中图分类号：　　单位代号：**10280**

密　　级：　　学　　号：**19721441**

上海大学

专业学位硕士学位论文

题目	生成式文本隐写及其检测技术研究

作　　者　易标

学科专业　电子与通信工程

导　　师　吴汉舟

完成日期　2022 年 5 月

姓　　名：易标　　学号：19721441

论文题目：生成式文本隐写及其检测技术研究

上海大学

本论文经答辩委员会全体委员审查,确认符合上海大学硕士学位论文质量要求。

答辩委员会签名：

主任：

委员：

导　　师：

答辩日期：

姓　　名：易标　　学号：19721441

论文题目：生成式文本隐写及其检测技术研究

原 创 性 声 明

本人声明：所呈交的论文是本人在导师指导下进行的研究工作。除了文中特别加以标注和致谢的地方外，论文中不包含其他人已发表或撰写过的研究成果。参与同一工作的其他同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

签　名：_____日 期：_____

本论文使用授权说明

本人完全了解上海大学有关保留、使用学位论文的规定，即：学校有权保留论文及送交论文复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容。

(保密的论文在解密后应遵守此规定)

签 名：_____导师签名：_____日期：_____

上海大学工学硕士学位论文

生成式文本隐写及其检测技术研究

姓　　名：　易标

导　　师：　吴汉舟

学科专业：　电子与通信工程

上海大学通信与信息工程学院

2022 年 5 月

A Dissertation Submitted to Shanghai University for the Degree

of Master in Engineering

Research on Generative Linguistic Steganography and Its Detection

M.A. Candidate：Biao Yi

Supervisor：Hanzhou Wu

Major：Electronic and Communication Engineering

School of Communication and Information Engineering,

Shanghai University

May, 2022

摘　要

文本隐写是一种将机密信息隐藏在文本中从而实现隐蔽通信的技术。传统的修改式文本隐写通过修改预先给定的文本实现机密信息的嵌入。近年来，得益于深度学习和自然语言处理技术的快速发展，生成式文本隐写成为了热点，它利用机密信息直接生成含密文本，而无需事先指定文本，取得了较传统修改式文本隐写更高的嵌入量和隐蔽性。为了对抗生成式文本隐写被不法分子利用且为了从对立面促进生成式文本隐写的发展，生成式文本隐写检测（又称隐写分析）也成为了重要的研究内容。在此背景下，本文针对现有生成式文本隐写及其检测技术中存在的不足进行了研究，主要工作如下：

（1）针对现有生成式文本隐写算法中接收方提取机密信息所需要的附加信息多以及计算复杂度高等问题，提出了一种位置驱动的生成式文本隐写算法。该算法可以自动的生成一段流畅的含密文本，含密文本中固定位置的单词串联即可构成机密信息，从而显著降低机密信息提取的复杂度。该算法利用双向的大规模预训练语言模型，使得在生成含密文本的过程中充分利用上下文语境信息，保障了含密文本的可读性。同时，搭配吉布斯采样通过多轮迭代的方式优化含密文本，使得生成的含密文本更符合自然文本分布。实验结果表明，所提出的方法可以生成流畅的高质量文本，抗隐写分析能力相较于传统生成式文本隐写算法提升约10%。此外，所提算法在接收方行为受限及即时通信场景下具有良好的应用前景。

（2）针对现有生成式文本隐写检测算法中语义建模方式简单和全局信息利用不充分等问题，提出了一种基于图神经网络的生成式文本隐写检测算法。不同于以往方法简单地将文本建模成序列，该方法将文本建模成图结构，从而显示地建模长距离单词间的依存关系和共现信息。同时，利用图神经网络学习文本局部敏感语义和文本间全局相关性，进而实现含密文本的高效检测。实验结果表明所提出的方法可以高效的检测多种生成式文本隐写算法，并且相较于传统方法而言取得了更好的检测效果。

（3）鉴于预训练语言模型具备检测异质文本的能力，提出了一种基于微调预训练语言模型的生成式文本隐写检测算法。该方法先利用自然文本得到预训练语

言模型，再通过微调预训练语言模型的方式将语言模型的先验知识迁移到生成式文本隐写检测分类器中，从而提升其检测性能。实验结果表明，所提出的方法相较于已有方法在检测性能与收敛速度上实现了双重提升，并且随着预训练数据集的增大，检测性能进一步提升，这为应对含密文本收集难度远高于自然文本的现实场景提供了解决方案。

关键词：隐蔽通信，文本隐写，隐写分析，图神经网络，语言模型

ABSTRACT

Linguistic steganography is a technique that hides confidential information in texts to achieve covert communication. The traditional modified linguistic steganography methods realize the embedding of confidential information by modifying a pre-given text. Thanks to the rapid development of deep learning and natural language processing technologies, generative linguistic steganography has become a hot topic. It uses confidential information to generate a stego text without specifying a text in advance and achieves higher embedding rate and concealment than traditional modified linguistic steganography methods. To combat the use of generative linguistic steganography by criminals and promote the development of generative linguistic steganography, generative linguistic steganography detection (also known as steganalysis) has also become an important research topic. This dissertation studies the shortcomings of existing generative linguistic steganography and its detection technology. The main works are as follows:

(1) A location-driven generative linguistic steganography algorithm is proposed to address the problems of much additional information and the high computational complexity required by the receiver to extract confidential information in existing generative linguistic steganography algorithms. The proposed algorithm can automatically generate a fluent stego text. The concatenation of words at fixed positions in the stego text can constitute confidential information, thereby significantly reducing the complexity of confidential information extraction. Specifically, the method uses a bi-directional large-scale pre-trained language model, which makes full use of contextual information in generating stego texts and guarantees the readability of stego texts. At the same time, gibbs sampling is used to optimize the stego texts through multiple iterations so that the generated stego texts are more in line with the distribution of the normal texts. The experimental results show that the proposed method can generate fluency and high-quality stego texts. The resistance ability to steganalysis is

improved by about 10% compared with the traditional generative linguistic steganography algorithms. In addition, the proposed algorithm has good application prospects in the scenarios of restricted receiver behavior and instant messaging.

(2) A generative linguistic steganography detection algorithm based on graph neural network is proposed to address the problems of simple semantic modeling and insufficient utilization of global information in existing generative linguistic steganography detection algorithms, unlike previous methods that model text as a sequence, this method models text as a graph structure, explicitly modeling the dependencies and co-occurrence information between long-distance words. At the same time, we use the graph neural network to learn the local sensitive semantics of text and the global correlation between texts to realize the efficient detection of stego texts. Experimental results show that the proposed method can efficiently detect a variety of generative linguistic steganography algorithms and achieves better detection results than traditional methods.

(3) A generative linguistic steganography detection algorithm based on fine-tuning the pre-trained language model is proposed, given the ability of the pre-trained language model to detect heterogeneous texts. Specifically, the method first obtains a pre-trained language model using normal texts. It then fine-tunes the pre-trained language model to transfer the prior knowledge of the language model into the linguistic steganalysis classifier, thus improving its detection performance. The experimental results show that the proposed method enhances detection performance and convergence speed compared with existing methods. The detection performance is further enhanced as the pre-training dataset increases, providing a solution to the realistic scenario where the collection of stego texts is much more difficult than normal texts.

Keywords: Covert Communication, Linguistic Steganography, Steganalysis, Graph Neural Network, Language Model

目　录

第一章 绪论

1.1 课题来源

本课题来源于国家自然科学基金青年项目“社交网络多用户协同的行为隐写”(项目编号：61902235)。

1.2 研究背景及意义

随着互联网技术的发展与普及，人类的信息交流频率愈发频繁，生产生活方式也发生了翻天覆地的变化。在用户层面，人们利用微博和微信等社交平台进行文本、图像及音视频交流，极大的提高了交流效率，丰富了交流形式。在企业层面，公司利用互联网和大数据等技术，实现精准广告投送，大大提高了生产效益。互联网对于现代人类社会的政治、经济与文化等领域的发展起到了积极的推动作用，但同时也带来了信息安全方面的诸多问题和挑战。例如，如何在互联网中安全地进行通信以及如何对抗非法分子利用互联网进行秘密通信等一系列问题亟待解决，这对于保护个人隐私，维护国防安全具有重要意义。

对于互联网中安全通信问题，人类进行了长时间探索，提出了多种解决方案。其中，研究的最为完备是密码系统，发送方通过加密技术将秘密信息编码成只有接收方通过密钥才能解码出秘密信息的密文，从而保护秘密信息内容不被第三方获取。值得注意的是，加密系统加密后得到的密文通常是无意义的乱码，因此加密系统并不隐藏秘密信息的存在，或者说秘密通信这一行为，这可能会引起攻击者感知到秘密通信行为[1, 2]，从而阻断或者攻击通信行为，具有极大的潜在安全风险。而以隐写术[3](Steganography)为代表的隐蔽通信系统在这方面具有得天独厚的优势，其将秘密信息隐藏在图像和文本等公共载体中，通过公共信道传输[4]，同时要求含密载体与自然载体在感官上无异，不仅仅隐藏秘密信息的内容，也隐藏了秘密信息的存在，具有极高的隐蔽性。

隐写术是一门古老的技术，早在公元前 5 世纪的希腊，就有文献记载有人通过在奴隶剃过发后的光头上书写秘密信息，待其头发长出后，再将秘密信息传递

出去[5]。不同于以往通过巧思在物理载体上隐藏少量的秘密信息，现代网络空间中无时无刻不在进行着大量的图像、文本与音视频交流，这些数字媒体存在大量的信息冗余，为隐写术创造了广阔的实用场景，使其在近几十年得到了飞速的发展，成为了一门成熟的学科。总而言之，隐写术不仅仅享有极高的隐蔽性，同时与现代网络空间的特性高度吻合，使得隐写术具有极大的潜力成为保障现代网络空间信息交互与通信安全的中坚技术。事实上，隐写术也已经成功于应用于实际场景中，例如，密码系统中的高级加密标准[6]在分发和传递加密操作所需要的密钥时，结合隐蔽通信系统实现隐蔽传输，使其更加安全。

隐写术旨在于将秘密信息隐藏在公共载体中而不引起第三方的怀疑从而实现秘密通信，具有很高的隐蔽性。这种技术一方面可以用于保护个人隐私与维护国防安全，但是如果被不法分子利用，那么可能会导致严重的安全问题。例如2001年，美国报道了一起恐怖组织利用隐写术将秘密信息隐藏在互联网的色情及聊天等网站上，用于传递实施恐怖袭击的计划。这也催生出了上文所提出的第二个问题，即“如何对抗非法分子利用互联网进行秘密通信？”。

就如同密码分析技术之于密码技术，为了防止不法分子利用隐写术在网络空间中进行秘密通信，我们很有必要研究高效的隐写分析[7](Steganalysis)技术(又称为隐写检测技术)进行对抗。隐写分析是一种用于检测载体中是否含有秘密信息的技术，通常建模为一个二分类任务，其通过学习含密载体与自然载体的统计分布差异，从而实现将给定的未知载体区分为含密载体或者自然载体，即检测秘密信息的存在。隐写与隐写分析是矛与盾的关系，同时螺旋上升发展，为了对抗强大的隐写技术被违法犯罪者所用以实施秘密通信，以及从对立面促进隐写术的发展，研究高效的隐写分析技术同样十分重要。

1.3 隐写与隐写分析概述

1.3.1 隐写术

隐写术是一种将秘密信息隐藏在图像、文本和音频等公共载体中，通过公共信道进行传输从而实现隐蔽通信的技术。学者西蒙用“囚徒困境”模型[8]给出了

一个易于理解的解释。如图 1.1 所示，Alice 和 Bob 被关押在监狱两个不同的房间中，他们正在商量如何越狱。虽然他们被允许通信，但是同时他们被第三方 Eve 所监视，他们之间交流的信号会被 Eve 审查，如果被 Eve 察觉到异常，那么 Eve 会阻止两人的通信，导致行动失败，如果 Eve 察觉不到异常，那么通信可以正常进行，所以 Alice 和 Bob 必须想办法把秘密信息隐藏到看似无异常的载体中，以实现隐蔽通信。

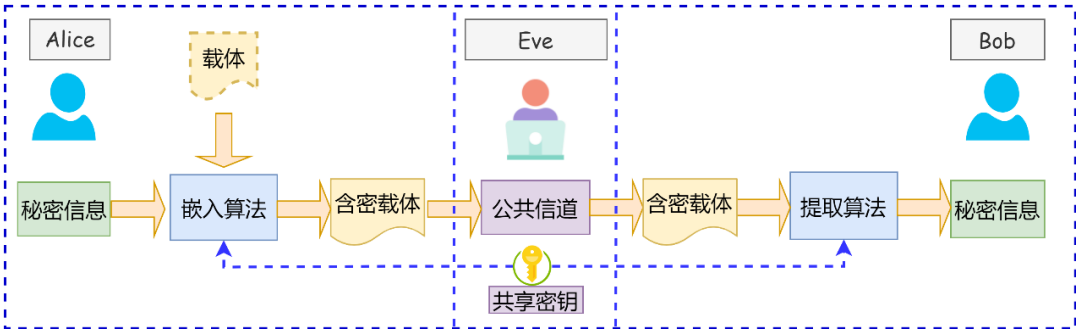

图1.1 隐写术的一般框架

根据隐写术使用的载体不同，可以将其分类为图像隐写、文本隐写、音频隐写及视频隐写等。由于图像信息冗余大和易分发的特性，以图像为载体的隐写术在近年来得到了充分的发展。其中最为经典的方法是最低有效位替换法[9] (Least Significant Bit)，其思想十分朴素，发送方将图像的最低有效位用秘密信息进行替换，接收方接收到该图像后直接取出最低有效位即可获取秘密信息。该方法利用了人类感官的冗余特性，即对于图像细微的像素值变化无法察觉。除了直接在图像的空域[9-12]上修改以嵌入秘密信息，研究者们也研究在图像变换域[13,14]上进行嵌入。

隐写术的评价指标主要为不可感知性、信息嵌入率及计算复杂度等[15,16]。所谓不可感知性，指的被嵌入了秘密信息的载体即含密载体需要能够通过第三方即 Eve 的审查。通常而言我们会从两个方面衡量含密载体的不可感知性，即感官不可感知性和统计不可感知性。感官不可感知性指的是人的感官要能够感觉不出含密载体与自然载体的区别。例如在文本隐写领域，我们希望含密文本依然是流利的，可读性高的，通常用困惑度 (Perplexity, PPL) [16, 17]进行定量衡量。含密载体不仅仅需要通过人的审查，而且需要抵抗机器的分析，这也称之为统计不可感知

性，其衡量的是含密载体与自然载体的统计分布差异。在实际中，通常通过隐写分析实验来定量衡量这一点。

高不可感知性可以确保隐蔽通信更难被察觉，但是同时我们也希望单位载体元素中嵌入的信息量尽可能的多，这就引伸出了信息嵌入率。首先值得注意的是，在现代隐写框架下，通常以嵌入比特流形式的秘密信息为目标。例如对于空域图像隐写而言，通常用比特每像素(bits per pixel, bpp)描述其信息嵌入率，对于文本隐写，则一般使用比特每单词(bits per word, bpw)进行描述。

除此之外，对于即时通信和通信双方行为受监控等严苛场景，我们还会要求隐写算法的时间及空间复杂度尽量的低，所需要的附加信息尽可能的少。

1.3.2 隐写分析

隐写分析是伴随隐写技术的发展而产生的，是隐写术的对立面，其目的是揭露机密信息的存在。隐写分析可以建模为一个二分类任务如图 1.2 所示，其假设检测方(通常称为隐写分析者)可以收集到一些自然载体与含密载体，其首先利用这些有标签的载体训练得到一个分类器，然后使用训练好的分类器判断未知载体是属于含密载体还是自然载体，即检测机密信息存在与否。值得注意的是，我们直观上感觉对于第三方可以收集到含密载体这一假设过于理想，而实际上这是合理的，这与密码学中的 Kerchhoffs 准则同理[18]，只有当隐写术在如此严苛的场景下依然能够逃避机器的分析，才能说明含密载体与自然载体的统计分布一致，才能真正保障隐写术是安全的。

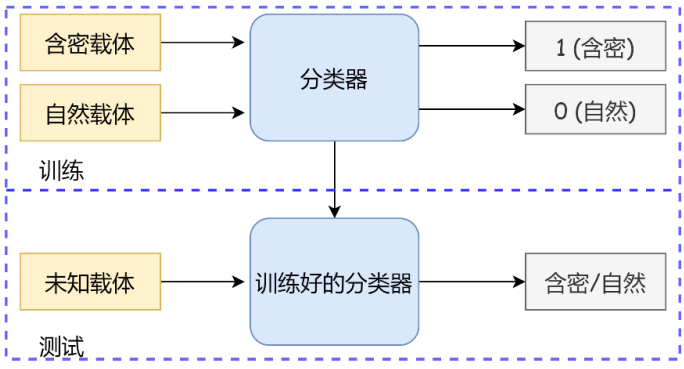

图1.2 隐写分析的一般框架

隐写分析的评价指标与常用的二分类任务指标一致，包括准确率(Accuracy, Acc)、精准率(Precision, Pre)、召回率(Recall, Rec)和 F1 值(F1-score, F1)。隐写分析的目的主要是为了找出含密载体，因此通常将含密载体视为正样本。各个指标的计算方式为：

$$\begin{aligned} \mathrm{Acc} &= (N_{\mathrm{tp}} + N_{\mathrm{tn}}) / (N_{\mathrm{tp}} + N_{\mathrm{fp}} + N_{\mathrm{tn}} + N_{\mathrm{fn}}) \\ \mathrm{Pre} &= N_{\mathrm{tp}} / (N_{\mathrm{tp}} + N_{\mathrm{fp}}) \\ \mathrm{Rec} &= N_{\mathrm{tp}} / (N_{\mathrm{tp}} + N_{\mathrm{fn}}) \\ \mathrm{F1} &= \frac{2 \times \mathrm{Pre} \times \mathrm{Rec}}{\mathrm{Pre} + \mathrm{Rec}} \end{aligned} \tag{1.1}$$

其中 N_{tp}，N_{tn}，N_{fp} 及 N_{fn} 分别表示正确分类的含密载体数，正确分类的自然载体数，错误分类的含密载体数以及错误分类的自然载体数。Acc、Pre、Rec 和 F1 越高表示隐写分析的性能越好，也反应了其检测的隐写术的统计隐蔽性越差。

1.4 文本隐写与隐写分析国内外研究现状

现代网络空间中频繁的文本交流行为为隐写提供了便利，使得以文本为载体的隐蔽通信行为能很好的隐藏在大量的普通文本交流行为中。除此之外，文本隐写在通信过程中还具有很强的鲁棒性，避免了如图像等数字媒体在传输中可能会由于重压缩等操作导致信息提取失败。以上原因促使研究者们探索利用文本隐写技术实现隐蔽通信，并取得了丰硕的成果。

文本隐写的历史十分悠久，例如在我国古代就有藏头诗这一体裁，诗人将秘密信息隐藏在诗中的每个诗句的首位，使得诗本身读起来就是一首正常的诗，只有知道如何提取的人才能正确提取出秘密信息。在隐写术数学化模型和框架[8]提出后，文本隐写开始得到系统的研究，形成了一个严肃的科学分支。近年来，自然语言处理与人工智能技术的快速发展为文本隐写领域注入了新的活力，使其迎来了更广泛的关注，在自然语言处理领域和安全领域的重要期刊与会议中涌现了大量相关工作。

近年来文本隐写的进展又主要集中在无需提供载体文本，由秘密信息直接引导含密文本生成的生成式文本隐写范式下，由于大规模文本数据及神经网络技术的引入，使之在嵌入率、文本可读性以及隐蔽性等方面相较于之前的方法得到了

显著的提升。接下来，本节将详细地阐述文本隐写与隐写分析的发展脉络，同时将着重介绍最新的生成式文本隐写及相关技术。

1.4.1 文本隐写研究现状

文本隐写从实现方式上主要可以分为载体选择式、载体修改式与载体生成式三大类。

(1)载体选择式文本隐写

载体选择式隐写是基于这样一个原理，天然存在的自然载体可能本身就携带秘密通信双方想要传输的秘密信息，因此可以利用载体的这些固有属性实现秘密信息的传递。这个思想最先出现在图像隐写领域[19]，例如想要传递的秘密信息为{1,1,0,1,1,0,1,0}，而某自然图像中某具体位置的像素强度值为 218，转换为比特流则为{1,1,0,1,1,0,1,0}，因此发送方可以直接发送该自然图片给接收方，接收方利用预先共享的位置密钥提取该像素值从而提取出秘密信息。

该思想可以迁移至文本隐写领域，我们可以通过检索在固定位置包含秘密信息的自然文本用于隐蔽通信。不同于图像隐写，现有载体选择式文本隐写通常以嵌入明文秘密信息(如单词或词组)而不是比特流信息为目的，但是由于明文秘密信息过于复杂，不是总能找到符合条件的自然文本。为此，研究者们做了很多探索，首先是构建尽可能多样的文本数据库，从一开始的本地数据库[20-23]到网络数据库[24]，其次是将秘密信息转换为易于检索的高频词[20]，然后在数据库中检索包括该高频词的自然文本进行传输。此外，此类算法经常会使用倒排表等技术提高文本大数据检索效率。

载体选择式文本隐写的优势是可以保证含密文本是自然的，由于不对自然文本进行任何改变，因此其能够抵抗各种文本隐写分析方法。其缺点是信息嵌入量低，而且由于不是总能检索到符合条件的文本，成功率无法得到完全保障。除此之外为了避免在通信时总是传输一样的文本，需要及时更新数据库，以提高系统的安全性，因此总的来说这类方法实用性不强。

