Kuhn M G. Information hiding—A survey. Proceedings of the IEEE, 1999, 87(7): 1062-1078

[3] Bernaille L, Teixeira R. Early recognition of encrypted applications. Lecture Notes in Computer Science, 2007, 44(27): 165-175

[4] Guan Meng-Meng, Cao Yun, Zhang Yi-Xuan, et al. A transcoding-resistant video steganographic algorithm based on adaptive singular value modification. Journal of Cyber Security, 2018, 3(6): 42-54(in Chinese)
(管萌萌，曹纭，张怡暄等. 基于自适应奇异值调制的抗转码视频隐写算法. 信息安全学报，2018，3(6)：42-54)

[5] Marvel L M, Boncelet C G, Retter C T. Spread spectrum image steganography. IEEE Transactions on Image Processing, 1999, 8(8): 1075-1083

[6] Luo Wei-Qi, Huang Fang-Jun, Huang Ji-Wu. Edge adaptive

image steganography based on LSB matching revisited. IEEE Transactions on Information Forensics and Security, 2010, 5(2): 201-214

[7] Liao Xin, Tang Zhi-Qiang, Cao Yun. Steganographic distortion function learning method for spatial color image based on generative adversarial network. Journal of Software, 2022, 33(9): 3470-3484(in Chinese)
(廖鑫，唐志强，曹纭. 基于生成对抗网络的空域彩色图像隐写失真函数设计方法. 软件学报，2022，33(9)：3470-3484)

[8] Ding Xu-Yang, Xie Ying, Zhang Xiao-Song. Evolutionary multi-objective optimization image steganography based on edge computing. Journal of Computer Research and Development, 2020, 57(11): 2260-2270(in Chinese)
(丁旭阳，谢盈，张小松. 基于边缘计算的进化多目标优化图像隐写算法. 计算机研究与发展，2020，57(11)：2260-2270)

[9] Yang Zhong-Liang, Peng Xue-Shun, Huang Yong-Feng. A sudoku matrix-based method of pitch period steganography in low-rate speech coding. Security and Privacy in Communication Networks, 2018, 238(10): 752-762

[10] Cai Sen, Ren Yan-Zhen, Wang Li-Na. AAC steganographic algorithm based on joint distortion. Journal of Cyber Security, 2022, 7(2): 1-15 (in Chinese)
(蔡森，任延珍，王丽娜. 基于联合失真的 AAC 安全隐写算法. 信息安全学报，2022，7(2)：1-15)

[11] Yue Feng, Zhu Hui, Su Zhao-Pin, et al. An adaptive audio steganography using BN optimizing SNGAN. Chinese Journal of Computers, 2022, 45(2): 427-440(in Chinese)
(岳峰，朱慧，苏兆品等. 基于 BN 优化 SNGAN 的自适应音频隐写. 计算机学报，2022，45(2)：427-440)

[12] Sharma S, Gupta A, Trivedi M C, et al. Analysis of different text steganography techniques: A survey. Computational Intelligence and Communication Technology, 2016, 20(18): 130-133

[13] Majeed M A, Sulaiman R, Shukur Z, et al. A review on text steganography techniques. Mathematics, 2021, 9(21): 1-28

[14] Lin Huo, Yu Chuan-Xiao. Synonym substitution-based steganographic algorithm with vector distance of two-gram dependency collocations//Proceedings of the IEEE International Conference on Computer and Communications. Chengdu, China, 2016: 2776-2780

[15] Kim M Y, Zaiane O R, Goebel R. Natural language watermarking based on syntactic displacement and morphological division//Proceedings of the Annual Computer Software and Applications Conference Workshops. Seoul, Korea, 2010: 164-169

[16] Yang Zhong-Liang, Jin Shu-Yu, Huang Yong-Feng. Automatically generate steganographic text based on Markov model and Huffman coding. Cryptography and Security, 2018, 11(3): 1-10

[17] Luo Yu-Bo, Huang Yong-Feng, Li Fu-Fang. Text steganography based on Ci-poetry generation using Markov chain model. KSII Transactions on Internet and Information Systems, 2016, 10(9): 4568-4584

[18] Yang Zhong-Liang, Guo Xiao-Qing, Chen Zi-Ming. RNN-Stega: Linguistic steganography based on recurrent neural networks. IEEE Transactions on Information Forensics and Security, 2019, 14(5): 1280-1295

[19] Yang Zhong-Liang, Wei Nan, Liu Qing-He, et al. GAN-TStega: Text steganography based on generative adversarial networks. International Workshop on Digital Watermarking, 2020: 18-31

[20] Wen Juan, Zhou Xue-Jing, Li Meng-Di, et al. A novel natural language steganographic framework based on image description neural network. Journal of Visual Communication and Image Representation, 2019, 61(10): 157-169

[21] Fang T, Jaggi M, Argyraki K. Generating steganographic text with LSTMs//Proceedings of the Association for Computational Linguistics. Vancouver, Canada, 2017: 100-106