(2)载体修改式文本隐写

载体修改式文本隐写指的是在给定的载体文本上进行修改或变换以嵌入秘

密信息。根据修改后得到的含密文本与预先给定的载体文本的语义区别，又可以细分为语义不变的修改式文本隐写和语义变化的修改式文本隐写。

语义不变的修改式文本隐写基于这样一个原理，不同知识背景的人对于同一个事件会有不同的表述方式，这之间就存在信息冗余用于嵌入秘密信息。根据对载体文本修改的粒度不同，又可以细分为词级别[15, 25-28]、短语级别[29]以及句子级别[30-34]。其中词级别的方法如同义词替换法在过去很长一段时间一直是文本隐写的主流方法，通信双方首先构建一个共享的同义词词典如 Wordnet[35],然后对每一组同义词组中的词进行编码，发送方通过选择与秘密信息对应的同义词进行替换以嵌入秘密信息，接收方则利用预先与发送方共享的同义词词典来提取对应的秘密信息。如表 1.1 所示，给定载体文本“*We complete the charitable labor*”，经过同义词替换后，含密文本“*We finish the charitable project*”中成功地嵌入了秘密信息“101”。在此框架下，后续研究者们从扩充同义词词典[28]，采取更好的编码方式如混合基数编码[36]、霍夫曼编码[15]及图编码[15]等进一步提高了算法嵌入率。同时，由于同一同义词组中的同义词语义虽然相近，但是进行替换时并不一定符合语境，因此也有研究者利用基于搭配[26]及基于语言模型[15]等方式测试替换是否合理。同理，短语级别的修改式文本隐写方法[29]通过构造同义短语集，然后编码替换从而实现信息嵌入。

表1.1 基于同义词替换的修改式文本隐写示例

	1-bit	Word		2-bit	Word
We	0	*complete*	the charitable	00	*labor*
	1	*finish*		01	*project*
				10	*task*
				11	*undertaking*

句子级别的方法是通过对载体文本的句法结构进行一定的变换从而嵌入秘密信息，接收方解析出含密文本的语法结构从而恢复秘密信息。一个直观的方法如主被动变换[30]，其将主动句式编码为 0，被字句式编码为 1，接收方通过识别接收到含密句子的主被动从而提取出对应的秘密信息。为了提高嵌入量，研究者们通过改变载体文本的句法结构树[30-34]，然后通过语言生成工具得到符合修改后句法结构的文本表述形式，从而实现信息嵌入，接收方解析出含密文本的句法结构从而恢复秘密信息。除此之外，也有学者利用翻译系统将载体文本翻译成多个

候选项[37-40]，然后通过哈希的方式进行编码选取出含密文本，接收方对含密文本进行同样的哈希操作从而提取出秘密信息。

由于人类语言经历了上千年的发展，以高效的交流为目的进行了多次进化，发展成为了一种高度编码的符号化表示，信息冗余相对较少，比如仅仅修改文本中的少量单词，可能会导致其表达的语义发生巨大的变化，尽管使用语义相近的同义词替换依然可能会导致语句不通顺，因此语义不变的修改式文本隐写方法的信息嵌入量通常很低，平均只有零点几个，最多数个比特每句。

语义可变的修改式文本隐写也是通过对给定的载体文本进行变换从而嵌入秘密信息，但是不对变换后的含密文本的语义进行限制。如 Chang 等人[41]通过改变载体文本的单词顺序产生一系列流畅的文本，然后进行编码并选取含密文本，接收方则从含密文本的单词顺序中解码出秘密信息。Ueoka 等人[42]利用预训练语言模型 BERT[43](Bidirectional Encoder Representations from Transformers)替换载体文本中的单词，并使用块编码的方式嵌入秘密信息从而得到含密文本，接收方利用共享的 BERT 模型重复发送方的操作从而提取出秘密信息。

(3)载体生成式文本隐写

载体生成式文本隐写是一类无需提前给定载体文本，由秘密信息直接引导含密文本生成的方法。其与载体修改式文本隐写方法的区别是无需提前指定载体文本，其与载体选择式文本隐写方法的区别是可以创造自然界中不存在的流畅文本，具有更高信息嵌入率，并能完全保障信息的嵌入。

如图 1.3 所示，该方法首先在大规模语料库上训练得到一个具有文本生成功能的模块即语言模型，然后利用该语言模型结合编码技术在嵌入秘密信息的同时生成可读性高的含密文本，接收方重复发送方的操作从而提取出秘密信息。下文从语言模型与信息嵌入算法两方面分别进行阐述。

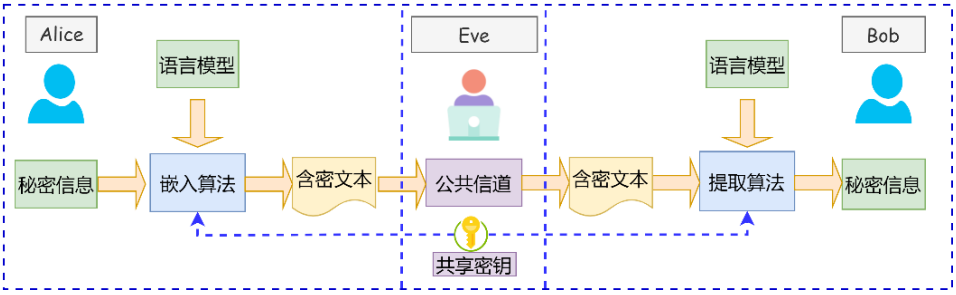

图1.3 生成式文本隐写的一般框架

1）语言模型

语言模型是用来计算一个句子 $\boldsymbol{s}=w_1,w_2,...,w_l$ 成立概率 $p(\boldsymbol{s})$ 的模型，l 表示句子的长度，w_i 表示句中的第 i 个单词。根据链式法则，这个概率可以展开：

$$p(\boldsymbol{s})=p(w_1)p(w_2|w_1)...p(w_l|w_1,w_2,...,w_{l-1}) \tag{1.2}$$

因此，语言模型本质上是在建模一个条件概率分布 $p(w_n|w_1,w_2,...,w_{n-1})$，即根据文本任意的前 n-1 个单词计算第 n 个单词的概率分布，由该条件概率分布就可以计算任意句子存在的概率。建模条件概率分布 $p(w_n|w_1,w_2,...,w_{n-1})$ 的方法多种多样。最初通过马尔可夫模型进行建模，马尔可夫模型基于这样一个假设，即第 n 个单词的概率分布只依赖于前少数个别单词，而不是所有的 n-1 的单词。这里以二阶马尔可夫模型为例，同理可以推广到任意阶马尔可夫模型。二阶马尔可夫模型建模第 n 个单词的条件概率分布时只基于其前 2 个单词，即：

$$p(w_n|w_1,w_2,...,w_{n-1})=p(w_n|w_{n-2},w_{n-1}) \tag{1.3}$$

同时通过从训练语料库中统计频率近似计算该概率：

$$p(w_n|w_{n-2},w_{n-1})=\frac{c(w_{n-2},w_{n-1},w_n)}{c(w_{n-2},w_{n-1})} \tag{1.4}$$

其中，$c(w_{n-2},w_{n-1},w_n)$ 表示词组 $\{w_{n-2},w_{n-1},w_n\}$ 在训练集语料库中出现的频次。

综上可知，马尔可夫模型对条件概率分布建模时进行了近似处理，只能建模短时间的依赖。在深度学习时代，学者们开始使用深度神经网络（Deep Neural Network, DNN）如循环神经网络[44]（Recurrent Neural Network, RNN）、卷积神经网络[45]（Convolutional Neural Network, CNN）和自注意力网络[46]（Transformer）等建模条件概率分布以学习长时间依赖。

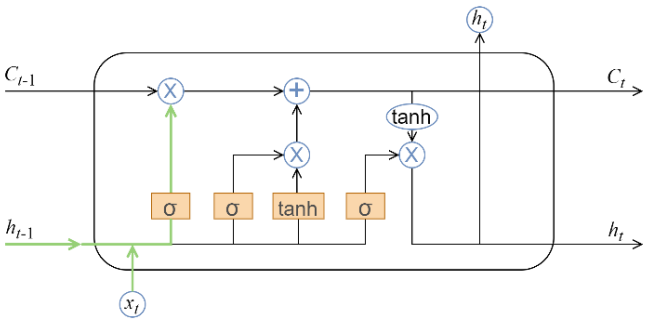

图1.4 LSTM 内部结构

RNN 是自然语言处理领域最常用的神经网络模型，其循环的结构十分适合处理不定长的文本序列。最初始版本的 RNN 具有梯度消失与梯度爆炸等问题，其变体长短期记忆网络[47](Long Short Term Memory, LSTM)可以有效缓解原始 RNN 的问题，在自然语言处理的各项任务上都取得了不错的成绩。LSTM 内部结构如图 1.4 所示，转移方程如下：

$$\begin{cases} \boldsymbol{I}_t = \sigma(\boldsymbol{W}_i \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_i) \\ \boldsymbol{F}_t = \sigma(\boldsymbol{W}_f \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_f) \\ \boldsymbol{C}_t = \boldsymbol{F}_t \cdot \boldsymbol{C}_{t-1} + \boldsymbol{I}_t \cdot \tanh(\boldsymbol{W}_c \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_c) \\ \boldsymbol{O}_t = \sigma(\boldsymbol{W}_o \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{o}_t] + \boldsymbol{b}_o) \\ \boldsymbol{h}_t = \boldsymbol{O}_t \cdot \tanh(\boldsymbol{C}_t) \end{cases} \tag{1.5}$$

其中 $\boldsymbol{x}_t$ 为t时刻输入向量，$\boldsymbol{h}_t$ 表示t时刻的隐状态向量，$\boldsymbol{I}_t$，$\boldsymbol{F}_t$ 分别表示输入门，遗忘门，它们分别控制着有多少输入信息，多少之前时刻的记忆信息保存在 t 时刻的记忆单元$\boldsymbol{C}_t$中，输出门$\boldsymbol{O}_t$控制有多少内部储存信息暴露。$\boldsymbol{W}$ 和 $\boldsymbol{b}$ 均为待训练的参数。

LSTM 在计算t时刻的隐状态$\boldsymbol{h}_t$时，不仅仅会利用t时刻当前的输入 $\boldsymbol{x}_t$，还会利用记忆单元中记忆的 t 时刻之前所有时刻的信息。为了叙述方便，本文将 LSTM 单元的信息传递函数整体表示为 $f_{\text{LSTM}}(*)$，对于 t 时刻的隐状态$\boldsymbol{h}_t$计算，可以简化描述为:

$$\boldsymbol{h}_t = f_{\text{LSTM}}(x_1, x_2, ..., x_t) \tag{1.6}$$

因此，我们可以利用 LSTM 来建模条件概率分布 $p(w_n | w_1, w_2, ..., w_{n-1})$，从而充分利用当前时刻及之前所有时刻的单词信息，解决马尔可夫模型由于近似处理导致只能学习到短时间依赖的弊端。建模方式如下：

$$\begin{cases} \boldsymbol{h}_{n-1} = f_{\text{LSTM}}(\xi(w_1), \xi(w_2), ..., \xi(w_{n-1})) \\ p(w_n | w_1, w_2, ..., w_{n-1}) = \text{SoftMax}(\boldsymbol{W}_h \boldsymbol{h}_{n-1} + \boldsymbol{b}_h) \end{cases} \tag{1.7}$$

其中 $\xi(*)$ 函数为词向量映射函数，将单词映射为输入向量，$\boldsymbol{W}_h$，$\boldsymbol{b}_h$ 为待训练的参数，其目的是将隐向量$\boldsymbol{h}_{n-1}$映射到词表大小的输出空间，然后通过 SoftMax 函数将其转换为概率分布。LSTM 模型的参数首先随机初始化，然后在训练语料库上通过最大化式(1.2)的方式优化模型参数得到最终模型。

在生成式文本隐写领域，最初研究者们采用马尔可夫模型建模语言模型以生成含密文本[48-52]，但是由于马尔可夫模型忽略了远距离单词间的依赖，导致生成的含密文本质量欠佳。后续学者们开始使用 LSTM[15, 53-56]等神经网络建模语言模型，从而显著提高了含密文本的质量。

2)信息嵌入

语言模型利用语料库建模出条件概率分布 $p(w_n|w_1,w_2,...,w_{n-1})$，即根据文本任意的前 *n*-1 个单词计算第 *n* 个单词的概率分布，从而可以计算任意文本成立的概率。同时根据这样一个条件概率分布，也可以实现文本生成。例如，先随机给定一个句首词 *you*，再根据建模的条件概率分布计算下一个词为词表中各个单词的概率分布 $p(w_2|you)$，假设结果为 *are* 概率为 0.5，*like* 概率为 0.3，*job* 概率为 0.0001 等。显然，概率越大的单词说明其更符合上文语境，直接选择概率最大的单词 *are*，然后继续计算下一个词的概率分布 $p(w_3|you,are)$，重复上面的操作，就可以得到高可读性的文本。

通过以上的例子看出，利用语言模型生成每个单词时，有许多合适的候选词可以选择，这提供了信息冗余进行信息隐藏。例如 Fang 等人[53]提出了一种将整个词表随机分为若干个桶，然后对每个桶进行编码，如表 1.2 所示为分 2 桶时的示例，然后在文本生成阶段从与比特流秘密信息匹配的桶中选择概率最大的单词作为结果，此时该算法的嵌入率为 1 个比特每单词。接收方只需要利用与发送方共享的桶编码表即可从含密文本中提取出比特流秘密信息。基于桶编码方法生成的含密文本的质量十分依赖于分桶的策略，例如常用词应该尽可能的平均分配在各个桶中，这样无论秘密信息对应的桶是哪个都可以生成概率比较大，比较符合语境的单词，否则将导致生成的含密文本可读性变差。

表1.2 桶编码示例

桶编码	单词
0	This,am,weather,…
1	was,attaching,today,…

Yang 等人[16]提出一种基于条件概率的编码方法 RNN-Stega，从而很好的解决了桶编码的缺陷。具体做法是在利用语言模型生成文本第 *n* 个单词时，其根据

条件概率选择概率最大的 2^k 个词作为候选池，然后对这 2^k 个词进行编码，编码方式既尝试了简单的定长编码，也测试了与候选词对应概率相关的霍夫曼编码。由于在使用定长编码编码单词时并不考虑其对应的概率，而是将候选池中的单词同等对待，霍夫曼编码则会考虑候选词对应概率，对嵌入率和含密文本的流畅度有一个更好的折中。需要注意的是，该方法需要接收方与发送方共享语言模型参数，才能提取出秘密信息。

随着编码方式的改进，含密文本质量得到了进一步提升。但是研究者们发现，对于含密文本质量的过度优化会导致含密文本与载体文本的分布产生较大的差异，从而导致很差的抗隐写分析能力。Dai 等人[57]指出，导致此问题的原因一方面是神经网络在语料库上拟合文本数据分布时存在建模缺陷，另一方面是由于现有追求文本质量的编码方式使得生成的含密文本符合的条件概率分布 $q(w)$ 与语言模型本身评估的条件概率分布 $p_{\mathrm{LM}}(w)$ 相差太大，这种距离可以用 KL 散度（Kullback-Leibler Divergence）进行度量，即 $D_{\mathrm{KL}}(p_{\mathrm{LM}} \| q)$。针对于第二个问题，Dai 等人提出了一种等待霍夫曼编码策略，也就是在生成每个单词时，先计算 $D_{\mathrm{KL}}(p_{\mathrm{LM}} \| q)$，若其大于预设的阈值 ε，则不嵌入秘密信息，直接从 $p_{\mathrm{LM}}(w)$ 采样进行生成，反之，则正常用霍夫曼编码的方式嵌入秘密信息。通过这种策略，可以有效地降低含密文本与载体文本的分布差异，提高抗隐写分析能力。

Ziegler 等人[58]将比特流秘密信息视为二进制小数，并使用算术编码进行编码，进一步降低了含密文本与载体文本的分布差异。Shen 等人[59]对算术编码进行了进一步的优化，提出了一种自适应算术编码方法，可以自适应的截取条件概率。最近，Zhang 等人[54]提出了一种可证安全的编码方式即自适应分组编码，其将词表分成概率和相等的若干个词组，然后对词组进行编码，最后从秘密信息对应的组中采样单词，取得了更好的抗隐写分析能力与更低的 KL 散度。此外，Zhang 等人[60]提出了一种将语义空间划分成若干个单元，然后对每个单元进行编码，通过生成符合秘密信息所对应单元语义的含密文本以承载信息，这个方法在生成含密文本时直接从整个词表按条件概率进行采样，KL 散度为 0，具有很高的安全性，但是缺点是嵌入量很低。

除了从编码方式进行改进，也有研究者从优化模型架构方面入手提高含密文

本的抗隐写分析能力。Yang 等人[56]提出了一种基于变分自动编码器(Variational Autoencoders, VAE)的生成式文本隐写算法 VAE-Stega，其使用编码器来学习大量自然文本的总体统计分布特征，然后使用解码器生成符合统计语言模型以及自然句子的整体统计分布的含密文本，在确保产生高质量含密文本的同时，使得含密文本与自然文本的分布一致。Zhou 等人[55]使用生成对抗网络(Generative Adversarial Networks, GAN)来解决普通 RNN 文本隐写算法中存在的误差累计等问题，并搭配新的自适应编码方案，以提高含密文本的安全性。

研究者们针对优化含密文本的流畅性及抗隐写分析能力，进行了一系列的探索，取得了不错的成效。但是由于现有生成式文本隐写框架下生成的隐写文本语义是随机的，而在现实场景中发布大量随机语义的文本可能会引起怀疑，因此也有研究者们试图在生成含密文本的过程中控制其语义，例如用关键词[61]、图片[62]、上下文[63]和知识图谱[64]等引导生成含密文本，使得生成的含密文本更符合特定的语境或者场景。

1.4.2 文本隐写分析研究现状

文本隐写分析的目的是判断未知文本是含密文本(又称隐写文本)还是载体文本(又称自然文本)，可以视为一个二分类问题，其关键是通过挖掘含密文本与自然文本的统计分布差异，从而实现含密文本的检测。文本隐写分析的发展过程大致可以分为两个阶段，第一个阶段是利用领域知识手动提取文本特征并搭配机器学习模型进行分类，第二个阶段是利用深度学习模型自动提取特征从而实现端到端的检测。

最初的文本隐写方法模式比较单一，与之对应的隐写分析手段主要是利用领域知识手动提取特征，主要是一些统计特征如词频分布等，并搭配机器学习分类算法实现文本分类。例如，Taskiran 等人[27]首先利用含密文本与载体文本训练一个语言模型来捕获不同类型文本的文本模式，并搭配词频分布等统计特征作为文本的特征表示，最后使用支持向量机将文本分类为含密文本或载体文本。Yang 等人[65]根据语言学理论，定义了 57 个基本统计特征用来表示文本，并利用免疫机制选取合适的特征以实现含密文本的检测。由于最初的生成式文本隐写算法生成

的含密文本可读性很差，Meng 等人[66]提出了一种基于统计语言模型的文本隐写检测算法。他们利用语言模型计算载体文本和含密文本的困惑度，然后通过设置阈值来判断未知文本中是否包含秘密信息。Samanta 等人[67]提出了一种基于贝叶斯估计与相关系数法的文本隐写分析方法。Dinet 等人[68]提出了一种形式化的遗传算法用于检测输入文本是否含有秘密信息。Chen 等[69]人分析了含密文本与载体文本在单词相关性统计分布方面存在的差异，依此提出了一种利用 N 窗口互信息计算文本中常用词之间的相关性统计特征，结合支持向量机来检测含密文本。Xiang 等人[70]基于文本中的同义词与其对应的词频信息，为文本构造了一组高效的特征向量，搭配支持向量机从而有效的对抗基于同义词替换的文本隐写方法。

随着深度学习以及自然语言处理技术的飞速发展，生成式文本隐写取得了惊人的成绩，其在维持高嵌入率的同时可以生成高质量的含密文本，仅仅依靠领域专业知识手动提取特征已经不足以有效的进行应对。因此，研究者们开始研究基于深度神经网络的文本隐写分析方法，这些方法可以自动提取深层语义特征，以一种通用的，端到端的方式实现文本隐写分析。

Yang 等人[71]最先将深度学习技术引入文本隐写分析领域，其首先将单词映射为多维稠密向量，然后使用一个全连接层来提取文本的语义特征，由于该模型并不能建模单词间的先后顺序，因此额外增加了 *n* 元语法特征，最终将提取到的特征送入到一个 SoftMax 分类器中进行分类。考虑到 CNN 强大的特征提取能力，Wen 等人[72]先利用词嵌入层将单词映射为词向量，然后使用不同尺寸的一维卷积核提取文本中不同长度的 *n* 元语法特征，并联结所有提取到的特征作为文本最终的特征表示，最后搭配 SoftMax 分类器实现分类。类似地，Yang 等人[73, 74]分析了文本中单词之间存在的各种关联，提出了基于 CNN 进行窗口滑动提取文本特征的方案。后续研究者们开始引入擅长处理序列信号的 RNN，如 Yang 等人[75]将文本映射到向量空间，然后利用 LSTM 提取特征进行分类，后续也有学者提出基于 LSTM 与特征金字塔结合[76]的方案。由于 CNN 擅长提取文本局部特征，而 LSTM 则擅长提取文本全局特征，因此研究者们开始尝试融合这两种网络进行隐写分析，如 Niu 等人[77]提出了一种基于双向长短期记忆网络（Bidirectional Long Short-Term Memory Network, BiLSTM）结合非对称卷积核的文本隐写分析方法，Bao 等人[78]提出了一种 BiLSTM 结合注意力机制（Attention Mechanism）及 CNN

的特征提取结构，从而实现文本隐写分析。

1.5 本文工作及论文安排

1.5.1 本文工作

本文对文本隐写技术的发展脉络进行了详细的梳理，其中，由于深度学习技术及自然语言处理技术的快速发展，生成式文本隐写成为了最新的研究热点，其在嵌入率及隐蔽性等性能上相较于传统方法而言提升显著。但是，现有生成式文本隐写方法都以嵌入比特级秘密信息为目的，使得接收方在提取机密信息时需要进行大量的计算和消耗较长的时延，为此本文提出了一种位置驱动的生成式文本隐写算法，直接嵌入单词级秘密信息从而显著缓解了该问题。同时，为了对抗不法分子利用生成式文本隐写技术进行秘密通信且为了从对立面促进生成式文本隐写的发展，本文也详细的调研了现有文本隐写分析技术的发展现状，并在此基础上提出了两种更高效的生成式文本隐写检测方法。本文的具体工作如下：

(1)提出了一种基于 BERT 与吉布斯采样的位置驱动型生成式文本隐写算法。现有生成式文本隐写方法以嵌入比特形式的秘密信息为目的，接收方需要通过复杂的解码操作以及很长的时延才能提取出秘密信息的具体内容。这在接收方行为受限及即时通信场景下十分不适用，因此本文提出了一种直接嵌入单词级秘密信息的生成式文本隐写算法。该方法利用双向的大规模预训练语言模型，使得在生成含密文本的过程中充分利用上下文语境信息，保障了含密文本的可读性，搭配吉布斯采样通过多轮迭代的方式优化含密文本，使得生成的含密文本更符合自然文本分布。同时接收方无需复杂且可疑的解码操作，只需要通过简单的位置密钥即可提取出单词级秘密信息，在接收方行为受限以及即时通信场景下具有很高的实用价值。

(2)提出了一种基于图神经网络的生成式文本隐写检测算法。考虑到文本中存在丰富的依存关系，以往的方法仅仅将文本建模成序列不足以充分挖掘这部分信息。因此，本文将文本建模成表征能力更强的图结构，图中节点表示单词，图中的边表示单词与单词之间的关联强度，从而显示地建模单词间的依存关系。同

时，利用图神经网络学习文本局部敏感语义和文本间全局相关性，进而实现含密文本的高效检测。实验结果表明所提出的方法可以高效的检测多种生成式文本隐写算法，相较于传统的方法而言取得了更好的检测性能。

(3)提出了一种基于微调预训练语言模型的生成式文本隐写检测算法。生成式文本隐写需要通过一个训练好的语言模型实现信息嵌入，因此生成的含密文本的统计特征会不可避免的暴露给该语言模型。我们通过进一步的实验证实了语言模型具有检测含密文本的能力，并依此设计了一种检测方法。该方法先在自然文本上预训练语言模型以学习先验知识，再通过微调预训练语言模型的方式将语言模型的先验知识迁移到生成式文本隐写检测分类器中，从而提升其检测性能。实验结果表明，所提出的方法相较于已有方法在检测性能与收敛速度上实现了双重提升，并且随着预训练数据集的增大，检测性能进一步提升，这为应对含密文本收集难度远高于自然文本的现实场景提供了解决方案。

1.5.2 论文安排

本文各章内容组织如下：

第一章介绍了隐写术与隐写分析的研究背景及基本框架；并介绍了文本隐写与隐写分析的研究现状；然后概括了本论文主要的研究工作和结构。

第二章提出了一种基于 BERT 与吉布斯采样的位置驱动型生成式文本隐写算法，并进行了实验分析，最终给出相应的结论。

第三章提出了一种基于图神经网络的文本隐写分析方法，并进行了实验分析，最终给出相应的结论。

第四章提出了一个基于微调预训练语言模型的高效文本隐写分析方法，并进行了实验分析，最终给出相应的结论。

第五章对本文的研究工作进行了总结，并对下一步研究进行了展望。

第二章 基于位置驱动的生成式文本隐写

2.1 引言

网络空间中频繁的文本交流行为以及文本在传输过程中固有的鲁棒性，使得文本隐写成为了保障互联网安全通信的重要技术。在最初，修改式文本隐写如同义词替换[15]是文本隐写领域的主流方法，其主要通过要求含密文本与载体文本的语义一致来保障含密文本的隐蔽性。但是由于文本高度编码和低冗余的特性，使得此类方法存在信息嵌入率低与文本可读性差等缺点，无法满足实际通信需求。近年来，无需预先给定载体文本，直接由秘密信息引导含密文本生成的生成式文本隐写[16]大放异彩，在信息嵌入率和文本可读性等方面相较于修改式方法而言提升显著，具有广阔的应用前景。

现有的生成式文本隐写方法主要是基于这样一个框架，发送方首先将明文秘密信息编码成二进制比特流，然后由比特流引导语言模型生成含密文本。接收方收到含密文本后，先利用共享的语言模型计算以提取比特流秘密信息，最终解码出明文秘密信息。因此，现有的生成式文本隐写方法在提取秘密信息时需要很长的时延，无法适用于即时通信场景。同时，由于接收方需要与发送方共享大量的边信息如模型参数等，然后通过复杂的计算以提取秘密信息，这可能会暴露接收方的身份，不适用于接收方行为受监控的场景。

这促使我们提出了一种以直接嵌入词级别机密信息为目标的位置驱动型生成式文本隐写方法，如图 2.1 所示，该算法可以自动的生成一段流畅的含密文本，含密文本中固定位置的单词串联即可构成机密信息。相较于传统比特级生成式文本隐写方法，其避免了复杂的秘密信息提取和解码过程，大大降低了信息提取算法的空间复杂度与时间复杂度，非常适合接收方行为受限及即时通信场景。通过社交网络发送隐蔽的含密文本，大量的普通文本社交行为很容易掩盖隐写行为。同时，一旦接收者观察到含密文本，他可以保持沉默并根据位置密钥迅速提取秘密信息，而不与任何人进行任何可疑的交互或执行任何可疑的数据解码操作，从而隐藏接收者的真实身份。值得注意的是，载体选择式文本隐写方法[20,21]也以嵌

入词级别秘密信息为目的，但是其无法完全保证秘密信息的成功嵌入，而且信息嵌入率低，实用性较差。

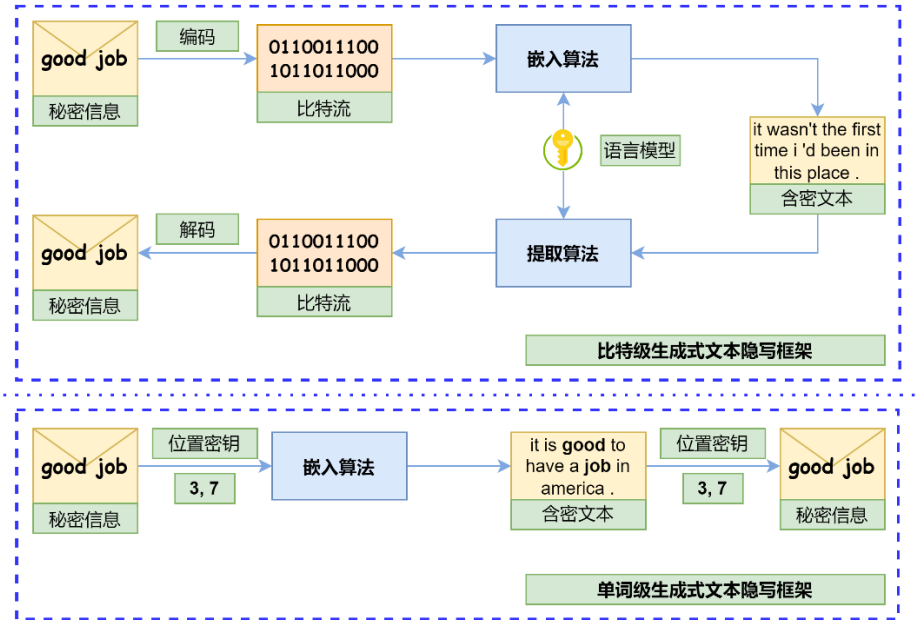

图2.1 比特级生成式文本隐写与单词级生成式文本隐写框架对比

在本文提出的方法中，首先构造一个用于嵌入词级别机密信息的输入模板，然后使用 BERT[43]和吉布斯采样[79-81]生成流畅且安全的含密文本，含密文本中固定位置的单词串联即可构成机密信息，从而显著降低机密信息提取的复杂度。该方法利用双向的大规模预训练语言模型 BERT，使得在生成含密文本的过程中充分利用上下文语境信息，保障了含密文本的可读性。同时，搭配吉布斯采样通过多轮迭代的方式优化含密文本，使得生成的含密文本更符合自然文本分布。后续实验结果显示，所提出的方法可以生成流畅的高质量含密文本，同时抗隐写分析能力相较于传统生成式文本隐写算法提升约 10%。

2.2 相关技术简介

2.2.1 BERT

近年来，预训练语言模型如 BERT[43]和 RoBERTa[82]（A Robustly Optimized BERT Pretraining Approach）等在大范围的自然语言处理任务上取得了最佳的性能，同时其成功也影响到了计算机视觉等其他领域[83]。这里主要介绍后文涉及到

的 BERT，从模型结构上来讲，BERT 由多个 Transformer Encoder[46]模块堆叠而成，Transformer Encoder 的核心结构则为多头自注意力(Multi-head Self-attention, MHS)模块。输入一个长度为n特征向量序列$\boldsymbol{X}$，序列中第i个向量$\boldsymbol{X}_i \in R^{1\times d_{in}}$通过多头注意力模块计算输出的方式为：

$$\mathrm{MHS}(\boldsymbol{X}_i) = Concat(\boldsymbol{h}_1,...,\boldsymbol{h}_m)\boldsymbol{W}^O, i \in \{1,...,n\} \tag{2.1}$$

其中

$$\begin{aligned} \boldsymbol{h}_j &= Attn(\boldsymbol{X}_i\boldsymbol{W}_j^Q, \boldsymbol{X}_i\boldsymbol{W}_j^K, \boldsymbol{X}_i\boldsymbol{W}_j^V),\ j = 1,...,m \\ Attn(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) &= \mathrm{softmax}(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{\boldsymbol{d}_k}})\boldsymbol{V} \end{aligned} \tag{2.2}$$

其中$\boldsymbol{W}_j^Q, \boldsymbol{W}_j^K \in R^{d_{in}\times d_k}$，$\boldsymbol{W}_j^V \in R^{d_{in}\times d_v}$，$\boldsymbol{W}^O \in R^{d_{m\times d_v}\times d_{out}}$是待训练的参数，$\boldsymbol{Q},\boldsymbol{K} \in R^{d_k\times 1}$和$\boldsymbol{V} \in R^{d_v\times 1}$是用于计算自注意力的辅助变量，$m$为多头注意力模块头的数量。

基于多头注意力模块，进一步介绍 Transformer Encoder 框架。Transformer Encoder 由 6 个重复的层组成，由图 2.2 虚线框所示，第一层的输入是输入文本的词向量与位置向量之和，二至六层每一层的输入是上一层的输出。每一层又由四个部分组成，第一个部分为上文所述的多头注意力模块，第二部分为第一部分的输入与输出相加，即残差连接[84]，然后进行正则化，第三个部分为一个前馈神经网络(Feed Forward Neural Networks, FFNN)，第四部分为第三部分的输入与输出相加然后进行正则化。

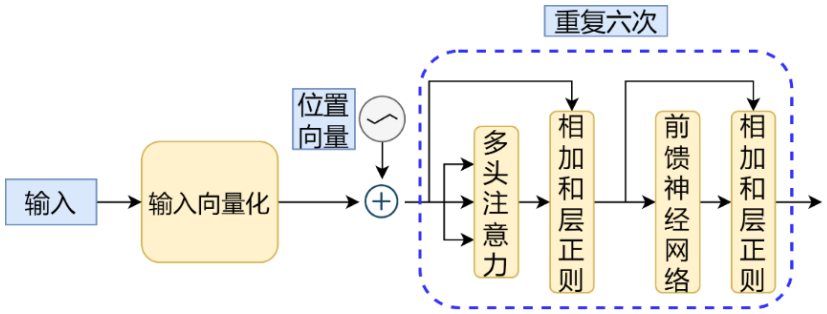

图2.2 Transformer Encoder 内部结构

如图 2.3 所示，BERT 是由 N 个 Transformer Encoder 堆叠而成，较小版本的 $\mathrm{BERT}_{\mathrm{base}}$ 中 N 为 12，大约包含 1.1 亿个参数，较大版本的 $\mathrm{BERT}_{\mathrm{large}}$ 中 N 为 24，大约包含 3.4 亿个参数。与 Transformer 不同的是，BERT 输入除了词向量与位置向量，另外添加了句子分割向量以区分不同的句子。

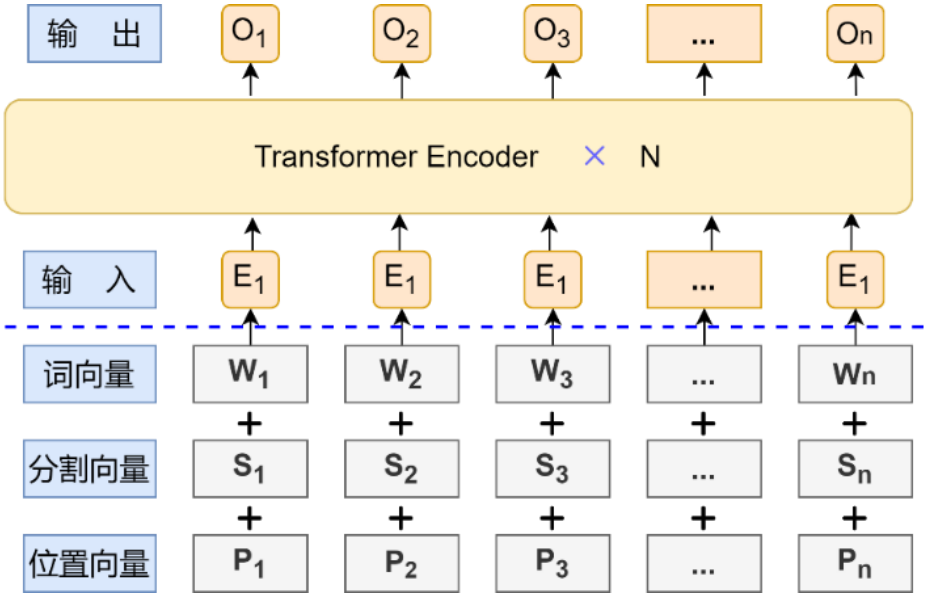

图2.3 BERT 内部结构

不同于传统的语言模型根据文本上文预测下文，BERT 主要采用的是掩码语言模型（Masked Language Model, MLM）作为训练目标，充分的利用了双向的语境信息。具体的，对于每一段训练文本，对其中的部分单词使用一个特殊符号“[MASK]”替换，掩码语言模型的目标是根据上下文恢复出被替换的单词。训练之后的结果是，BERT 可以为被掩盖的单词计算概率分布，从而实现完型填空。

2.2.2 吉布斯采样

吉布斯采样[79-81]是一种马尔可夫蒙特卡洛算法，其目的是从难以直接采样的复杂多元概率分布中采样。假设想要从中采样的目标分布是 $p(z_1, z_2, ..., z_t)$，吉布斯采样进行如下操作以实现采样，首先从一个初始状态 $\boldsymbol{z}^{(0)} = (z_1^{(0)}, z_2^{(0)}, ..., z_t^{(0)})$ 出发，然后进行 T 次迭代完成采样。在每次迭代过程中，算法根据全条件概率分布 $p(z_j^{(i)} \mid z_{<j}^{(i)}, z_{>j}^{(i-1)}) = p(z_j^{(i)} \mid z_1^{(i)}, z_2^{(i)}, ..., z_{j-1}^{(i)}, z_{j+1}^{(i-1)}, ..., z_t^{(i-1)})$ 有序的更新多元变量中所有的元素。在经历了 T 轮迭代后，可以认为 $\boldsymbol{z}^{(T)} = (z_1^{(T)}, z_2^{(T)}, ..., z_t^{(T)})$ 是从目标分布的近似分布中采样得到。

2.3 基于位置驱动的生成式文本隐写

2.3.1 问题描述

本节将描述了一个单词级生成式文本隐写框架，它直接根据位置密钥 $\boldsymbol{p}=\{p_1,p_2,...,p_n\}$ 在生成的含密文本 $\boldsymbol{s}=\{s_1,s_2,...,s_m\}$ 中隐藏单词级别秘密信息 $\boldsymbol{w}=\{w_1,w_2,...,w_n\}$。 p_i 表示 $\boldsymbol{p}$ 中的第 i 个元素， s_i 表示 $\boldsymbol{s}$ 中的第 i 个单词， w_i 表示 $\boldsymbol{w}$ 中的第 i 个单词，$\boldsymbol{s}$ 与 $\boldsymbol{w}$ 中的所有单词都来自于词表 $\boldsymbol{V}$。同时这里满足 $1<n\le m$ 以及 $1\le p_1<p_2<...<p_n\le m$。为了成功实现秘密信息嵌入，含密文本需要满足：

$$s_{p_i}=w_i,\forall i\in\{1,...,n\} \tag{2.3}$$

例如，发送方想要根据位置密钥 $\boldsymbol{p}=\{3,7\}$ 发送秘密信息 $\boldsymbol{w}=\{good,job\}$。那么他需要构建一个这样的文本生成算法，输入 $\boldsymbol{w}$， $\boldsymbol{p}$，输出满足条件的含密文本 $\boldsymbol{s}$，比如在这里 $\boldsymbol{s}$ 可以是“*it is **good** to have a **job** in Shanghai .*”。接收方收到 $\boldsymbol{s}$ 后，根据预先共享的位置密钥 $\boldsymbol{p}$ 直接提取出秘密信息 $\boldsymbol{w}$，对于上面的例子，接收方直接根据位置密钥提取出含密文本的第 3 个与第 7 个单词即可得到秘密信息，而不需要复杂的解码过程，这非常适用于接收方行为受限制以及即时通信场景。该框架的核心问题是发送方如何根据 $\boldsymbol{w}$， $\boldsymbol{p}$ 生成流利且隐蔽的含密文本 $\boldsymbol{s}$。形式化的描述，本文的目标是最大化如下条件概率：

$$\boldsymbol{s}^*=\arg\max p(\boldsymbol{s}|\boldsymbol{w},\boldsymbol{p}) \tag{2.4}$$

上述生成式文本隐写框架与现有的文本隐写方法相比具有显著的差异，也带了很大的挑战，主要体现在：

(1)传统的修改式文本隐写主要通过逼近给定的载体文本的语义来保证含密文本的隐蔽性。本文方法无需指定载体文本，直接通过秘密信息引导生成含密文本，在这种情况下保障含密文本的流畅性与隐蔽性具有更大的难度。

(2)传统的生成式文本隐写算法通常使用从左到右逐字生成的方式生成含密文本，其利用一个预先训练好的自回归式语言模型，然后轻微改变文本生成过程中的单词挑选策略，比特级秘密信息就可以轻松的实现嵌入。换句话说，含密文本在生成过程没有严格的限制。然而，在本文所提框架中，其会提前在指定的位

置固定一些单词，这将导致有序逐字的自回归建模策略在生成含密文本过程中生成非常糟糕的文本。

(3)一个比较直接的想法是通过预训练掩码语言模型如 BERT 进行完型填空的方式来解决本文所提问题。然而，这类模型在预训练过程中的单词掩盖率往往很低，一般在 15%左右，也就是说其任务是给定文本中的大部分词填空少部分未给定的词。而本文所提生成式文本隐写框架的需求则与之相反，即给定少量秘密信息单词，填充其余大部分的空白位置以得到流畅且隐蔽的含密文本。因此，两个任务之间的差距导致直接使用 BERT 类的掩码预训练模型解决本文问题会生成可读性很差的含密文本。

通过以上分析，可以得出本文所提框架与传统的文本隐写方法具有很大的差异，同时以前的建模方法也不足以解决此问题。这促使我们提出了一个新的基于 BERT 与吉布斯采样的生成式文本隐写算法来解决此问题，接下来就机密信息嵌入与机密信息提取这两个阶段进行详细阐述。

2.3.2 机密信息嵌入

机密信息嵌入过程即是通过秘密信息引导含密文本生成的过程，目标为式(2.2)，使用吉布斯采样处理该式，那么此时希望从中采样的目标分布是：

$$p(\boldsymbol{s}|\boldsymbol{w},\boldsymbol{p}) = p(s_1, s_2, ..., s_m|\boldsymbol{w},\boldsymbol{p}) \tag{2.5}$$

按照前面所述的吉布斯采样的步骤，在第 i 轮迭代更新文本中第 j 个单词所使用的全条件概率分布为：

$$\begin{aligned} p(s_j^{(i)}|s_{<j}^{(i)}, s_{>j}^{(i-1)}, \boldsymbol{w}, \boldsymbol{p}) &= \frac{p(s_j^{(i)}, s_{<j}^{(i)}, s_{>j}^{(i-1)}, \boldsymbol{w}, \boldsymbol{p})}{p(s_{<j}^{(i)}, s_{>j}^{(i-1)}, \boldsymbol{w}, \boldsymbol{p})} \\ &\propto p(s_j^{(i)}, s_{<j}^{(i)}, s_{>j}^{(i-1)}, \boldsymbol{w}, \boldsymbol{p}) \\ &= p(s_{<j}^{(i)}, s_{>j}^{(i-1)}) p(s_j^{(i)}|s_{<j}^{(i)}, s_{>j}^{(i-1)}) p(\boldsymbol{w}, \boldsymbol{p}|s_j^{(i)}, s_{<j}^{(i)}, s_{>j}^{(i-1)}) \\ &\propto p(s_j^{(i)}|s_{<j}^{(i)}, s_{>j}^{(i-1)}) p(\boldsymbol{w}, \boldsymbol{p}|s_j^{(i)}, s_{<j}^{(i)}, s_{>j}^{(i-1)}) \end{aligned} \tag{2.6}$$

式(2.6)可以分解为两个部分，第一部分 $p(s_j^{(i)}|s_{<j}^{(i)}, s_{>j}^{(i-1)})$ 可以通过掩码预训练语言模型进行评估，第二部分 $p(\boldsymbol{w}, \boldsymbol{p}|s_j^{(i)}, s_{<j}^{(i)}, s_{>j}^{(i-1)})$ 可以利用一个二分决策函数进行处理。对于第一个部分，这与 BERT 模型的预训练目标掩码语言模型任务十分

相似，因此可以利用训练好的 BERT 模型及其掩码语言模型头部进行计算以评估此项：

$$p(s_j^{(i)} \mid s_{<j}^{(i)}, s_{>j}^{(i-1)}) \approx \text{BERT\_MLM}(s_j^{(i)} \mid s_{<j}^{(i)}, s_{>j}^{(i-1)}) \tag{2.7}$$

第二部分表示文本 $\boldsymbol{s}$ 与条件 $(\boldsymbol{w}, \boldsymbol{p})$ 的匹配程度。如果 $\boldsymbol{s}$ 符合条件，则表示 $\boldsymbol{s}$ 成功地嵌入了秘密信息，该项值为 1，否则，则表示 $\boldsymbol{s}$ 没有嵌入成功，该项值为 0。注意到，如果该项为 0，那么会导致采样不成功。因此，为了提升采样效率，本文对于所有的 $i \in [1,n]$，直接提前设置 $s_{p_i} = w_i$，并且固定这些位置的单词不参与到后续的迭代更新，从而确保第二部分的值始终为 1。