[22] Yang Zhong-Liang, Zhang Si-Yu, Hu Yu-Ting. VAE-Stega: Linguistic steganography based on variational auto-encoder. IEEE Transactions on Information Forensics and Security, 2021, 16: 880-895

[23] Yang Jin-Shuai, Yang Zhong-Liang, Zhang Si-Yu. SeSy: Linguistic steganalysis framework integrating semantic and syntactic features. IEEE Signal Processing Letters, 2022, 29: 31-35

[24] Din R, Yusof S M, Amphawan A. Performance analysis on text steganalysis method using a computational intelligence approach. Electrical Engineering Computer Science and Informatics, 2015, 2(3): 67-73

[25] Xiang Ling-Yun, Sun Xing-Ming, Luo Gang, et al. Linguistic steganalysis using the features derived from synonym frequency. Multimedia Tools and Applications, 2014, 71(3): 1893-1911

[26] Xiang Ling-Yun, Yu Jing-Min, Yang Chun-Fang, et al. A word-embedding-based steganalysis method for linguistic steganography via synonym substitution. IEEE Access, 2018, 28(6): 64131-64141

[27] Wen Juan, Zhou Xue-Jing, Zhong Ping, et al. Convolutional neural network based text steganalysis. IEEE Signal Processing Letters, 2019, 26(3): 460-464

[28] Yang Zhong-Liang, Wang Ke, Li Jian, et al. TS-RNN: Text steganalysis based on recurrent neural networks. IEEE Signal Processing Letters, 2019, 26(12): 1743-1747

[29] Niu Yan, Wen Juan, Zhong Ping, et al. A hybrid R-BILSTM-C neural network based text steganalysis. IEEE Signal Processing Letters, 2019, 26(12): 1907-1911

[30] Yang Hao, Bao Yong-Jian, Yang Zhong-Liang, et al. Linguistic steganalysis via densely connected LSTM with feature pyramid. Association for Computing Machinery, 2020, 20(3): 5-10

[31] Peng Wan-Li, Zhang Jin-Yu, Xue Yi-Ming, et al. Real-time text steganalysis based on multi-stage transfer learning. IEEE Signal Processing Letters, 2021, 28(7): 1510-1514

[32] Xue Yi-Ming, Yang Bo-Ya, Deng Ya-Qian. Domain adaptational text steganalysis based on transductive learning//Proceedings of the ACM Workshop on Information Hiding and Multimedia Security. California, USA, 2022: 91-96

[33] Wen Juan, Zhang Zi-Wei, Yang Yu. Few-shot text steganalysis based on attentional meta-learner//Proceedings of the ACM Workshop on Information Hiding and Multimedia Security. California, USA, 2022: 97-106

[34] Sabour S, Frosst N, Hinton G. Dynamic routing between capsules//Proceedings of the 31st Conference on Neural Information Processing Systems. Long Beach, USA, 2017: 3856-3866

[35] Chen Ya-Hui. Convolutional Neural Network for Sentence Classification [M. S. dissertation]. University of Waterloo, Canada, 2015

[36] Zhao Wei, Ye Jian-Bo, Yang Min, et al. Investigating capsule networks with dynamic routing for text classification// Proceedings of the Empirical Methods in Natural Language Processing. Brussels, Belgium, 2018: 3110-3119

[37] Xia Cong-Ying, Zhang Chen-Wei, Yan Xiao-Hui, et al. Zero-shot user intent detection via capsule neural networks// Proceedings of the Empirical Methods in Natural Language Processing. Brussels, Belgium, 2018: 3090-3099

[38] Geng Rui-Ying, Li Bin-Hua, Li Yong-Bin, et al. Induction networks for few-shot text classification//Proceedings of the Empirical Methods in Natural Language Processing. Hong Kong, China, 2019: 3904-3913

[39] Li H, Jin S. Text steganalysis based on capsule network with dynamic routing. IETE Technical Review, 2021, 38 (1): 72-81

[40] Lake B M, Salakhutdinov R, Tenenbaum J B. Human-level concept learning through probabilistic program induction. Science, 2015, 350(6266): 1332-1339

[41] Munkhdalai T, Yu H. Meta networks. Machine Learning, 2017, 70: 2554-2563

[42] Mishra N, Rohaninejad M, Chen Xi, et al. A simple neural attentive meta-learner// Proceedings of the International Conference on Learning Representations. Vancouver, Canada, 2018: 1-17

[43] Gu Jia-Tao, Wang Yong, Chen Yun, et al. Meta-learning for low-resource neural machine translation//Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, Belgium, 2018: 3622-3631

[44] Bao Yu-Jia, Wu Meng-Hua, Chang Shi-Yu, et al. Few-shot text classification with distributional signatures// Proceedings of the International Conference on Learning Representations. Ethiopia, Africa, 2020: 1-24

[45] Koch G, Zemel R, Salakhutdinov R. Siamese neural networks for one-shot image recognition//Proceedings of the International Conference on Machine Learning. Lille, France, 2015: 31-38

[46] Luong M. Multi-task sequence to sequence learning//Proceedings of the International Conference on Learning Representations. Puerto Rico, US, 2016: 1-9