综上分析，可以通过使用 BERT_MLM 与决策函数评估全条件概率，进而使用吉布斯采样生成含密文本，嵌入算法伪代码如算法 1 所示。

算法 1 信息嵌入伪代码

输入：$\boldsymbol{w}$，$\boldsymbol{p}$，BERT_MLM，迭代轮数 T

输出：含密文本 $\boldsymbol{s}$

1. 初始化 $\boldsymbol{s}^{(0)} = \{s_1^{(0)}, s_2^{(0)}, ..., s_m^{(0)}\}$ 为一个含有 m 个“[MASK]”的文本，这里 $m \geq \max(\boldsymbol{p})$
2. 对于所有的 $i \in [1,n]$，令 $s_{p_i}^{(0)} = w_i$
3. for $i = 1, 2, ..., T$ do
4. 　对于所有的 $j \in [1,m]$，令 $s_j^{(i)} = s_j^{(i-1)}$
5. 　for $j = 1, 2, ..., m$ do
6. 　　if $j \notin \boldsymbol{p}$ then
7. 　　使用 BERT_MLM 采样得到 $s_j^{(i)}$
8. 　　更新 $s_j^{(i)}$ 为采样的结果
9. 　　end if
10. 　end for
11. end for
12. 返回含密文本 $\boldsymbol{s} = \{s_1^{(T)}, s_2^{(T)}, ..., s_m^{(T)}\}$

算法的输入为待嵌入的单词集合 $\boldsymbol{w}$ 及对应的位置集合 $\boldsymbol{p}$，吉布斯采样迭代轮数 T。首先，构造一个输入模版 $\boldsymbol{s}^{(0)}$，并保证其长度大于等于 $\boldsymbol{p}$ 中的最大值以保证机密信息可以全部实现嵌入，模板中 p_i 处单词设置为相应的机密信息 w_i，其余位置均设置为“[MASK]”。然后利用 BERT_MLM 对 $\boldsymbol{s}^{(0)}$ 中的“[MASK]”进行逐字采样填充，直至整个句子填充完成，然后重复这个过程 T 轮得到整个含密文本 $\boldsymbol{s}$。在对每个位置采样单词时，首先使用 BERT_MLM 得到候选单词(整个词

表)及每个单词对应的概率，然后用 Top k 采样策略进行采样。Top k 采样指的是在采样过程中，首先从目标概率分布中选择概率值最大的 k 个元素，然后对其概率进行重新归一化构造一个新的概率分布，最终从这个新的概率分布中采样单词。

图 2.4 给出了一个示例，在示例中，发送方意图在文本的第 3 个位置嵌入秘密信息“*run*”，其首先随机选择一个大于 3 的整数作为含密文本的长度，这里为 4，然后构造一个输入模板“[M] [M] *run* [M]”，“[M]”表示“[MASK]”。然后，使用 BERT 依次为输入模板中的“[M]”采样单词并进行替换，经过多轮迭代后得到流畅的含密文本“*the tears run .*”。

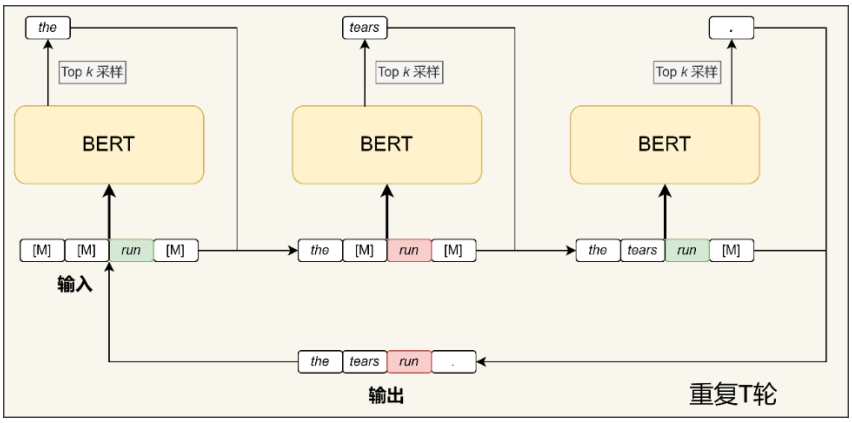

图2.4 机密信息嵌入示例

2.3.3 机密信息提取

对于接收者而言，直接根据共享的位置密钥集合 $\boldsymbol{p}$ 从含密文本 $\boldsymbol{s}$ 中提取对应的单词串联即可重建秘密信息。信息提取算法所需的时间复杂度和空间复杂度极低，这使得本文所提方法非常适用于接收方行为受限制以及即时通信场景。

2.4 实验与分析

2.4.1 实验设置

本文使用 Hugging face 开源[85]的 transformers 工具包中预训练好的模型参数 $\text{BERT}_{\text{base,uncased}}$ 进行含密文本生成。同时，我们从数据集 BookCorpus[86]中随机选择 10000 个自然文本句子作为载体文本，对于每一个句子，随机选择 $c > 0$ 个单

词与其对应的位置来构建秘密信息 $\boldsymbol{w}$ 与位置密钥 $\boldsymbol{p}$，对于除嵌入秘密信息外其他的位置，用“[MASK]”符号替换从而成功构造出输入模板，最后输入到本文所提算法中生成含密文本，显然此处生成的含密文本的长度与载体文本句子的长度相等。为了验证 Top k 采样时 k 的大小对于含密文本性能的影响，因此在生成含密文本时测试了不同的 k 值设置。综上所述，对于不同的 c 和 k，均会生成 10000 段含密文本。实际上，本文所提的生成式文本隐写方法是不需要提供载体文本的，此处为了更好的进行实验对比，所以用载体文本来构建秘密信息与位置密钥。本文采用单位句子嵌入的单词个数（Tokens Per Text, TPT）来衡量算法的信息嵌入率，易知 TPT=c。

同时，本文还增加了一个基准对照组，那就是不使用吉布斯采样，而直接使用 BERT 一步并行地采样填充除预设秘密信息单词外的所有“[MASK]”，称之为 BERT-Only，与之对应本文所提方法称之为 BERT-Gibbs。

2.4.2 感知隐蔽性分析

含密文本的流畅度或者可读性是保障文本隐写算法安全的第一道关卡，只有这样才能避免第三方人员察觉到异常，这通常也称之为感知隐蔽性。本文使用自然语言处理领域中最常用的方案，即用参数量为 117M 的预训练语言模型 GPT2[87]（Generative Pre-trained Transformer 2）来计算每个句子的困惑度（Perplexity, PPL），最终计算 10000 句子的平均 PPL 作为评判指标，PPL 越低则代表文本越流畅。假设 $\boldsymbol{s}=w_1,w_2,...,w_l$ 是一段文本，l 是该文本的长度，w_i 表示文本的第 i 个单词，GPT2 通过如下的方式计算该文本的 PPL：

$$\mathrm{PPL}(\boldsymbol{s})=2^{-\frac{1}{l}\times\log_2 p(w_1,w_2,...,w_l)} \tag{2.8}$$

对于不同方法和不同参数设置下生成的含密文本，平均 PPL 的计算结果如表 2.1 所示，首先，对于不同的隐写方法，随着对于采样候选词词表 k 的扩充，生成的含密文本的流畅度会逐渐降低，这点是符合直觉的，因为随着采样词表的扩充，低概率的单词就有可能被采样到。其次，相较于载体文本的 PPL，可以发现本文所提算法 BERT-Gibbs 在 $k\le 20$ 时实现了相当不错的流畅性保障。最后，可以发现，之所以 BERT-Gibbs 可以生成如此高质量的文本，吉布斯采样起到了

相当关键的作用。BERT-Only 生成的含密文本与载体文本的质量相差甚远，但是使用吉布斯采样后文本的流畅度得到了很大的提升。

表2.1 不同隐写方法在不同嵌入率下的文本困惑度比较

嵌入率	隐写方法	Top-k 采样策略			
		k =1	k =10	k =20	k =|V|
1 TPT	BERT-Only	290.57	549.35	662.29	3531.62
	BERT-Gibbs	**125.79**	**182.68**	**209.72**	**816.88**
2 TPT	BERT-Only	358.23	588.49	683.83	2761.82
	BERT-Gibbs	**127.47**	**183.56**	**208.62**	**688.12**
3 TPT	BERT-Only	371.14	554.91	642.60	2083.29
	BERT-Gibbs	**133.01**	**188.24**	**211.17**	**607.80**
4 TPT	BERT-Only	352.54	503.40	566.63	1549.22
	BERT-Gibbs	**139.30**	**190.61**	**212.79**	**519.11**
载体文本		207.94			

同时本文也给出了一些含密文本生成示例如表 2.2 所示，两个示例嵌入的位置都为第四个单词与第十个单词，嵌入的秘密信息分别为“*come on*”和“*go ahead*”，k 设置为 20，从表中不难观察到，本文所提的 BERT-Gibbs 算法生成的含密文本可读性很高，而 BERT-Only 生成的含密文本则质量很差。

表2.2 不同隐写方法生成的含密文本示例

示例 1	秘密信息	*come on*
	BERT-Only	*” you , **come** . ? , to put **on** the the , , .*
	BERT-Gibbs	*a woman had **come** in , her eyes fixed **on** the closed doorframe .*
示例 2	秘密信息	*go ahead*
	BERT-Only	*” , , **go** now , ” ’ go **ahead** of you ’ . .*
	BERT-Gibbs	*but who would **go** back to the town , **ahead** of the big boss ?*

2.4.3 统计隐蔽性分析

仅仅保障含密文本的流畅度不足以完全保障文本隐写算法的安全性，除此之外还需要保证含密文本的统计安全性，也就是抵抗隐写分析的能力，以抵抗机器的检测。在本文中，文本隐写分析算法使用的是 LS-RNN[75]和 LS-BERT[88]，在进行每次隐写分析实验时，将含密文本与载体文本混合成一个二分类数据集，按 7:3 随机划分为训练集与测试集，同时从训练集中随机选择 10%作为验证集。本章使用隐写分析中最常用的评价指标准确率(Acc)和 F1 值(F1)来评估各个文本隐写算法的抗隐写分析性能。

实验结果如表 2.3 所示，我们可以得出以下结论。首先，BERT-Only 不仅生成的含密文本流畅性很差，其抗隐写分析能力也很差，这是因为 BERT 的预训练方法在单词掩盖率上与本文任务之间有很大的差距，尤其是在 TPT 很小的时候，因此 BERT-Only 生成的含密文本质量很差，如表 2.2 中的示例所示，含密文本甚至不符合自然语言的语法结构，从而导致其很容易被隐写分析检测器检测。而吉布斯采样一方面可以提高含密文本的流畅度，同时又能提高其抗隐写分析能力，使得含密文本的隐蔽性达到了实用水平。

表2.3 不同采样策略下的不同隐写算法的抗隐写分析性能对比

嵌入率	采样策略	隐写方法	LS-RNN		LS-BERT	
			Acc	F1	Acc	F1
1 TPT	Top-1	BERT-Only	0.9948	0.9948	0.9993	0.9993
		BERT-Gibbs	**0.9540**	**0.9530**	**0.9863**	**0.9863**
	Top-10	BERT-Only	0.9853	0.9854	0.9987	0.9987
		BERT-Gibbs	**0.9002**	**0.9025**	**0.9583**	**0.9576**
	Top-20	BERT-Only	0.9848	0.9848	0.9985	0.9985
		BERT-Gibbs	**0.8957**	**0.8938**	**0.9475**	**0.9476**
2 TPT	Top-1	BERT-Only	0.9785	0.9784	0.9978	0.9978
		BERT-Gibbs	**0.8995**	**0.8985**	**0.9537**	**0.9537**
	Top-10	BERT-Only	0.9508	0.9507	0.9953	0.9953
		BERT-Gibbs	**0.8325**	**0.8317**	**0.9153**	**0.9167**
	Top-20	BERT-Only	0.9485	0.9487	0.9957	0.9957
		BERT-Gibbs	**0.8062**	**0.8013**	**0.9140**	**0.9144**
3 TPT	Top-1	BERT-Only	0.9440	0.9427	0.9905	0.9905
		BERT-Gibbs	**0.8287**	**0.8248**	**0.9057**	**0.9078**
	Top-10	BERT-Only	0.9055	0.9039	0.9853	0.9854
		BERT-Gibbs	**0.7890**	**0.7887**	**0.8683**	**0.8665**
	Top-20	BERT-Only	0.8852	0.8848	0.9847	0.9846
		BERT-Gibbs	**0.7548**	**0.7561**	**0.8642**	**0.8603**
4 TPT	Top-1	BERT-Only	0.8837	0.8806	0.9727	0.9724
		BERT-Gibbs	**0.7877**	**0.7921**	**0.8600**	**0.8641**
	Top-10	BERT-Only	0.8392	0.8368	0.9698	0.9696
		BERT-Gibbs	**0.7343**	**0.7252**	**0.8235**	**0.8216**
	Top-20	BERT-Only	0.8232	0.8174	0.9670	0.9667
		BERT-Gibbs	**0.7132**	**0.7044**	**0.8120**	**0.8138**

其次，在嵌入率保持不变时，随着采样词汇表 k 的减少，BERT-Gibbs 生成

的含密文本的流畅性逐渐提高，但是抗隐写分析性能却在变差，这是因为对含密文本流畅度的过度优化会导致其分布与载体文本的分布产生较大差异，使得它们很容易被区分，这与生成式文本隐写算法 VAE-Stega[56]论文中的结论一致。因此在实际场景中，可以通过牺牲一定的含密文本流畅度（只需要保证其与载体文本相当），从而来提升文本的统计安全性，因此在下面的实验中默认设置 $k = 20$。此外，当嵌入率增加时，含密文本的抗隐写分析能力更强。这是因为秘密信息本身是一个自然的文本片段。随着这一部分的比例越来越大，生成的含密文本将更加自然。在极端情况下，秘密信息是整个含密文本，那么含密文本本身就是一个自然文本，但显然此时，隐蔽通信的功能已经丧失。

同时，为了进一步验证本文所提方法的统计隐蔽性，本文将其与最先进的以嵌入比特级秘密信息为目标的生成式文本隐写方法进行了对比。比特级生成式文本隐写方法由训练好的自回归语言模型搭配隐写编码方法构成，为了进行公平的对比，本文使用与 BERT 参数量相当的自回归预训练语言模型 GPT[89]（Generative Pre-trained Transformer），隐写采样方法测试了算数编码[58]（Arithmetic Coding, AC）与自适应分组编码[54]（Adaptive Dynamic Grouping, ADG），在生成时选择每个载体文本的句首词作为含密文本的句首词。两个方案在本文中分别命名为 GPT-AC 与 GPT-ADG，对于 GPT-AC，测试了温度系数 τ 分别为 0.25、0.50、0.75 及 1.00 时的结果，其对应的信息嵌入率分别为 0.65、1.72、2.92 及 4.07 比特每单词（bpw），大致可以对应于本文所提方法中 TPT 分别为 1、2、3 和 4 的情况。对 GPT-ADG，其信息嵌入率是自适应的，本文复现的结果为 3.84 bpw。

实验结果如表 2.4 所示，可以发现本文所提方法相较于比特级生成式文本隐写方法而言取得了更好的抗隐写分析性能。出现这个现象的原因如下，以嵌入比特流为目标的生成式文本隐写方法将秘密信息均匀地隐藏在生成的每个单词之中，因此在生成每个单词时都会对候选词条件概率分布进行改变，导致其生成的含密文本与载体文本的分布差异较大。而本文所提方法仅仅对含密文本的极少量位置进行操纵，其他位置都从 BERT 评估的真实条件概率分布中进行采样，因此生成的含密文本与载体文本分布更为接近，抗隐写分析能力更强。

表2.4 单词级与比特级文本隐写算法的抗隐写分析性能对比

Method	Parameter	LS-RNN		LS-BERT	
		Acc	F1	Acc	F1
GPT-AC	τ=0.25	0.9860	0.9860	0.9958	0.9958
BERT-Gibbs	TPT=1	**0.8957**	**0.8938**	**0.9475**	**0.9476**
GPT-AC	τ=0.5	0.9492	0.9495	0.9782	0.9784
BERT-Gibbs	TPT=2	**0.8062**	**0.8013**	**0.9140**	**0.9144**
GPT-AC	τ=0.75	0.8893	0.8843	0.9378	0.9377
BERT-Gibbs	TPT=3	**0.7548**	**0.7561**	**0.8642**	**0.8603**
GPT-AC	τ=1	0.8040	0.7906	0.8925	0.8884
GPT-ADG	-	0.7913	0.7745	0.8930	0.8883
BERT-Gibbs	TPT=4	**0.7132**	**0.7044**	**0.8120**	**0.8138**

2.5 本章小结

本章提出了一种以嵌入词级秘密信息为目的位置驱动型生成式文本隐写方法，相较于传统以嵌入比特流形式秘密信息为目的方法而言在解码时延以及解码复杂度等方面具有很大的优势，进一步隐藏了接收者的真实身份。具体的，无需重新训练新的模型，结合现有的双向语言预训练模型 BERT 以及吉布斯采样直接生成含密文本。实验结果显示，本文所提方法不仅可以生成流畅可读性高的含密文本，同时具有不错的抗隐写分析能力。

第三章 基于图神经网络的文本隐写分析

3.1 引言

现代网络空间中大量的文本交流行为，以及文本传输的高鲁棒特性，使得文本隐写成为了实现隐蔽通信的重要技术。文本隐写可以有效地应用于维护个人与国防通信安全，但是一旦其被不法分子所利用，则会对网络空间安全造成严重的威胁。尤其是随着近些年来自然语言处理技术的飞速发展，生成式文本隐写利用大规模语料库以及具有强大拟合能力的神经网络，可以生成高隐蔽性的含密文本。仅仅依靠人的感官已区分不了其生成的含密文本与自然文本的差异，除此之外，依赖于手工提取文本特征的传统文本隐写分析方法由于其低效以及不通用，也不足以应对愈来愈强大的生成式文本隐写算法。

为了应对强大的生成式文本隐写技术对于网络空间安全造成的威胁，研究者们开始研究基于神经网络的端到端的文本隐写分析技术。具体的，研究者们将文本视为一维的序列信号，首先通过词嵌入技术将每个单词映射成为一个多维稠密向量，从而将整个文本表示为一个向量序列，然后通过全连接神经网络[71](Fully Connected Neural Network, FCN)、CNN[72]和 RNN[75]等神经网络进行特征提取，最终将提取的特征向量输入到一个 SoftMax 分类器中实现分类。该类方法都将文本建模成序列进行处理，并且主要侧重于从文本本身连续单词组合中提取特征。然而自然语言具有复杂的句法结构、丰富的依存关系及单词共现信息，单词之间的关联性也不仅仅局限于相邻的单词，仅仅用序列的方式无法有效的进行建模。除此之外，以往的方法更多的侧重于提取文本本身的特征，无法很好的利用不同文本之间的全局信息。

近年来，一个新的研究领域即图神经网络引起了巨大的关注。图神经网络非常适用于具有丰富关系结构的任务，并且能够有效地利用全局信息。因此，本文提出了一种基于图神经网络的生成式文本隐写检测算法。在该算法中，我们首先为每一段文本构建一个有向带权图，图中的节点即为单词，图中有向边的构建方法为以文本中每个单词对应的节点为起点，以起点单词相邻特定窗口大小内的单

词所对应的节点为终点形成有向边，从而显示的建模长距离单词之间的依存关系。最后利用图神经网络提取图结构的特征作为文本特征，输入到一个 SoftMax 分类器中实现分类。节点的表示以及边的权重大小是全局共享的，并在训练过程中不断更新。在训练过程中，每个单词可以利用从附近单词中收集到的信息，以及自己的原始信息更新自己的表示，这可以有效解决一词多义的问题。同时，本文设计了一个全局共享的边权重矩阵来记录单词之间的关联强度，这样每个文本都可以利用该全局信息来获得更好的自我表达。实验结果显示，相较于传统方法，本文所提方法在应对多种不同的生成式文本隐写方法时都取得了更好的检测效果。

3.2 相关技术简介

3.2.1 图论基础

图是一种具有强大建模能力的数据类型，能建模现实中很多实际场景，如社交网络关系和蛋白质结构等。例如，在建模社交网络关系时，可以将用户视为节点，用户与用户之间的关系视为边。形式化地描述，一个图$\boldsymbol{G}$由组成该图的节点集合与边的集合共同表达，写作$\boldsymbol{G}=\{\boldsymbol{V},\boldsymbol{E}\}$，这里$\boldsymbol{V}=\{v_1,v_2,...,v_n\}$表示节点数量为$n$的节点集合，$\boldsymbol{E}$表示边集合。令$v_i \in \boldsymbol{V}$表示图中的一个节点，$\boldsymbol{e}_{ij}=(v_i,v_j)\in \boldsymbol{E}$表示图中一条由节点$v_i$指向$v_j$的边。

如果图中的边全部为有向边，则称该图为有向图，如果图中的边全部为无向边，则称该图为无向图。对于无向图而言，$\boldsymbol{e}_{ij}$与$\boldsymbol{e}_{ji}$指代的是同一条边，而在有向图中则具有不同的含义。同时根据图中边有没有权重，又可以分为带权图和无权图。无权图中一般用 0 代表边不存在，1 代表边存在，带权图中边的权重则可以为任意实数。为了便于存储和计算，可以通过邻接矩阵的方式来存储图中边的信息。例如无向图的邻接矩阵可以表示为$\boldsymbol{A}\in\{0,1\}^{n\times n}$，其中如果$\boldsymbol{e}_{i,j}\in\boldsymbol{E}$那么$\boldsymbol{A}_{i,j}=1$，如果$\boldsymbol{e}_{i,j}\notin\boldsymbol{E}$则$\boldsymbol{A}_{i,j}=0$。

3.2.2 图卷积神经网络

卷积神经网络是一种常用且高效的特征提取网络，在计算机视觉与自然语言处理等领域都取得了惊人的成绩。同样的，在图神经网络领域[90]，研究者们注意到了这一点并利用卷积的思想创造了图卷积神经网络(Graph Convolutional Neural Network, GCN)，用于提取图结构的特征。图卷积神经网络主要可以分为基于空域[91-97]和基于谱域[98-100]这两大类。基于谱域的方法通过引入图信号处理中的滤波器，将图卷积操作视为从图信号中去除噪声。基于空域的方法则是从图中节点的空间关系出发，利用消息传播机制处理图信号。近年来，空域的方法由于其简单、高效及通用的特性，逐渐在图卷积神经网络研究中占据了主流。因此本节主要介绍基于空域的图卷积神经网络，后文所提的文本隐写分析算法利用的图卷积神经网络也属于这一类别。

消息传播机制神经网络[94](Message Passing Neural Network, MPNN)概述了空域图卷积神经网络的一个通用框架。其将图卷积视为一个消息传递过程，每个节点的信息可以通过边传递给它的邻居节点，经过多轮的传递，节点的信息可以传递给距离更远的节点。消息传递(也就是空域图卷积)可以描述为：

$$\boldsymbol{h}_{v_i}^{k} = U_k(\boldsymbol{h}_{v_i}^{k-1}, \sum_{u \in \boldsymbol{N}(v_i)} M_k(\boldsymbol{h}_{v_i}^{k-1}, \boldsymbol{h}_{u}^{k-1}, \boldsymbol{e}_{uv_i})) \tag{3.1}$$

其中 $\boldsymbol{h}_{v_i}^{k}$ 表示第 k 轮传递后节点 $\boldsymbol{v_i}$ 的向量表示，$\boldsymbol{h}_{v_i}^{0}$ 表示节点 $\boldsymbol{v_i}$ 的初始向量表示，$\boldsymbol{N}(v_i)$ 表示节点 $\boldsymbol{v_i}$ 的邻域节点集合。$M_k(\cdot)$，$U_k(\cdot)$ 表示包含待训练参数的函数，其分别定义了各节点如何收集邻域节点传递过来的信息及各节点如何通过收集到的信息以及其本身的信息更新自身表示。

$$\boldsymbol{h}_G = R(\boldsymbol{h}_v^k \,|\, v \in \boldsymbol{V}) \tag{3.2}$$

迭代更新后的节点向量表示可以用于节点级别的任务如节点分类。如式(3.2)，利用函数 $R(\cdot)$ 聚合所有节点的特征向量则可以得到图级别的特征表示，从而用于后续图级别的任务如图分类。

现有的空域图卷积网络[93, 95-97]大都属于以上描述的这个框架，只是在函数 $M_k(\cdot)$，$U_k(\cdot)$，$R(\cdot)$ 的定义方式上有所区别。

3.3 基于图神经网络的文本隐写分析

如图 3.1 所示，本文所提出的基于图神经网络的文本隐写分析方法主要可以分为两个阶段，即训练阶段与测试阶段。在训练阶段，每个文本首先被转换成一个有向带权图，图中的节点表示单词，图中的边表示单词与单词之间的关联强度。节点和边的几何表征分别由一个全局的词向量矩阵与一个全局共享的边矩阵获得，这两个矩阵最初随机初始化，然后参与到训练过程进行迭代更新。构建的文本图会输入到一个图神经网络中进行特征提取得到图特征向量，然后输入到 SoftMax 分类器中得到预测，最终通过交叉熵计算损失，反向传播更新两个全局矩阵与模型参数。训练好的模型参数和全局矩阵被使用于测试阶段，具体的，首先使用两个全局共享矩阵将文本转换为文本图，然后文本图将被输入到训练好的模型中得到预测结果。

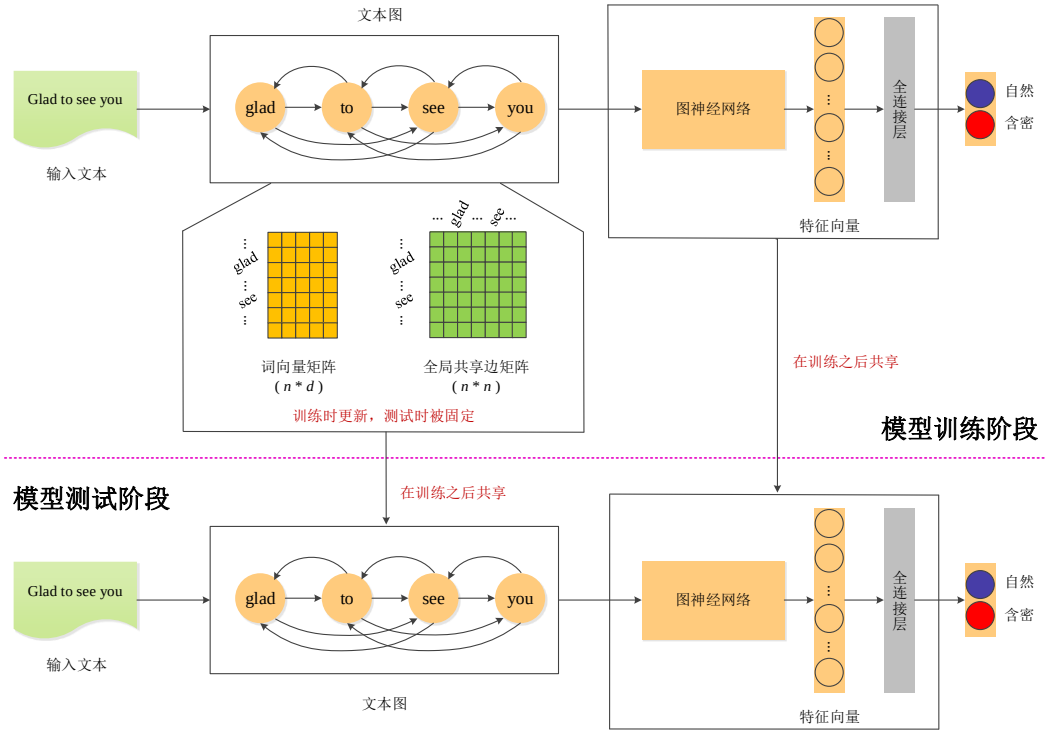

图3.1 基于图神经网络的文本隐写分析框架

3.3.1 文本图构建

对于数据集中的每一段文本，可以表示为 $\boldsymbol{s}=w_1,w_2,...,w_l$，其中 l 为文本单词

总数，w_i 表示此文本的第 i 个单词。本节的目标是为文本 $\boldsymbol{s}$ 构建其文本图 $\boldsymbol{G}=\{\boldsymbol{V},\boldsymbol{E}\}$，其中图中的节点为单词，图中的边为单词与单词的关联信息。

首先，为数据集中所有的单词建立一个词典集合 $\boldsymbol{x}=\{x_1,x_2,...,x_n\}$，$n$ 表示数据集中包含的不重复单词总数。然后，将词典中的每个单词映射为一个 d 维的向量，为此，我们随机初始化一个词向量矩阵 $\boldsymbol{D}=(d_{i,j})_{n\times d}\in R^{n\times d}$，词典中第 i 个单词所对应的词向量即为 $\boldsymbol{D}$ 中的第 i 行。词向量矩阵 $\boldsymbol{D}$ 全局共享，并且在训练过程中不断更新。

对于文本 $\boldsymbol{s}$ 中的第 i 个单词 w_i，假定其在词典 $\boldsymbol{x}$ 中的索引为 j，于是可以用一个独热向量 $\boldsymbol{h}_i\in R^{n\times 1}$ 表示它，n 为词典大小，$\boldsymbol{h}_i$ 除了第 j 位元素为 1，其余元素均为 0。此时 w_i 的词向量表示 $\boldsymbol{b}_i=\boldsymbol{h}_i^T\boldsymbol{D}$，因此 $\boldsymbol{V}$ 可以表示为：

$$\boldsymbol{V}=\{\boldsymbol{b}_1,\boldsymbol{b}_2,...,\boldsymbol{b}_l\}=\{\boldsymbol{h}_1^T\boldsymbol{D},\boldsymbol{h}_2^T\boldsymbol{D},...,\boldsymbol{h}_l^T\boldsymbol{D}\} \tag{3.3}$$

图由节点集合和边集合共同构成，上文已经将文本图中的单词节点映射到了几何空间中从而成功的构建了节点集合 $\boldsymbol{V}$，接下来需要解决的是边集 $\boldsymbol{E}$ 的构建。在本文中，图中边的构建方法为以每个单词对应的节点为起点，以起点单词相邻特定窗口大小 p 内的单词所对应的节点为终点形成有向边，从而显示的建模长距离单词间的依存关系。公式化描述如下：

$$\boldsymbol{E}=\{\boldsymbol{e}_{i,j}=(\boldsymbol{b}_i,\boldsymbol{b}_j)\,|\,1\le i\le l,1\le j\le l,|i-j|\le p,i\ne j\} \tag{3.4}$$

其中 $\boldsymbol{e}_{i,j}$ 表示一条从节点 $\boldsymbol{b}_i$ 指向节点 $\boldsymbol{b}_j$ 的有向边，p 是一个提前设置的超参数，用于控制窗口的大小。注意，由于文本图是有向图，因此 $\boldsymbol{e}_{i,j}$ 不等同于 $\boldsymbol{e}_{j,i}$。每条边 $\boldsymbol{e}_{i,j}\in\boldsymbol{E}$ 的权值来源于一个全局共享的权重矩阵 $\boldsymbol{W}=(w_{i,j})_{n\times n}\in R^{n\times n}$。$\boldsymbol{W}$ 与 $\boldsymbol{D}$ 一样都是随机初始化，在训练过程中进行参数更新。令 $q_{i,j}$ 是边 $\boldsymbol{e}_{i,j}$ 的权重，其值为矩阵 $\boldsymbol{W}$ 中第 i 行第 j 列的元素值，即 $q_{i,j}=\boldsymbol{h}_i^T\boldsymbol{W}\boldsymbol{h}_j$。

图 3.2 中给出了一个构建文本图的例子，给定的文本是“*i hope everything goes well with you*”，此处为“*goes*”设置窗口大小 p 为 2，其余单词 p 设为 1。因此对于“*goes*”，与其距离等于 1 或 2 的单词会与其连边，对于其余单词，与其距离等于 1 的单词会与其连边，最终构建的文本图如图 3.2 所示。后文实验中，所有的文本中所有的单词设置的 p 相等。

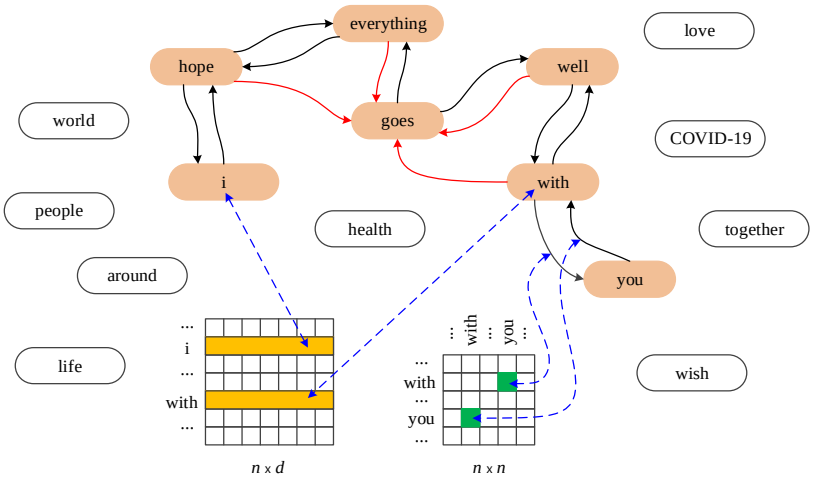

图3.2 文本图构建示例图

通过以上的步骤，最终成功的为每段文本 $\boldsymbol{s}$ 构建了一个文本图 $G(\boldsymbol{V},\boldsymbol{E})$。接下来将其输入到图卷积神经网络中进行图学习。

3.3.2 图学习

本文采用基于空域的图卷积神经网络来进行图学习以提取图特征，本质思想即各节点利用收集到的邻域信息以及节点原始的信息更新节点的自身表示。

对于节点 $\boldsymbol{b}_i$，将其邻域节点向量表示乘上该邻域节点到 $\boldsymbol{b}_i$ 的边的权重作为 $\boldsymbol{b}_i$ 从该邻接节点收集到的信息，然后对收集到的所有邻域节点信息进行池化操作，作为 $\boldsymbol{b}_i$ 收集到的全部邻域信息：

$$\boldsymbol{a}_i = \mathrm{MAXV}\{\boldsymbol{b}_j \times q_{j,i} \,|\, \boldsymbol{b}_j \in \boldsymbol{N}_p(\boldsymbol{b}_i)\} \tag{3.5}$$

其中 $\boldsymbol{N}_p(\boldsymbol{b}_i) = \{\boldsymbol{b}_j \,|\, (\boldsymbol{b}_j,\boldsymbol{b}_i) \in \boldsymbol{E}\}$ 表示节点 $\boldsymbol{b}_i$ 的邻域节点集合，$q_{j,i}$ 为边 $\boldsymbol{e}_{j,i}$ 的权重。MAXV 表示一种最大池化操作，其通过取出待池化的不同向量在每个维度上的最大值从而获得最终结果，例如 $\mathrm{MAXV}\{(1,2,3),(6,5,2),(2,7,1)\} = (6,7,3)$。得到收集的邻域信息 $\boldsymbol{a}_i$ 后，再利用此信息与节点本身信息 $\boldsymbol{b}_i$ 乘上各自的权重并求和，从而得到节点的新表示：

$$\boldsymbol{b}_i \leftarrow \boldsymbol{w}^T \boldsymbol{q}_i \boldsymbol{a}_i + (1 - \boldsymbol{w}^T \boldsymbol{q}_i)\boldsymbol{b}_i \tag{3.6}$$

其中 $\boldsymbol{w} \in R^{2p \times 1}$ 是待训练的参数，$\boldsymbol{q}_i$ 则表示节点 $\boldsymbol{b}_i$ 的邻域节点指向该点的边的权

重向量，即：

$$\boldsymbol{q}_i = (q_{i-p,i}, \ldots, q_{i-1,i}, q_{i+1,i}, \ldots, q_{i+p,i})^T \tag{3.7}$$

$\boldsymbol{q}_i$ 用于计算两部分信息在更新过程中给自所占的权重，需要注意的是，由于节点索引须在区间 $[0,l]$ 内，因此式(3.7)只适用于 $i \in [p+1, l-p]$ 的情形。为了使该式可以适用于文本中所有的单词节点即 $i \in [1,l]$，对于 $i \in [1,l]$，$|\delta| \in [1,p]$，如果 $i+\delta \notin [1,l]$，此时用 0 代替 $q_{i+\delta,i}$。上述操作的物理意义就是，为了方便用一个定长变量表示 $\boldsymbol{q}_i$，用 0 代替不存在的边的权值。

图 3.3 展示了单词“*goes*”如何收集邻域信息与更新自身表示，图 3.3(a)描述了其收集邻域各节点的信息并进行最大池化，图 3.3(b)描述了其利用收集到的邻域信息更新自身表示。

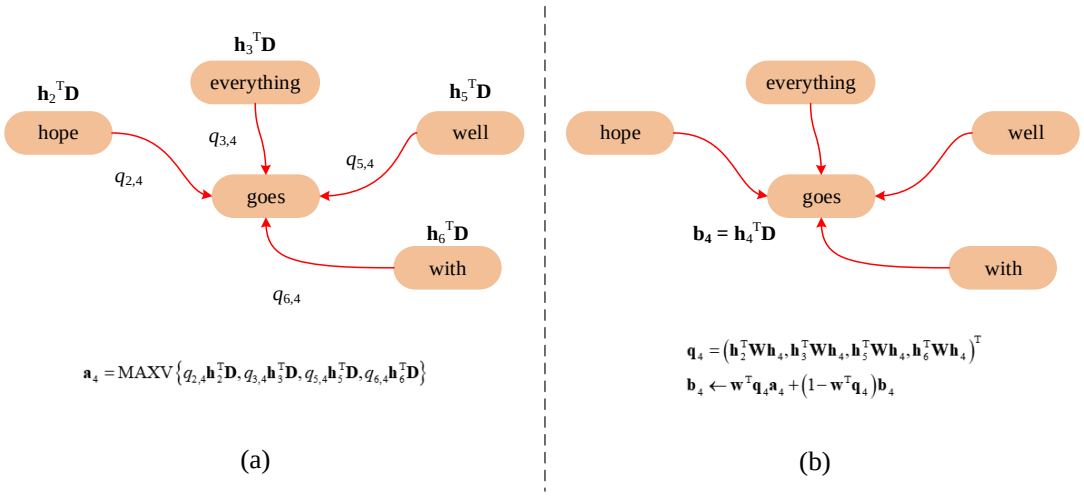

图3.3 邻域信息收集与信息更新示例图

$\boldsymbol{V}$ 中的每个节点都更新完自身的表示之后，我们通过平均池化操作得到文本图的特征向量作为文本的特征向量：

$$\boldsymbol{f} = \sum_{i=1}^{l} \boldsymbol{b}_i^T / l \tag{3.8}$$

其中 l 表示节点总数。然后将该特征向量输入到一个输出维度为 2 的全连接层中，最后通过一个 SoftMax 操作得到概率分布：

$$\boldsymbol{o}_x = (p_0, p_1)^T = \text{SoftMax}(\tanh(\boldsymbol{W}_h \boldsymbol{f} + \boldsymbol{b}_h)) \tag{3.9}$$

其中 $\boldsymbol{W}_h \in R^{2\times d}$ 以及 $\boldsymbol{b}_h \in R^{2\times 1}$ 都是待训练的参数，$\tanh(x) = (e^x - e^{-x})/(e^x + e^{-x})$ 为

激活函数。最后利用交叉熵损失函数计算损失，通过最小化损失学习待学习的全局共享矩阵与模型参数：

$$L = -g_0 \log p_0 - g_1 \log p_1 \tag{3.10}$$

其中 $\boldsymbol{g} = (g_0, g_1)^T$ 表示文本 $\boldsymbol{s}$ 标签的独热向量表示，以上步骤同理可以推广到批量化操作。

3.4 实验与分析

在本节中，首先描述了待检测文本隐写数据集的构建方法及隐写分析检测算法基线等实验设置，然后展示了本文所提算法与各基线算法的检测性能对比，最后测试了不同超参数设置对于本文所提方法性能的影响。

3.4.1 实验设置

本节检测的生成式文本隐写方法包括 Bins[53]、FLC[16]和 VLC[16]，这些方法均是先在语料库上训练一个 LSTM 语言模型，然后利用该语言模型生成含密文本，不同点在于生成含密文本时使用了不同的编码策略。Bins 将词表平均分成若干个桶，然后为每个桶编码，生成含密文本时选择秘密信息对应的桶中概率最大的单词。FLC 与 VLC 则是根据条件概率选择若干个概率最大的词进行编码，前者使用与概率无关的定长编码，后者则使用与概率相关的霍夫曼编码，然后在生成含密文本时选择秘密信息对应的单词。

本文使用了两个不同的数据集 Twitter[101] 与 IMDB[102]，分别用其训练得到两个 LSTM 语言模型。IMDB 与 Twitter 数据集的具体细节如表 3.1 所示，其文本的平均长度分别约为 20 和 10。对于每个语言模型，使用上述的编码方式在不同的信息嵌入率设置下各生成 10000 条含密文本作为正样本，然后从其训练语料库中随机选择 10000 条载体文本作为负样本，从而构建出隐写分析数据集。每次隐写分析实验时，分别随机选择数据集的 70%作为训练集，30%作为测试集，另外从训练集中随机选择 10%作为验证集。

表3.1 训练数据集的细节

数据集	Twitter[101]	IMDB[102]
句子平均长度	9.68	19.94
包含句子总数	2,639,290	1,283,813
包含单词总数	2,551,044	25,601,794

本文对比的隐写分析方法包括 LS_FCN[71]、LS_CNN[72]和 LS_RNN[75]。它们都先将文本映射为词向量序列，然后利用神经网络提取文本特征，最终输入到一个 SoftMax 分类器中实现含密文本的检测。不同点在于提取特征的方式不同，LS_FCN 使用一个全连接层提取文本特征，LS_CNN 使用不同尺寸的一维卷积核提取文本特征，LS_RNN 则利用循环神经网络提取文本特征。

在本文提出的方法 LS_GNN 中，绝大多数参数在模型训练过程中通过迭代优化得到，少数超参数需要提前手动设置。如词向量的维度为 200，并使用 Glove[103]词向量进行初始化，全局共享的边权重矩阵通过统计隐写分析数据集中单词之间的 PMI（Pointwise Mutual Information）进行初始化。本文使用 Adam[104]优化器优化模型参数，学习率设置为 0.001，批量大小为 128。最后的全连接层之后使用了保持率为 0.5 的 Dropout[105]函数。此外，在每次隐写分析实验中，本文测试了在文本图构造过程中设置不同的窗口大小对于检测性能的影响，即 p 的取值范围为[2,3]，并选取最佳的 p（根据在验证集上评估的检测精度），用于测试以得到测试集检测结果。本文使用隐写分析中最常用的评价指标准确率（Acc）和 F1 值（F1）来评估各个隐写分析算法的性能。

3.4.2 检测性能对比

不同的文本隐写检测算法在检测不同的隐写方法时的结果如表 3.2 和表 3.3 所示，结果分别对应于 Twitter 数据集与 IMDB 数据集。生成式文本隐写算法的嵌入率用比特每单词（bpw）衡量，例如“bpw=2.000”表示含密文本平均每个单词携带 2 比特的秘密信息。从表 3.2 与表 3.3 中可以发现，随着含密文本嵌入率的增加，各种隐写分析方法的检测性能都在降低。这与论文 VAE-Stega[56]中的实验结果一致，这是因为在嵌入率低时，生成的含密文本具有更高的流畅度，量化指标体现在更低的 PPL，而对于含密文本流畅度的过度追求会导致其与载体文本的

分布差距变大，更容易被隐写分析算法检测。

表3.2 不同隐写分析方法应对不同隐写方法时的检测性能对比(Twitter)

隐写分析方法	隐写术	Bins			FLC			VLC		
	嵌入率	1.000	2.000	3.000	1.000	2.000	3.000	1.000	2.150	3.260
LS_FCN	Acc	0.814	0.783	0.765	0.799	0.766	0.739	0.795	0.769	0.752
	F1	0.816	0.784	0.774	0.791	0.770	0.751	0.790	0.771	0.751
LS_CNN	Acc	0.907	0.895	0.895	0.895	0.894	0.894	0.894	0.881	0.884
	F1	0.908	0.898	0.896	0.897	0.896	0.894	0.896	0.883	0.886
LS_RNN	Acc	0.910	0.900	0.900	0.900	0.882	**0.895**	0.904	0.881	**0.888**
	F1	0.912	**0.903**	**0.900**	0.902	0.885	**0.899**	0.905	**0.888**	**0.892**
LS_GNN	Acc	**0.913**	**0.901**	**0.901**	**0.902**	**0.897**	0.889	**0.907**	**0.891**	0.887
	F1	**0.913**	0.902	**0.900**	**0.904**	**0.898**	0.886	**0.906**	0.887	0.886

表3.3 不同隐写分析方法应对不同隐写方法时的检测性能对比(IMDB)