[47] Snell J, Swersky K, Zemel R. Prototypical networks for few-shot learning. Neural Information Processing Systems, 2017, 17(10): 4078-4088

[48] Sung F, Yang Yong-Xin, Zhang Li. Learning to compare: Relation network for few-shot learning//Proceedings of the Conference on Computer Vision and Pattern Recognition. Salt Lake, US, 2018: 1199-1208

[49] Ravi S, Larochelle H. Optimization as a model for few-shot learning//Proceedings of the International Conference on Learning Representations. Toulon, France, 2017: 1-11

[50] Nichol A, Schulman J. Reptile: A scalable meta learning algorithm. Computation and Language, 2018, 7(6): 113-127

[51] Yu Mo, Guo Xiao-Xiao, Yi Jin-Feng, et al. Diverse few-shot text classification with multiple metrics//Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics. Louisiana, USA, 2018: 1206-1215

[52] Jiang Xiang, Havaei M, Chartrand G. On the importance of attention in meta-learning for few-shot text classification// Proceedings of the Conference and Workshop on Neural Information Processing Systems. Montréal, Canada, 2018: 53-64

[53] Lin Zhou-Han, Feng Min-Wei, Cicero D S, et al. A structured self-attentive sentence embedding//Proceedings of the International Conference on Learning Representations. Toulon, France, 2017: 1-15

[54] Hinton G E, Krizhevsky A, Wang S D. Transforming auto-encoders//Proceedings of the International Conference on Artificial Neural Networks. Espoo, Finland, 2011: 44-51

[55] Socher R, Chen D, Manning C D, et al. Reasoning with neural tensor networks for knowledge base completion. Neural Information Processing Systems, 2013: 1-10

[56] Wan Sheng-Xian, Lan Yan-Yan, Guo Jia-Feng, et al. A deep architecture for semantic matching with multiple positional sentence representations. Association for the Advancement of Artificial Intelligence, 2016: 2835-2841

[57] Geng Rui-Ying, Jian Ping, Zhang Ying-Xue, et al. Implicit discourse relation identification based on tree structure neural network//Proceedings of the International Conference on Asian Language Processing. Bandung, Indonesia, 2018: 334-337

[58] Li He-He, Wang Jing-Ge, Tang Miao-Miao, et al. Deep sparse rectifier neural networks. International Conference on Artificial Intelligence and Statistics, 2011, 34(7): 1114-1118

[59] Duchi J, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research, 2011, 12(7): 2121-2159

[60] Byrkjeland M, Lichtenberg F G, Gambäck B. Ternary twitter sentiment classification with distant supervision and sentiment specific word embeddings. Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, 2018: 97-106

[61] Lin T Y, Maire M, Belongie S, et al. Microsoft COCO: Common objects in context//Proceedings of the European Conference on Computer Vision. Zurich, Switzerland, 2014: 740-755

[62] Raymond E, Pham P T, Huang D, et al. Learning word vectors for sentiment analysis//Proceedings of the Association for Computational Linguistics: Human Language Technologies. PA, USA, 2007: 142-150

[63] Pennington J, Socher R, Manning C D. GloVe: Global vectors for word representation. Empirical Methods in Natural Language Processing, 2014, 19(5): 1532-1543

[64] Finn C, Abbeel P, Levine S. Model-agnostic meta-learning for fast adaptation of deep networks//Proceedings of the International Conference on Machine Learning. Sydney, Australia, 2017: 1126-1135

YANG Yu, M. S. candidate. Her research interests include information security and text steganalysis.

ZHANG Zi-Wei, M. S. candidate. Her research interests include information security and text steganalysis.

WEN Juan, Ph. D., associate professor. Her research interests include natural language processing and information hiding.

Background

Text Steganalysis is the counter-steganography domain that aims at detecting the existence of steganography within a text, which has become one of the hot topics in the information security area. With the rapid development of neural network language models, text steganography techniques have been significantly developed. Particularly, generation-based text steganography has achieved state-of-the-art performance in generating high-quality texts that are hard to distinguish by both human eyes and computers.

Traditional text steganalysis techniques rely on manually designed features. Their performance is not satisfactory and can only detect a specific steganography algorithm. The current state-of-the-art approach, which is based on deep neural network models, solves the problem of manually designing features. However, the existing text steganalysis models can only detect a certain type of text that has independent and identically distribution with the training data. When the text domains, steganography algorithms, and embedding capacities change, the performance of these algorithms will be reduced. To summarize, the current text steganalysis techniques cannot meet the needs for real-time detection of stego text generated by a variety advanced steganography models. It is vital to develop a compelling and efficient steganalysis algorithm that can be adapted to numerous text steganalysis models under few-shot learning scenarios.

Therefore, in this paper, a capsule network-based approach, named CN-Stega, is proposed for text steganalysis to enhance the model performance in different tasks, making the model achieve fast adaptation in few-shot scenarios. Extensive experiments demonstrate that our model can effectively adapt to various tasks and achieve advanced detection performance in few-shot scenarios.

This work was supported by the National Natural Science Foundation of China (No. 61802410).