隐写分析方法	隐写术	Bins			FLC			VLC		
	嵌入率	1.000	2.000	3.000	1.000	2.000	3.000	1.000	2.215	3.147
LS_FCN	Acc	0.897	0.858	0.805	0.885	0.833	0.782	0.878	0.836	0.802
	F1	0.894	0.857	0.803	0.883	0.832	0.785	0.874	0.835	0.796
LS_CNN	Acc	**0.961**	0.939	**0.926**	0.949	**0.936**	0.918	0.949	0.931	**0.923**
	F1	**0.962**	0.940	**0.928**	0.950	**0.937**	**0.921**	0.950	0.933	**0.924**
LS_RNN	Acc	0.954	0.941	0.915	0.953	0.923	0.903	0.947	0.933	0.917
	F1	0.954	0.942	0.918	**0.954**	0.927	0.908	0.948	0.935	0.920
LS_GNN	Acc	0.954	**0.943**	0.916	**0.954**	**0.936**	**0.921**	**0.952**	**0.939**	**0.923**
	F1	0.954	**0.943**	0.915	**0.954**	0.936	0.920	**0.952**	**0.939**	0.922

与其他的文本隐写分析方法相比，本文所提方法在绝大多数情况下取得了最佳的性能，除了这些情况之外，本文方法与最佳的性能也十分相近。这表明，与其余将文本建模成序列的方法相比，本文方法将文本建模成图具有更强的表征能力，因为在图结构下，可以建模不相邻单词之间的显示关联，此外每个单词可以收集邻域的信息实现更好的自我表征。同时，通过学习和利用全局信息，每个文本可以进一步从其他文本中收集信息，这使得所提出的模型可以捕获更多的文本鉴别特征，实现更好的检测效果。

另一个有趣的现象是，在同样的文本隐写算法以及同样的嵌入率设置下，IMDB 数据集相较于 Twitter 数据集而言生成的含密文本更容易被检测。这可能是由于 Twitter 数据集的平均句子长度更短，如上文所述，平均长度约为 10，而 IMDB 中的句子平均长度为 20，这导致在 Twitter 数据集上训练的语言模型倾向

于生成更短的含密文本。因此在同等嵌入率下其生成的每个含密文本所携带的机密信息总量更少，更难被检测。

3.4.3 改变超参数 p 对于检测性能的影响

除此之外，本文还测试了在建模文本图时使用不同的窗口大小 p 对于文本隐写分析检测准确率的影响。图 3.4 显示了本文所提方法在两个数据集上针对不同隐写方法不同嵌入率的平均检测准确率随着 p 增大的变化情况，显然随着 p 的增大，准确率会先增加后降低，最佳值为 2 或 3。这是因为当 p 太小时，则无法建模长距离单词之间的显示依赖，而当窗口 p 过大时，文本图会更接近于全连接图，忽略了局部特征，不利于学习，例如当文本图成为一个全连接图时，将失去文本中单词的先后时序信息。因此，在实际使用中，可以利用验证集在小范围中搜索最佳的 p，例如上文所使用的 p 的取值范围为[2,3]，从而既能保持高检测性能又能降低计算成本。

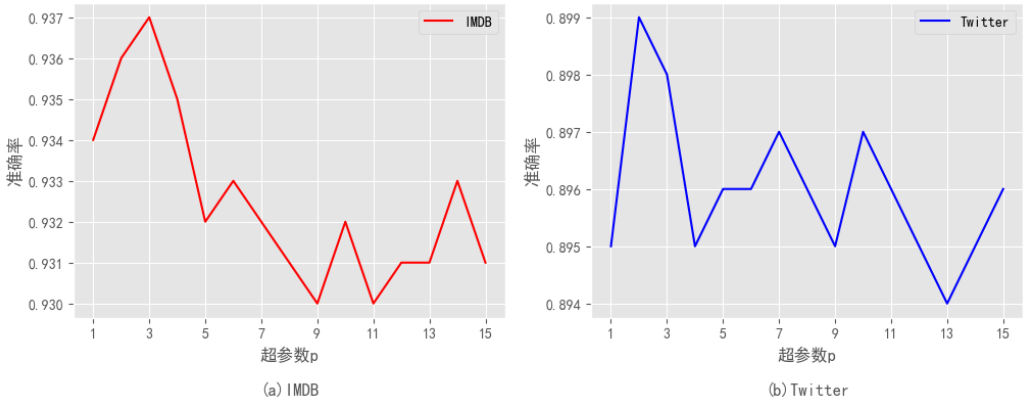

图3.4 检测准确率随 p 值增大的变化趋势

3.5 本章小结

本章提出了一种基于图神经网络的文本隐写分析算法。首先，与之前将文本建模为序列的方法不同，本章所提方法将文本建模为可以显式表征长距离单词之间依存关系的图结构。其次，在模型训练过程中，文本中的每个词都可以从邻域单词中收集信息来更新自己的表示，从而有效地利用上下文信息。最后，全局共

享矩阵的设置使得该算法能够更好地利用全局信息，即文本之间可以相互收集有用的信息。实验结果表明，所提方法在应对多种生成式文本隐写算法时，取得了较传统方法而言更好的检测性能。

第四章 基于预训练语言模型的文本隐写分析

4.1 引言

基于大数据与神经网络的生成式文本隐写算法能够在维持高嵌入率的同时生成隐蔽性高的含密文本，为充斥着文本信息的现代网络空间带来了极大的安全威胁。为此，研究者们以“以子之矛，攻子之盾”的思想，利用深度神经网络技术学习含密文本与自然文本的分布差异，从而检测含密文本，取得了不错的成效。从 2018 年开始，研究者们将文本视为一维序列，相继引入了 FCN[71]、CNN[72]及 RNN[75]等神经网络架构提取文本特征实现隐写分析，在本文第三章中，更进一步的引入了 GNN，并将文本建模成图结构，实现了更好的检测性能。但是总的来说，现有的方法均将文本隐写分析视为一个常规的文本分类任务，寄希望于通过寻求更好的网络架构或特征提取器来实现更好的性能。在这种趋势下，使得文本隐写分析领域的发展十分依赖于文本特征提取与表征技术的发展。本章节提出了一个新的观点，即不能仅仅将文本隐写分析视为一个常规的文本分类任务，实际上其有其特殊之处，挖掘其特殊的领域知识是推动文本隐写分析发展的另一个角度与契机。

生成式文本隐写不同于传统的修改式方法通过修改载体文本以嵌入秘密信息，而是通过训练好的语言模型实现信息嵌入，换而言之，含密文本是通过语言模型按一定的模式生成的，例如使用定长编码生成含密文本时可以视为语言模型使用类似 Top k 采样的方式生成文本，这不可避免的导致生成的含密文本会暴露统计特征给该语言模型。后续通过实验进一步证实了这一点，首先选取同等数量的载体文本与不同嵌入率下的含密文本，然后使用语言模型计算其各自的文本级别困惑度与位置级别困惑度的分布情况，结果显示语言模型可以明显的建模出载体文本与含密文本的分布差异，甚至还能观察到随着嵌入率的提高，含密文本分布的变化趋势。实际场景中，隐写分析方通常不具备获取发送方所使用的语言模型的能力，所能获取的只是训练该语言模型的部分载体文本，为此，我们利用部分文本训练一个语言模型的替代模型，发现该替代模型依然具有建模含密文本与

载体文本分布差异的能力。

由上述分析可知，语言模型不仅仅是一个高超的文本隐写者，也天然具有检测含密文本的能力，迁移语言模型的该种能力将有助于提高文本隐写分析性能。因此，本文提出了一个基于微调预训练语言模型[43, 88, 106]的高效文本隐写分析框架，首先在收集到的载体文本上预训练一个语言模型，然后通过适配的方式（如添加一个全连接层）微调该语言模型进行文本隐写分析，这样预训练语言模型的领域知识可以迁移至文本隐写分析分类器上，实现更好的检测性能。在本文中，提出了两种有效的预训练方法，一种是基于 RNN 的语言模型[44]，另一种是基于 RNN 的序列自编码器[106, 107]。后续的隐写分析实验结果表明，相较于随机初始化的 RNN 文本隐写分析方法，本文所提两种方案都取得了不同程度的检测性能提升，同时显著提升了隐写分析训练时损失的收敛效率。此外，实验结果显示，随着预训练数据集的增大，检测性能进一步提升，这为应对含密文本收集难度远高于自然文本的现实场景提供了解决方案。

4.2 基于预训练语言模型的文本隐写分析

4.2.1 研究动机

生成式文本隐写利用训练好的语言模型进行信息嵌入，换而言之，含密文本是通过语言模型按一定的模式生成的，这不可避免的导致生成的含密文本会暴露统计特征给该语言模型。为了进一步论证这一点，我们设置了如下实验。首先在 MOVIE[102]数据集上训练一个基于 LSTM 的语言模型（实现方式与 RNN-Stega[16] 文中方法一致），然后用该语言模型搭配定长编码（FLC）在不同嵌入率（bpw 为 1、2 和 3）设置下各生成 1000 段含密文本，同时从 MOVIE 数据集中随机选取 1000 段载体文本。假设 $\boldsymbol{s}=w_1,w_2,...,w_l$ 是一段文本，l 是该文本的长度，w_i 表示文本的第 i 个单词，用语言模型计算文本困惑度（Perplexity, PPL）的方式如下：

$$\mathrm{PPL}(\boldsymbol{s})=2^{-\frac{1}{l}\times\log_2 p(w_1,w_2,...,w_l)} \tag{4.1}$$

同时对于文本的每个位置 $i\in[1,l]$，可以通过以下方式计算其位置困惑度（Position-wise Perplexity, PWP）：

$$\mathrm{PWP}(i) = 2^{-\log_2 p(w_i|w_1,w_2,\ldots,w_{i-1})} \tag{4.2}$$

对于每种类型的 1000 段文本，利用训练好的语言模型按式(4.2)计算每段文本的位置困惑度向量，然后计算其在每个位置上的平均值作为最终结果，绘制成折线图如图 4.1 所示。同时，计算每段文本的文本困惑度，可以得到不同类型文本的文本困惑度分布情况如图 4.2 所示。

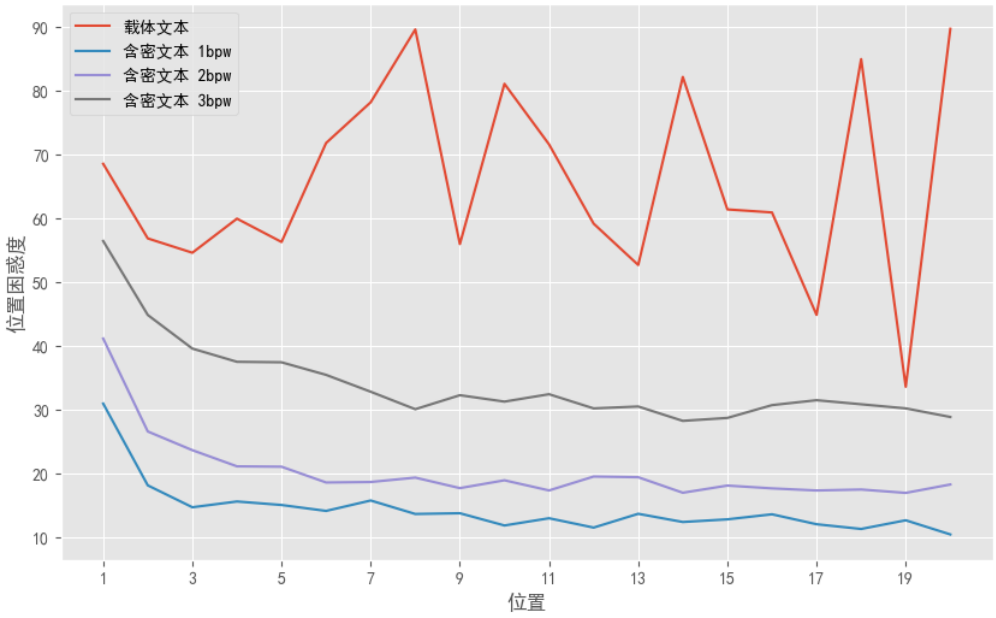

图4.1 载体文本与含密文本的位置困惑度分布

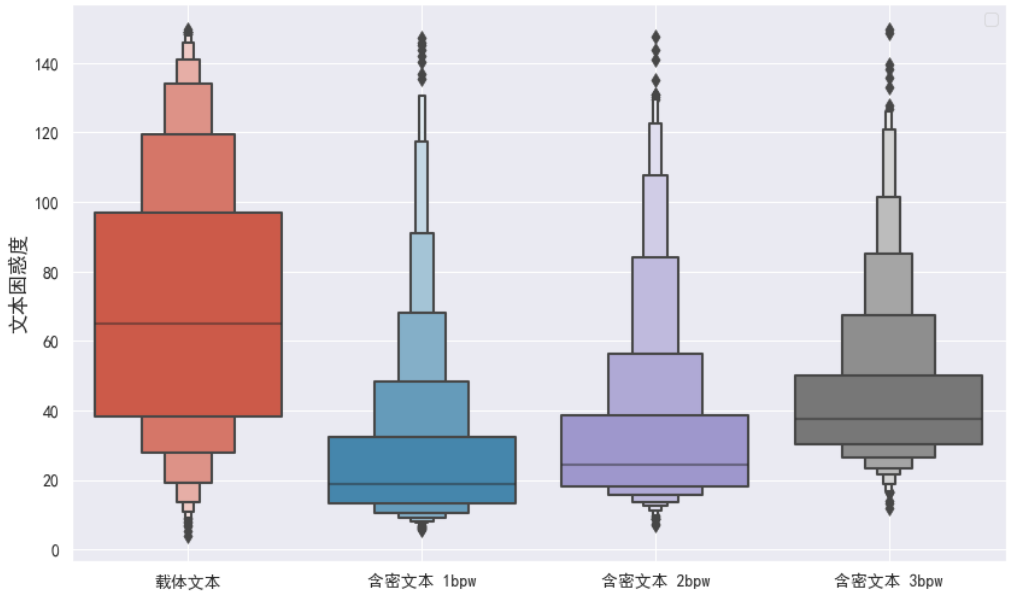

图4.2 载体文本与含密文本的文本困惑度分布

如图 4.1，就位置困惑度分布而言，含密文本和载体文本之间存在明显的差异。具体的，含密文本在各个位置上的位置困惑度较载体文本而言普遍偏低，而且方差更小，更平滑。如图 4.2 所示，含密文本与载体文本在文本困惑度分布上同样存在明显的差异，载体文本的困惑度较为平均的分散在一个合理的大区间中，而含密文本的困惑度则集中于极端的小区间中。

互联网中的自然文本是由不同背景和不同身份的人书写的，不同的人会有不同的书写风格与特色，因此其位置困惑度分布方差较大，文本困惑度分布更为分散。而现有生成式文本隐写算法由于其对文本流畅度的过度优化，导致其生成的含密文本与自然文本存在较大的分布差异。语言模型天然可以建模出这种差异，如图 4.1 和 4.2 所示，甚至于只要设置一个合适的统计特征值阈值，就能将含密文本和自然文本中的一部分样本区分开来，实现含密文本的检测。

通过上述实验我们发现，语言模型不仅仅一个高超的隐写者，同时其本身也是一个隐写分析专家。但是在主流的隐写分析框架下，隐写分析方无法获取隐写方所使用的语言模型，隐写分析方能够获取的只是训练该语言模型的一部分载体文本。在这种更严苛的场景下，我们进行了进一步的实验，即从语言模型的训练语料库中随机选取 10000 段载体文本用于训练一个语言模型的替代模型(基于同样的模型架构)，然后用其计算不同类型文本的文本困惑度分布。此外，我们还使用了一个随机初始化的相同模型架构的语言模型计算不同类型文本的文本困惑度分布以进行对比。

实验结果如图 4.3 所示，对于随机初始化的语言模型，其建模不出含密文本和载体文本之间的文本困惑度分布差异，而对于仅仅在少量载体文本上预训练的替代语言模型，却能明显的建模出载体文本与含密文本的困惑度分布差异。这说明通过预训练语言模型的方式的确可以学习到用于区分载体文本与含密文本的先验知识。这促使我们提出了一个基于微调预训练语言模型的文本隐写分析方法，其首先通过在载体文本上预训练语言模型的方式学习先验知识，然后以微调的方式将先验知识迁移至隐写分析分类器中，以实现更好的隐写分析性能。

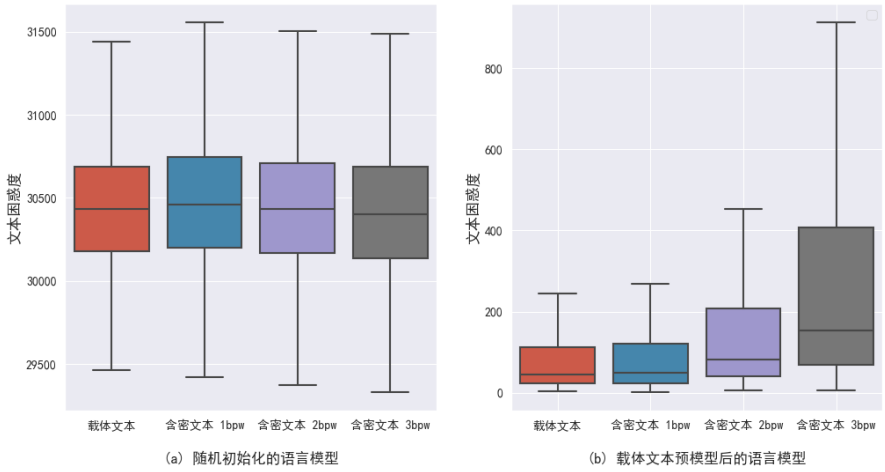

图4.3 不同语言模型计算不同类型文本的文本困惑度分布

4.2.2 预训练微调框架

本文提出的基于微调预训练语言模型的文本隐写分析框架如图 4.4 所示，给定一个神经网络模型如 RNN 或 Transformer 等，本文以 LSTM 为例，首先在载体文本上预训练语言模型以学习先验知识，然后用训练好的模型参数作为后续文本隐写分析任务的初始值，最后通过微调的方式实现高效的文本隐写分析。

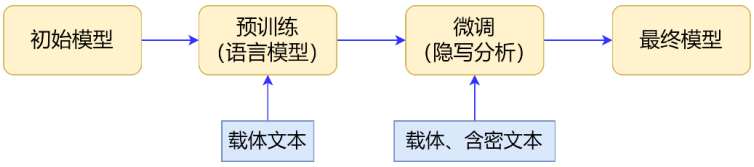

图4.4 基于微调预训练语言模型的文本隐写分析框架

本文提出的预训练语言模型方式有两种，第一种是基于 RNN 的传统自回归语言模型[44](Language Model, LM)，给定文本的上文从而预测文本接下来的部分，如图 4.5 所示，给定“*Thank*”预测“*you*”，给定“*Thank you*”预测“*so*”，直至生成整个文本“*Thank you so much*”，其中“<*eos*>”为文本终止符号。

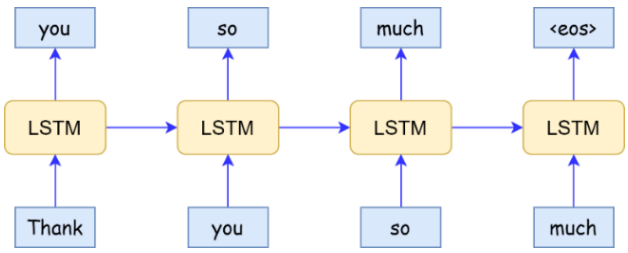

图4.5 语言模型训练示例

对于任意一段文本，可以将其表示为 $\boldsymbol{s}=w_1,w_2,...,w_l$，其中 l 为文本所包含的单词总数量，w_i 表示此文本的第 i 个单词。通过如下方式实现由任意前缀预测下一个单词，即对于任意的 $n\le l$ 建模条件概率分布 $p(w_n|w_1,w_2,...,w_{n-1})$：

$$\begin{cases}\boldsymbol{h}_{n-1}=f_{\text{LSTM}}(\xi(w_1),\xi(w_2),...,\xi(w_{n-1}))\\ p(w_n|w_1,w_2,...,w_{n-1})=\text{SoftMax}(\boldsymbol{W}_h\boldsymbol{h}_{n-1}+\boldsymbol{b}_h)\end{cases} \tag{4.3}$$

其中 $\xi(*)$ 函数为包含待训练参数的词向量映射函数，将单词映射为向量，$\boldsymbol{W}_h$，$\boldsymbol{b}_h$ 为待训练的参数，其目的是将隐向量 $\boldsymbol{h}_{n-1}$ 映射到词表大小的输出空间，然后通过 SoftMax 函数将其转换为概率分布。

神经网络的模型参数，词向量映射函数中待训练的参数，都需要通过训练才能得到。语言模型的训练目标为最小化每句话的概率负对数：

$$\begin{aligned}Loss&=-\log(p(\boldsymbol{s}))=-\log(p(w_1,w_2,...,w_l))\\&=-\log(p(w_1)p(w_2|w_1)...p(w_l|w_1,w_2,...,w_{l-1}))\\&=-\sum_{t=1}^{l}p(w_t|w_1,w_2,...,w_{t-1})\end{aligned} \tag{4.4}$$

预训练语言模型的另一种方式是训练一个序列自编码器[106, 107](Sequence Autoencoder, SAE)，其将文本用编码器(Encoder)编码成一个向量，然后以该向量作为初始隐向量用解码器(Decoder)重构输入文本。如图 4.6 所示，给定编码器输出的隐向量与起始符号“<*sos*>”，逐字预测整个文本。需要注意的是，编码器和解码器使用的是同一个 LSTM 网络，即其参数是共享的。

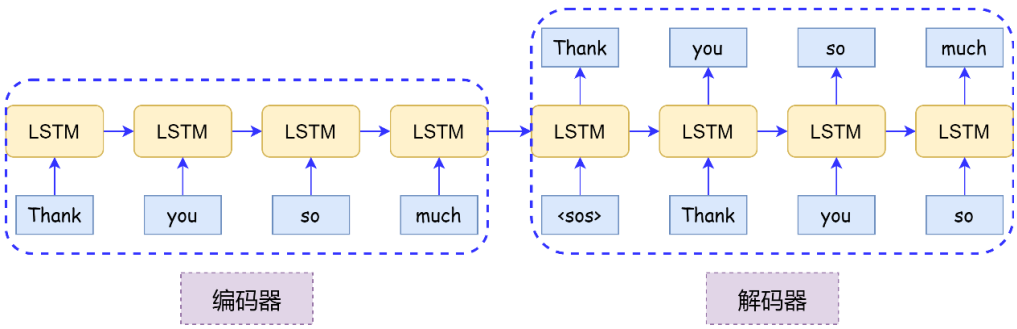

图4.6 序列自编码器训练示例

形式化的描述，首先利用编码器编码得到输入文本的语义向量：

$$\boldsymbol{h}_l=f_{\text{Encoder}}(\xi(w_1),\xi(w_2),...,\xi(w_l)) \tag{4.5}$$

然后用该语义向量重构输入文本，序列自编码器的训练目标为最小化如下损

失函数：

$$\begin{aligned} Loss &= -\log(p(\boldsymbol{s}|\boldsymbol{h}_l)) = -\log(p(w_1, w_2, ..., w_l|\boldsymbol{h}_l)) \\ &= -\log(p(w_1|\boldsymbol{h}_l)p(w_2|\boldsymbol{h}_l, w_1)...p(w_l|\boldsymbol{h}_l, w_1, w_2, ..., w_{l-1})) \\ &= -\sum_{t=1}^{l} p(w_t|\boldsymbol{h}_l, w_1, w_2, ..., w_{t-1}) \end{aligned} \tag{4.6}$$

通过以上两种方式中的任意一种实现预训练后，使用预训练后具有语言模型知识的模型参数作为隐写分析分类器的初始值，然后通过微调的方式实现高效的文本隐写分析。如图 4.7 所示，首先将训练文本输入到预训练好的词向量函数以及 LSTM 网络中，得到最后时刻的隐向量，然后用全连接层将其投射到大小为 2 的输出空间，最后用 SoftMax 函数将其转化为概率分布，训练过程的目标即为最小化预测分布与真实分布之间的交叉熵。

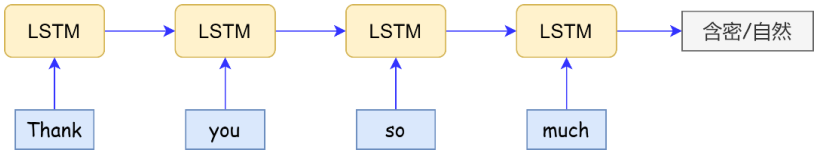

图4.7 微调示例

综上所述，本文所提框架首先在载体文本上预训练语言模型，使得神经网络具备先验知识，然后在载体文本与含密文本上微调分类任务，得到最终模型，最终用该模型检测未知文本中是否含有秘密信息，实现文本隐写分析。

4.3 实验与分析

在此部分，首先描述了待检测数据集的构建方法及隐写分析检测算法基线等实验设置，然后展示了本文所提算法与各基线算法的检测性能对比以及训练效率对比，最后测试了增加预训练数据量对于本文所提方法性能的影响。

4.3.1 实验设置

本文检测的隐写算法包括 Bins[53]、FLC[16]与 VLC[16]三种，三种都是生成式文本隐写算法，只是采用了不同的编码方法，分别是桶编码、定长编码以及霍夫曼编码。实验过程中采用的语料库为 Twitter[101]和 MOVIE[102]两个，首先分别使用这两个语料库训练语言模型，然后按不同的嵌入率以及不同的编码方式各生成

10000 段含密文本，同时从训练语料库中随机选择 10000 段作为载体文本。在每次隐写分析实验中，分别选择数据集中的 70%作为训练集，30%作为测试集，另外从训练集中选择 10%作为验证集。本文对比的文本隐写分析方法包括 LS_FCN[71]、LS_CNN[72]、LS_RNN[75]和上章所提出的 LS_GNN。分类任务中常用的评价指标准确率(Acc)与 F1 值(F1)作为本节隐写分析任务的性能评价指标。

本文所提的两种方案 LM_RNN，AE_RNN 所采用的超参数设置是相同的。分词方法与 BERT[43]论文中提出的分词方法一致，然后将单词先通过词嵌入层映射为维度为 128 的向量，嵌入层后所接的 Dropout[105]的保持率为 0.5，LSTM 的层数设置为 2，隐向量维度为 256，所采用的优化器为 Adam[104]，学习率设置为 0.001。预训练时的批量大小均为 128，训练轮数为 50。预训练后，直接采用预训练好的模型参数作为分类器初始值进行隐写分析，因此超参数都不变，只是最后添加一个全连接层将最后时刻的隐向量映射到维度为 2 的输出空间。

4.3.2 检测性能对比

不同的文本隐写分析算法在检测不同的文本隐写方法时的结果如表 4.1 和表 4.2 所示，结果分别对应于 Twitter 数据集与 IMDB 数据集。

表4.1 不同隐写分析方法应对不同隐写方法时的检测性能对比(Twitter)

隐写分析方法	隐写术	Bins			FLC			VLC		
	嵌入率	1.000	2.000	3.000	1.000	2.000	3.000	1.000	2.150	3.260
LS_FCN	Acc	0.814	0.783	0.765	0.799	0.766	0.739	0.795	0.769	0.752
	F1	0.816	0.784	0.774	0.791	0.770	0.751	0.790	0.771	0.751
LS_CNN	Acc	0.907	0.895	0.895	0.895	0.894	0.894	0.894	0.881	0.884
	F1	0.908	0.898	0.896	0.897	0.896	0.894	0.896	0.883	0.886
LS_GNN	Acc	0.913	0.901	0.901	0.902	0.897	0.889	0.907	0.891	0.887
	F1	0.913	0.902	0.900	0.904	0.898	0.886	0.906	0.887	0.886
LS_RNN	Acc	0.910	0.900	0.900	0.900	0.882	0.895	0.904	0.881	0.888
	F1	0.912	0.903	0.900	0.902	0.885	0.899	0.905	0.888	0.892
LM_RNN	Acc	0.908	0.902	0.899	0.909	0.900	**0.901**	**0.908**	0.898	0.896
	F1	0.907	0.902	0.898	0.907	0.900	**0.903**	**0.907**	0.897	0.895
AE_RNN	Acc	**0.918**	**0.906**	**0.908**	**0.911**	**0.904**	0.897	0.904	**0.907**	**0.903**
	F1	**0.917**	**0.907**	**0.909**	**0.912**	**0.904**	0.900	0.906	**0.907**	**0.902**

表4.2 不同隐写分析方法应对不同隐写方法时的检测性能对比(IMDB)

隐写分析方法	隐写术	Bins			FLC			VLC		
	嵌入率	1.000	2.000	3.000	1.000	2.000	3.000	1.000	2.215	3.147
LS_FCN	Acc	0.897	0.858	0.805	0.885	0.833	0.782	0.878	0.836	0.802
	F1	0.894	0.857	0.803	0.883	0.832	0.785	0.874	0.835	0.796
LS_CNN	Acc	0.961	0.939	0.926	0.949	0.936	0.918	0.949	0.931	0.923
	F1	0.962	0.940	0.928	0.950	0.937	0.921	0.950	0.933	0.924
LS_GNN	Acc	0.954	0.943	0.916	0.954	0.936	0.921	0.952	0.939	0.923
	F1	0.954	0.943	0.915	0.954	0.936	0.920	0.952	0.939	0.922
LS_RNN	Acc	0.954	0.941	0.915	0.953	0.923	0.903	0.947	0.933	0.917
	F1	0.954	0.942	0.918	0.954	0.927	0.908	0.948	0.935	0.920
LM-RNN	Acc	**0.963**	**0.952**	0.934	0.959	0.945	0.930	0.961	0.943	**0.940**
	F1	**0.964**	**0.952**	0.934	0.959	0.946	0.929	0.961	0.943	**0.940**
AE-RNN	Acc	0.963	0.949	**0.935**	**0.962**	**0.950**	**0.932**	**0.964**	**0.948**	0.937
	F1	0.963	0.949	**0.936**	**0.962**	**0.950**	**0.933**	**0.965**	**0.948**	0.937

由实验结果可以得到以下结论：首先，本文提出的 LM_RNN，AE_RNN 取得了最佳的检测结果，验证了本文所提框架的可行性和优越性。其次，本文所提出的预训练微调方法相较于随机初始化的 LS_RNN 模型而言，在检测性能上有明显的提升，这说明迁移语言模型的先验知识确实可以提高后续分类器的检测性能。此外，AE_RNN 相较于 LM_RNN 而言检测性能提升更为明显，这是因为传统语言模型的训练目的是根据文本前缀预测下一个单词，而序列自编码器不仅仅能做到这一点，而且可以编码整个句子的信息，具有更强的文本建模能力。

4.3.3 训练效率对比

本文也比较了随机初始化 LS_RNN 与预训练的 RNN 在隐写分析训练过程中的损失收敛效率，结果如图 4.8 所示，其中的纵坐标代表的是隐写分析方法在数据集上进行不同嵌入率与不同编码方式隐写分析实验的平均损失。从图中可以发现，预训练会大大提高隐写分析训练效率，且 AE_RNN 相较于 LM_RNN 而言效率提升更为明显。实际上，由于在确定的数据集上进行不同编码方式不同嵌入率的隐写分析时使用的载体文本是一样的，因此，只需要预训练一次，就可以同时提高针对不同编码方式不同嵌入率的文本隐写分析实验的训练效率，从而节省大量计算资源。

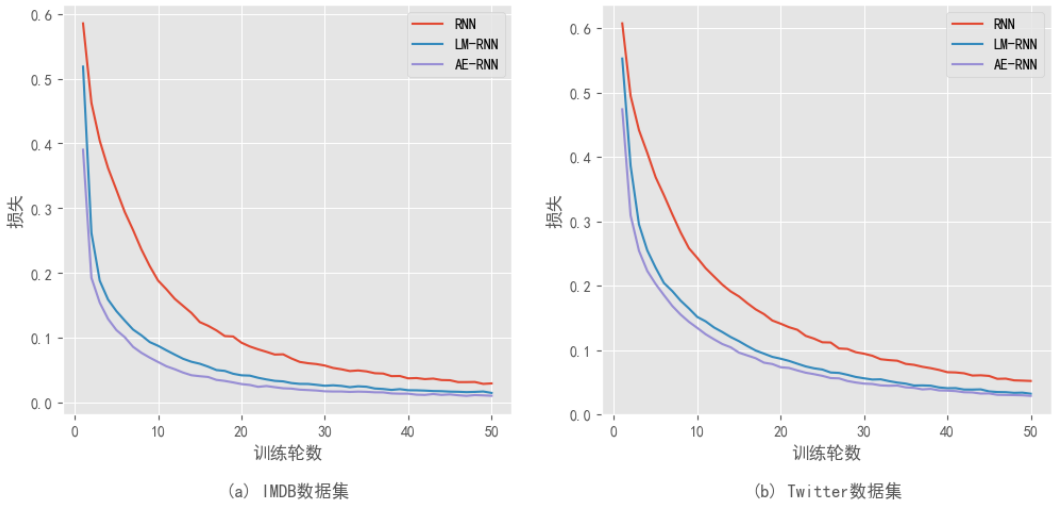

图4.8 不同数据集不同隐写分析方法损失收敛效率对比

4.3.4 改变预训练数据量对于检测性能的影响

在实际场景中，隐写检测方往往可以收集到比含密文本更多的载体文本，因为载体文本的收集难度远远低于含密文本。因此，此部分测试了提高预训练使用的载体文本数量对后续隐写分析性能的影响。表 4.3 和表 4.4 展示了本文所提方法在应对不同嵌入率与不同编码方式的隐写方法时，检测准确率随着预训练载体文本数量增加的变化情况。不难发现，通过使用更多的载体文本进行预训练，可以进一步提高后续隐写分析性能，这为自然文本远多于含密文本的现实场景提供了解决方案。

表4.3 增加预训练样本数对后续隐写分析检测准确率的影响(Twitter)

预训练样本数	隐写分析	Bins			FLC			VLC		
		1.000	2.000	3.000	1.000	2.000	3.000	1.000	2.150	3.260
6300	LM-RNN	0.908	0.902	0.899	0.909	0.900	0.901	0.908	0.898	0.896
	AE-RNN	0.918	0.906	0.908	0.911	0.904	0.897	0.904	0.907	0.903
12000	LM-RNN	0.918	0.909	0.909	0.914	0.902	0.908	0.912	0.904	0.902
	AE-RNN	0.920	0.912	0.912	0.915	0.913	0.907	0.915	0.909	0.910
18000	LM-RNN	0.920	0.914	0.915	0.913	0.909	0.909	0.912	0.906	0.910
	AE-RNN	0.920	0.916	0.919	0.920	0.921	0.910	0.922	0.912	0.913
>10G	BERT	0.936	**0.935**	**0.941**	0.935	**0.927**	**0.936**	**0.932**	0.930	**0.933**
	GPT2	**0.937**	0.934	0.932	**0.936**	0.925	0.931	0.927	**0.932**	0.931

表4.4 增加预训练样本数对后续隐写分析检测准确率的影响(IMDB)

预训练样本数	隐写分析	Bins			FLC			VLC		
		1.000	2.000	3.000	1.000	2.000	3.000	1.000	2.215	3.147
6300	LM-RNN	0.963	0.952	0.934	0.959	0.945	0.930	0.961	0.943	0.940
	AE-RNN	0.963	0.949	0.935	0.962	0.950	0.932	0.964	0.948	0.937
12000	LM-RNN	0.966	0.954	0.941	0.962	0.952	0.939	0.962	0.950	0.941
	AE-RNN	0.968	0.949	0.940	0.964	0.952	0.932	0.963	0.950	0.941
18000	LM-RNN	0.962	0.956	0.942	0.966	0.955	0.940	0.963	0.951	0.947
	AE-RNN	0.967	0.956	0.938	0.965	0.952	0.939	0.963	0.949	0.946
>10G	BERT	**0.975**	**0.967**	**0.963**	**0.975**	**0.967**	**0.961**	**0.972**	**0.968**	**0.963**
	GPT2	0.971	**0.967**	0.952	0.972	0.964	0.960	0.970	0.966	0.962

此外，本文还测试了通过微调大规模预训练语言模型 GPT2 及 BERT 实现文本隐写分析，这里 BERT 的版本为 $\text{BERT}_{\text{base,uncased}}$，GPT2 的模型大小为 117M，由表 4.3 及表 4.4 可以看出其均取得了极佳的检测性能，并超过了本文所提方法。这是由于 GPT2 及 BERT 都在超过 10G 的通用自然语料库上进行了语言模型预训练，远多于本文所使用的载体文本数据量，因此其检测性能更强，这也进一步证实了上文的结论。

4.4 本章小结

本章通过实验证实了语言模型不仅仅是一个高超的文本隐写者，也具有强大的文本隐写分析能力。因此，本文提出了一个基于微调预训练语言模型的高效文本隐写分析框架，将语言模型的先验知识迁移到文本隐写分析分类器中。本文提出了两种有效的预训练方法，一种是基于 RNN 语言模型，另一种是基于 RNN 的序列自编码器。后续的隐写分析实验结果表明，相较于随机初始化的 RNN 文本隐写分析方法，这两种方案都取得了不同程度的效果提升，同时大大提升了隐写分析训练时损失的收敛效率。同时，实验结果显示，随着预训练时使用的载体文本数量的增多，后续文本隐写分析的性能提升越大，这为载体文本远多于含密文本的现实场景提供了解决方案。

第五章 结论与展望

5.1 结论

文本隐写是一种将机密信息隐藏在文本中从而实现隐蔽通信的技术，该技术被广泛的应用于军事和国防等领域中。传统的修改式文本隐写通过修改指定文本以实现信息嵌入，具有嵌入率低和隐蔽性低等弊端。生成式文本隐写无需提前指定文本，直接由秘密信息引导生成含密文本，在嵌入率与隐蔽性等方面相较于传统方法而言具有明显的优势，应用前景广阔。但是现有方法中存在接收方提取机密信息所需要的附加信息多以及计算复杂度高等问题，为此本文提出了一种位置驱动的生成式文本隐写算法，直接嵌入单词级秘密信息从而显著缓解了上述问题。同时，为了对抗不法分子利用生成式文本隐写技术进行秘密通信且为了从对立面促进生成式文本隐写的发展，本文也研究了更高效的生成式文本隐写检测方法。本文研究内容总结如下：

⑴现有的生成式文本隐写算法在感知隐蔽性及统计隐蔽性方面取得了不错的成效，但是在信息提取过程中需要利用大量的附加信息并进行复杂的计算，这可能会暴露接收者的身份。因此，本文提出了一种基于 BERT 与吉布斯采样的位置驱动型生成式文本隐写算法，该算法由单词级秘密信息引导直接生成流畅的含密文本，含密文本中固定位置的单词串联起来即可恢复秘密信息，从而解决了传统方法在信息提取时存在的不足。所提方法采用双向预训练语言模型，在生成含密文本时充分利用文本的上下文语境信息，并搭配吉布斯采样经过多轮迭代的方式优化含密文本。实验结果显示，本文所提方法可以生成流利的含密文本，并具有较传统方法而言更好的抗隐写分析能力。最重要的是，本文所提方法在接收方行为受限及即时通信场景下具有良好的应用前景。

⑵现有生成式文本隐写检测算法将文本建模成序列，然后通过 CNN 和 RNN 等神经网络提取文本特征实现分类。然而文本中单词之间存在丰富的依存关系与共现信息，仅仅将文本建模成序列的方式不足以建模这部分信息。为此，本文将文本建模成表征能力更强图结构，图中的节点为文本中的单词，图中的边为单词

之间的关联强度，从而显示的建模单词之间的依存关系。并采用图神经网络学习文本的局部敏感语义特征与不同文本间的全局信息，实现了高效的含密文本检测。实验结果显示，本文所提方法可以检测多种生成式文本隐写算法，且相较于传统方法而言取得了更好的检测性能。

(3)现有的生成式文本隐写检测算法着力于设计更好的特征提取网络，而忽略了领域知识的挖掘。本文通过实验发现语言模型可以建模出含密文本与载体文本的分布差异，天然具有检测含密文本的能力。基于这一发现，本文进一步提出了一种微调预训练语言模型的文本隐写分析方法。该方法首先在载体文本上进行语言模型预训练，预训练方式尝试了传统语言模型与序列自编码器两种。然后通过微调的方式将语言模型学习到的先验知识迁移到隐写分析分类器中，以提高其检测性能。实验结果表明，所提出的方法相较于已有方法在检测性能与收敛速度上实现了双重提升，并且随着预训练数据集的增大，检测性能进一步提升，为应对含密文本收集难度远高于自然文本的现实场景提供了解决方案。

5.2 展望

本文提出了一种机密信息提取复杂度很低的生成式文本隐写算法，并从模型架构及挖掘领域知识的角度提升了生成式文本隐写检测算法的性能。尽管如此，生成式文本隐写及其检测技术由于其本身的特性以及技术的限制，依然存在许多问题有待解决。未来的研究工作可以从以下几个方面展开：

(1)现有的生成式文本隐写在生成含密文本时不限制其语义，这将导致发送方发送大量随机语义的文本，容易引起第三方的怀疑。此外，由于含密文本的语义得不到控制，会导致生成式文本隐写不适用于很多语义要求高的场景如社交聊天等。因此，研究可控语义的生成式文本隐写具有重要的意义，而这个问题到目前为止还没有得到有效的解决。

(2)现有的生成式文本隐写取得了很好的抗隐写分析效果，但是在应对基于大规模预训练语言模型的隐写分析方法时这种能力会急剧下降，因此进一步提高生成式文本隐写的统计隐蔽性依然值得研究。

(3)现有的生成式文本隐写检测算法取得了不错的效果，但都是建立在隐写

分析方可以收集到与自然文本相当的含密文本的理想情况下进行的，实际情况中，隐写分析方可能收集不到这么多含密文本甚至收集不到含密文本，研究这种严苛场景下的文本隐写检测算法具有重要的现实意义。

参考文献

[1] BERNAILLE L, TEIXEIRA R. Early recognition of encrypted applications[C]//Proceedings of the 8th Internatinoal Conference on Passive and Active Network Measurement, April 5-6, 2007, Louvain-la-neuve. Belgium: Springer, 2007: 165-175.

[2] BASH B A, GHEORGHE A H, PATEL M, et al. Quantum-secure covert communication on bosonic channels[J]. Nature Communications, 2015, 6(1): 1-9.

[3] 张新鹏, 钱振兴, 李晟. 信息隐藏研究展望[J]. 应用科学学报, 2016, 34(5): 475-489.

[4] 张大奇. 信息隐藏技术的研究与应用[D]. 西安: 西北大学, 2006.

[5] HERODOTUS. The Histories[M]. New York: Penguin Classics, 2003.

[6] STALLINGS W. The Advanced Encryption Standard[J]. Cryptologia, 2002, 26(3): 165-188.

[7] 梁小萍, 何军辉, 李健乾, 等. 隐写分析——原理, 现状与展望[J]. 中山大学学报: 自然科学版, 2004, 43(6): 93-96.

[8] SIMMONS G J. The prisoners' problem and the subliminal channel[C]//Proceedings of the Advances in Cryptology, August 21-24, 1983, Santa Barbara, California. Boston: Springer, 1983: 51-67.

[9] NEETA D, SNEHAL K, JACOBS D. Implementation of LSB Steganography and Its Evaluation for Various Bits[C]//Proceedings of the First International Conference on Digital Information Management, December 6-8, 2006, Christ College, Bangalore. India, IEEE, 2006: 173-178.

[10] JOHNSON N F, JAJODIA S. Exploring steganography: Seeing the unseen[J]. Computer, 1998, 31(2): 26-34.

[11] NI Z, SHI Y Q, ANSARI N, et al. Reversible data hiding[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2006, 16(3): 354-362.

[12] RAJENDRAN S, DORAIPANDIAN M. Chaotic map based random image steganography using LSB technique[J]. International Journal of Network Security, 2017, 19: 593-598.

[13] SAIDI M, HERMASSI H, RHOUMA R, et al. A new adaptive image steganography scheme based on DCT and chaotic map[J]. Multimedia Tools and Applications, 2017, 76(11): 13493-13510.

[14] WESTFELD A. F5-a steganographic algorithm[C]//Proceedings of the Information Hiding: 4th International Workshop, April 25-27, 2001, Pittsburgh, USA. Boston:Springer, 2001: 289-302.

[15] CHANG C Y, CLARK S. Practical Linguistic Steganography using Contextual Synonym Substitution and a Novel Vertex Coding Method[J]. Computational Linguistics, 2014, 40(2): 403-448.

[16] YANG Z L, GUO X Q, CHEN Z M, et al. RNN-Stega: Linguistic Steganography Based on Recurrent Neural Networks[J]. IEEE Transactions on Information Forensics and Security, 2019, 14(5): 1280-1295.

[17] BROWN P F, PIETRA V J D, MERCER R L, et al. An estimate of an upper bound for the entropy of English[J]. Computational Linguistics, 1992, 18(1): 31-40.

[18] KAHN D. The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet[M]. New York: Scribner, 1996.

[19] ZHOU Z, SUN H, HARIT R, et al. Coverless Image Steganography Without Embedding[C]//Proceedings of the International Conference on Cloud Computing and Security, August 13-15, 2015, Nanjing, China. Berlin: Springer, 2015: 123-132.

[20] CHEN X, SUN H, TOBE Y, et al. Coverless Information Hiding Method Based on the Chinese Mathematical Expression[C]//Proceedings of the International Conference on Cloud Computing and Security, August 13-15, 2015, Nanjing, China. Berlin: Springer, 2015: 133-143.

[21] CHEN X, CHEN S. Text coverless information hiding based on compound and selection of words[J]. Soft Computing, 2019, 23(15): 6323-6330.

[22] ZHOU Z, MU Y, YANG C N, et al. Coverless Multi-keywords Information Hiding Method Based on Text[J]. International Journal of Security and Its Applications, 2016, 10(9): 309-320.

[23] ZHANG J, SHEN J, WANG L, et al. Coverless Text Information Hiding Method Based on the Word Rank Map[C]//Proceedings of the International Conference on Cloud Computing and Security, July 29-31, 2016, Nanjing, China. Berlin: Springer, 2016: 145-155.

[24] LONG Y, LIU Y, ZHANG Y, et al. Coverless Information Hiding Method Based on Web

Text[J]. IEEE Access, 2019, 7: 31926-31933.

[25] TOPKARA M, TASKIRAN C M, III E J D. Natural language watermarking[C]// Proceedings of the Security, Steganography, and Watermarking of Multimedia Contents VII, January 17-20, 2005, San Jose, USA. Bellingham: SPIE, 2005: 441-452.

[26] BOLSHAKOV I A. A method of linguistic steganography based on collocationally-verified synonymy[C]//Proceedings of the 6th international conference on Information Hiding, May 23-25, 2004, Toronto, Canada. Berlin: Springer, 2004: 180-191.

[27] TASKIRAN C M, TOPKARA U, TOPKARA M, et al. Attacks on lexical natural language steganography systems[C]//Proceedings of the Security, Steganography, and Watermarking of Multimedia Contents VIII, January 15, 2006, San Jose, USA. Bellingham: SPIE, 2006: 97-105.

[28] RAFAT K F. Enhanced text steganography by changing word's spelling[C]//Proceedings of the 7th International Conference on Frontiers of Information Technology, December 16-18, 2009, Abbottabad, Pakistan. New York: ACM, 2009: 1-4.

[29] CHANG C Y, CLARK S. Linguistic Steganography Using Automatically Generated Paraphrases[C]//Proceedings of the Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics., June 2-4, 2010, Los Angeles, California. Stroudsburg: ACL, 2010: 591-599.

[30] TOPKARA M, TOPKARA U, ATALLAH M J. Words are not enough: sentence level natural language watermarking[C]//Proceedings of the 4th ACM international workshop on Contents protection and security. New York: ACM, 2006: 37-46.

[31] ATALLAH M J, MCDONOUGH C J, RASKIN V, et al. Natural language processing for information assurance and security: an overview and implementations[C]// Proceedings of the 2000 workshop on New security paradigms, September 18-21, 2000, Ballycotton, Ireland. New York: ACM, 2001: 51-65.

[32] MERAL H M, SANKUR B, SUMRU ÖZSOY A, et al. Natural language watermarking via morphosyntactic alterations[J]. Computer Speech & Language, 2009, 23(1): 107-125.

[33] ATALLAH M J, RASKIN V, CROGAN M, et al. Natural Language Watermarking: Design,

Analysis, and a Proof-of-Concept Implementation[C]//Proceedings of the International Workshop on Information Hiding, April 25-27, 2001, Pittsburgh, USA. Berlin: Springer, 2001: 185-200.

[34] LIU Y, SUN X, WU Y. A Natural Language Watermarking Based on Chinese Syntax[C]//Proceedings of the International Conference on Natural Computation, August 27-29, 2005, Changsha, China. Berlin: Springer, 2005: 958-961.

[35] MILLER G A. WordNet: An Electronic Lexical Database[M]. Massachusetts: MIT Press, 1998.

[36] BOLSHAKOV I A. A method of linguistic steganography based on collocationally-verified synonymy[C]//Proceedings of the International Workshop on Information Hiding, May 23-25, 2004, Toronto, Canada. Berlin: Springer,2004: 180-191.

[37] GROTHOFF C, GROTHOFF K, ALKHUTOVA L, et al. Translation-Based Steganography[C]//Proceedings of the International Workshop on Information Hiding, June 6-8, 2005, Barcelona, Spain. Springer, Berlin, Heidelberg, 2005: 219-233. Berlin: Springer, 2005: 219-233.

[38] STUTSMAN R, GROTHOFF C, ATALLAH M, et al. Lost in just the translation[C]//Proceedings of the 2006 ACM symposium on Applied computing, April 23-27, 2006, Dijon, France. New York: ACM, 2006: 338-345.

[39] MENG P, SHI Y Q, HUANG L, et al. LinL:Lost in n-best list[C]//Proceedings of the 3th International Conference on Information Hiding, May 18-20, 2011, Prague, Czech Republic. Berlin: Springer, 2011: 329-341.

[40] VENUGOPAL A, USZKOREIT J, TALBOT D, et al. Watermarking the Outputs of Structured Prediction with an application in Statistical Machine Translation[C]//Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, July 27-31, 2011, Edinburgh, UK. Stroudsburg: ACL, 2011: 1363-1372.

[41] CHANG C Y, CLARK S. The Secret's in the Word Order: Text-to-Text Generation for Linguistic Steganography[C]//Proceedings of the International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 8-15, 2012, Mumbai,

India. Mumbai: Indian Institute of Technology Bombay, 2012: 511-528.

[42] UEOKA H, MURAWAKI Y, KUROHASHI S. Frustratingly Easy Edit-based Linguistic Steganography with a Masked Language Model[C]//Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, June 6-11, 2021, Online. Stroudsburg: ACL, 2021: 5486-5492.

[43] DEVLIN J, CHANG M W, LEE K, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[C]//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, June 2-7, 2019, Minneapolis, USA. Stroudsburg: ACL, 2019: 4171-4186.

[44] MIKOLOV T, KARAFIÁT M, BURGET L, et al. Recurrent neural network based language model[C]//Proceedings of the 11th Annual Conference of the International Speech Communication Association, September 26-30, 2010, Makuhari, Japan. New York: ISCA, 2010: 1045-1048.

[45] PHAM N Q, KRUSZEWSKI G, BOLEDA G. Convolutional Neural Network Language Models[C]//Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, November 1-4, 2016, Austin, USA. Stroudsburg: ACL, 2016: 1153-1162.

[46] VASWANI A, SHAZEER N, PARMAR N, et al. Attention Is All You Need[J]. arXiv preprint arXiv:1706.03762, 2017.

[47] HOCHREITER S, SCHMIDHUBER J. Long Short-term Memory[J]. Neural computation, 1997, 9: 1735-1780.

[48] DAI W, YU Y, DAI Y, et al. Text Steganography System Using Markov Chain Source Model and DES Algorithm[J]. Journal of Software, 2010, 5(7): 785-792.

[49] DAI W, YU Y, DENG B. BinText steganography based on Markov state transferring probability[C]//Proceedings of the 2nd International Conference on Interaction Sciences Information Technology, 24-26 November, 2009, Seoul, Korea. New York: ACM Press, 2009: 1306-1311.

[50] MORALDO H H. An Approach for Text Steganography Based on Markov Chains[J]. arXiv preprint arXiv:1409.0915, 2014.

[51] YANG Z, JIN S, HUANG Y, et al. Automatically Generate Steganographic Text Based on Markov Model and Huffman Coding[J]. arXiv preprint arXiv:1811.04720, 2018.

[52] SHNIPEROV A N, NIKITINA K A. A text steganography method based on Markov chains[J]. Automatic Control and Computer Sciences, 2016, 50(8): 802-808.

[53] FANG T, JAGGI M, ARGYRAKI K. Generating Steganographic Text with LSTMs[C]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, July 30 - August 4, 2017. Vancouver, Canada. Stroudsburg: ACL, 2017: 100-106.

[54] ZHANG S, YANG Z, YANG J, et al. Provably Secure Generative Linguistic Steganography[C]//Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, August 1-6, 2021, Online. Stroudsburg: ACL, 2021: 3046-3055.

[55] ZHOU X, PENG W, YANG B, et al. Linguistic Steganography Based on Adaptive Probability Distribution[J]. IEEE Transactions on Dependable and Secure Computing, Early Access.

[56] YANG Z L, ZHANG S Y, HU Y T, et al. VAE-Stega: Linguistic Steganography Based on Variational Auto-Encoder[J]. IEEE Transactions on Information Forensics and Security, 2021, 16: 880-895.

[57] DAI F, CAI Z. Towards Near-imperceptible Steganographic Text[C]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence, July 28-August 2, 2019, Florence, Italy. Stroudsburg: ACL, 2019: 4303-4308.

[58] ZIEGLER Z, DENG Y, RUSH A. Neural Linguistic Steganography[C]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. Hong Kong, China. Stroudsburg: ACL 2019: 1210-1215.

[59] SHEN J, JI H, HAN J. Near-imperceptible Neural Linguistic Steganography via Self-Adjusting Arithmetic Coding[J]. arXiv preprint arXiv:2010.00677, 2020.

[60] ZHANG S, YANG Z, YANG J, et al. Linguistic Steganography: From Symbolic Space to Semantic Space[J]. IEEE Signal Processing Letters, 2021, 28: 11-15.

[61] KANG H, WU H, ZHANG X. Generative Text Steganography Based on LSTM Network and

Attention Mechanism with Keywords[J]. Electronic Imaging, 2020, 2020(4): 291-1-291-298.

[62] WEN J, ZHOU X, LI M, et al. A novel natural language steganographic framework based on image description neural network[J]. Journal of Visual Communication and Image Representation, 2019, 61: 157-169.

[63] YANG Z, XIANG L, ZHANG S, et al. Linguistic Generative Steganography With Enhanced Cognitive-Imperceptibility[J]. IEEE Signal Processing Letters, 2021, 28: 409-413.

[64] LI Y, ZHANG J, YANG Z, et al. Topic-aware neural linguistic steganography based on knowledge graphs[J]. ACM/IMS Transactions on Data Science, 2021, 2(2): 1-13.

[65] YANG H, CAO X. Linguistic Steganalysis Based on Meta Features and Immune Mechanism[J]. Chinese Journal of Electronics, 2010, 19: 661-666.

[66] MENG P, HANG L, YANG W, et al. Linguistic Steganography Detection Algorithm Using Statistical Language Model[C]//Proceedings of the 2009 International Conference on Information Technology and Computer Science, July 25-26, 2009, Kiev, Ukraine. Piscataway: IEEE, 2009: 540-543.

[67] SAMANTA S, DUTTA S, SANYAL G. A real time text steganalysis by using statistical method[C]//Proceedings of the 2016 IEEE International Conference on Engineering and Technology, March 17-18, 2016, Coimbatore, India. Piscataway: IEEE, 2016: 264-268.

[68] DIN R, YUSOF S A M, AMPHAWAN A, et al. Performance Analysis on Text Steganalysis Method Using A Computational Intelligence Approach[J]. Proceeding of the Electrical Engineering Computer Science and Informatics, 2015, 2(1): 67-73.

[69] CHEN Z, HUANG L, YU Z, et al. Linguistic Steganography Detection Using Statistical Characteristics of Correlations between Words[C]//Proceedings of the International Workshop on Information Hiding, May 19-21, 2008, Santa Barbara, USA. Berlin: Springer, 2008: 224-235.

[70] XIANG L, SUN X, LUO G, et al. Linguistic steganalysis using the features derived from synonym frequency[J]. Multimedia tools and applications, 2014, 71(3): 1893-1911.

[71] YANG Z, HUANG Y, ZHANG Y J. A Fast and Efficient Text Steganalysis Method[J]. IEEE Signal Processing Letters, 2019, 26(4): 627-631.

[72] WEN J, ZHOU X, ZHONG P, et al. Convolutional Neural Network Based Text Steganalysis[J]. IEEE Signal Processing Letters, 2019, 26(3): 460-464.

[73] YANG Z, HUANG Y, ZHANG Y J. TS-CSW: text steganalysis and hidden capacity estimation based on convolutional sliding windows[J]. Multimedia Tools and Applications, 2020, 79(25-26): 18293-18316.

[74] YANG Z, WEI N, SHENG J, et al. TS-CNN: Text Steganalysis from Semantic Space Based on Convolutional Neural Network[J]. arXiv preprint arXiv:1810.08136, 2018.

[75] YANG Z, WANG K, LI J, et al. TS-RNN: Text Steganalysis Based on Recurrent Neural Networks[J]. IEEE Signal Processing Letters, 2019, 26(12): 1743-1747.

[76] YANG H, BAO Y, YANG Z, et al. Linguistic Steganalysis via Densely Connected LSTM with Feature Pyramid[C]//Proceedings of the 2020 ACM Workshop on Information Hiding and Multimedia Security, June 22-24, 2020, Denver, USA. New York: ACM, 2020: 5-10.

[77] NIU Y, WEN J, ZHONG P, et al. A Hybrid R-BILSTM-C Neural Network Based Text Steganalysis[J]. IEEE Signal Processing Letters, 2019, 26(12): 1907-1911.

[78] BAO Y, YANG H, YANG Z, et al. Text Steganalysis with Attentional LSTM-CNN[J]. arXiv preprint arXiv:1912.12871, 2019.

[79] GELFAND A E, SMITH A F M. Sampling-Based Approaches to Calculating Marginal Densities[J]. Journal of the American Statistical Association, 1990, 85(410): 398-409.

[80] SU J, XU J, QIU X, et al. Incorporating Discriminator in Sentence Generation: a Gibbs Sampling Method[J]. arXiv preprint arXiv:1802.08970, 2018.

[81] WANG A, CHO K. BERT has a Mouth, and It Must Speak: BERT as a Markov Random Field Language Model[J]. arXiv preprint arXiv:1902.04094, 2019.

[82] LIU Y, OTT M, GOYAL N, et al. RoBERTa: A Robustly Optimized BERT Pretraining Approach[J]. arXiv preprint arXiv:1907.11692, 2019.

[83] DOSOVITSKIY A, BEYER L, KOLESNIKOV A, et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale[J]. arXiv preprint arXiv:2010.11929, 2021.

[84] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition,, June 27-30, 2016, Las

Vegas, USA. Piscataway: IEEE, 2016: 770-778.

[85] WOLF T, DEBUT L, SANH V, et al. HuggingFace's Transformers: State-of-the-art Natural Language Processing[J]. arXiv preprint arXiv:1910.03771, 2020.

[86] ZHU Y, KIROS R, ZEMEL R, et al. Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books[J]. arXiv preprint arXiv:1506.06724, 2015.

[87] RADFORD A, WU J, CHILD R, et al. Language models are unsupervised multitask learners[J]. OpenAI blog, 2019, 1(8): 9.

[88] PENG W, ZHANG J, XUE Y, et al. Real-Time Text Steganalysis Based on Multi-Stage Transfer Learning[J]. IEEE Signal Processing Letters, 2021, 28: 1510-1514.

[89] RADFOR A, NARASIMHAN K, SALIMANS T, et al. Improving language understan ding by generative pre-training[EB/OL]. (2018-6-12)[2022-5-20]. https://s3-us-west-2.am azonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding _paper.pdf

[90] SCARSELLI F, GORI M, TSOI A C, et al. The Graph Neural Network Model[J]. IEEE Transactions on Neural Networks, 2009, 20(1): 61-80.

[91] ZHUANG C, MA Q. Dual Graph Convolutional Networks for Graph-Based Semi-Supervised Classification[C]//Proceedings of the 2018 World Wide Web Conference on World Wide Web, April 23-27, 2018, Lyon, France. New York :ACM, 2018: 499-508.

[92] HUANG L, MA D, LI S, et al. Text Level Graph Neural Network for Text Classification[C]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, November 3-7, 2019, Hong Kong, China. Stroudsburg: ACL, 2019: 3444-3450.

[93] DUVENAUD D K, MACLAURIN D, IPARRAGUIRRE J, et al. Convolutional networks on graphs for learning molecular fingerprints[J]. arXiv preprint arXiv:1509.09292, 2015.

[94] GILMER J, SCHOENHOLZ S S, RILEY P F, et al. Neural message passing for quantum chemistry[C]//International conference on machine learning, 6-11 August, 2017, Sydney, Australia. New York: PMLR, 2017: 1263-1272.

[95] KEARNES S, MCCLOSKEY K, BERNDL M, et al. Molecular graph convolutions: moving beyond fingerprints[J]. Journal of computer-aided molecular design, 2016, 30(8): 595-608.

[96] SCHÜTT K T, ARBABZADAH F, CHMIELA S, et al. Quantum-chemical insights from deep tensor neural networks[J]. Nature Communications, 2017, 8(1): 13890.

[97] KIPF T N, WELLING M. Semi-supervised classification with graph convolutional networks[J]. arXiv preprint arXiv:1609.02907, 2016.

[98] BRUNA J, ZAREMBA W, SZLAM A, et al. Spectral Networks and Locally Connected Networks on Graphs[J]. arXiv preprint arXiv:1312.6203, 2014.

[99] LI R, WANG S, ZHU F, et al. Adaptive Graph Convolutional Neural Networks[J]. arXiv preprint arXiv:1801.03226, 2018.

[100] HAMMOND D K, VANDERGHEYNST P, GRIBONVAL R. Wavelets on graphs via spectral graph theory[J]. Applied and Computational Harmonic Analysis, 2011, 30(2): 129-150.

[101] GO A, BHAYANI R, HUANG L. Twitter sentiment classification using distant supervision[J]. CS224N project report, Stanford, 2009, 1(12): 2009.

[102] MAAS A L, DALY R E, PHAM P T, et al. Learning Word Vectors for Sentiment Analysis[C]//Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, June 19-24, 2011, Portland, USA. Stroudsburg: ACL, 2011: 142-150.

[103] PENNINGTON J, SOCHER R, MANNING C. GloVe: Global Vectors for Word Representation[C]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, October 25-29, 2014, Doha, Qatar. Stroudsburg: ACL, 2014: 1532-1543.

[104] KINGMA D P, BA J. Adam: A Method for Stochastic Optimization[J]. arXiv preprint arXiv:1412.6980, 2017.

[105] SRIVASTAVA N, HINTON G, KRIZHEVSKY A, et al. Dropout: A Simple Way to Prevent Neural Networks from Overfitting[J]. Journal of Machine Learning Research, 2014, 15(56): 1929-1958.

[106] DAI A M, LE Q V. Semi-supervised Sequence Learning[J]. arXiv preprint arXiv:1511.01432, 2015.

[107] SUTSKEVER I, VINYALS O, LE Q V. Sequence to sequence learning with neural networks[J]. arXiv preprint arXiv:1409.3215, 2014.

作者在攻读硕士学位期间公开发表的论文

[1] **Yi B**, WU H, FENG G, et al. ALiSa: Acrostic linguistic steganography based on BERT and Gibbs sampling[J]. IEEE Signal Processing Letters, 2022, 29: 687-691. **（SCI: 000767838300003, EI: 20220811686580）**

[2] **Yi B**, WU H, FENG G, et al. Exploiting language model for efficient linguistic steganalysis[C]// Proceedings of the 2022 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), May 23-27, 2022, Singapore, Singapore. Piscataway:IEEE, 2022: 3074-3078.**（EI, 已发表）**

[3] WU H, **YI B**, DING F, et al. Linguistic steganalysis with graph neural networks[J]. IEEE Signal Processing Letters, 2021, 28: 558-562. **（SCI: 000633387100004, EI: 20211010020266）**

作者在攻读硕士学位期间所参与的项目

[1] 国家自然科学基金青年项目“社交网络多用户协同的行为隐写”（项目编号：61902235）.

致　　谢

东区小河旁的花开了又谢，树叶绿了又黄，转眼间，已是经历了三个轮回，这也预示着我的研究生生涯已经接近了尾声。细细回首这三年，一开始进入学校的懵懂无知，初接触科研的好奇与迷茫，慢慢努力前行的充实与满足，实习求职的辛苦与艰难，一切彷佛都历历在目。逝者如斯夫，不舍昼夜，无论是痛苦还是快乐的回忆，这些都是我成长的足迹，都让我受益良多。值此毕业之际，请允许我向在学习生活中一切帮助过我的人们致以谢意。

首先要感谢的是我的研究生导师吴汉舟老师。吴老师年轻有为，跟我的年龄相差不大，在学习和生活中可谓是亦师亦友的关系。在学术上，从一开始的研究选题，到最后的论文写作，吴老师都亲力亲为，悉心指导，比如我所发表的小论文都经过吴老师逐字逐句的修改。吴老师带我走进了科研的大门，带我领略了科研的魅力，其严谨的科研态度是我学习的榜样。在生活上，吴老师也给予了兄长般的关怀，经常为我们排忧解难，我生病住院时吴老师经常打电话关怀，这让离家千里的我感受到了家人般的温暖。同时我还要感谢张新鹏老师，与张老师的几次交谈都让我受益良多，印象最深的是，张老师说我们要做真正自己觉得有意义的课题，每天一睁眼就要想到它，以它为乐趣进行奋斗，张老师的科研高度让我钦佩。我还要感谢同课题组的冯国瑞，任艳丽，侯丽敏，吕东辉老师在科研上对我提供的帮助。

除此之外，我最要感谢的是我的家人，感谢我的爷爷奶奶，外公外婆，父母亲，还有对我生活上百般关心的姑姑，在学习上给予我鼓励的叔叔，舅舅，以及我所有的家人。

人生的每一段旅程就像是一次次轮回，初中毕业的时候我躺在宿舍看着天花板发呆，心里想着下一次再这样发呆是在哪里呢，高中毕业的时候我又如期想到这一幕，然后又是大学，研究生，此时此刻，恰如彼时彼刻，那么下一次又会在哪里呢？