中图分类号：　　单位代号：10280
密　　级：　　学　　号：20721306

上海大学

硕士学位论文

SHANGHAI UNIVERSITY
MASTER'S DISSERTATION

题目	语义可控的自然语言隐写技术研究

作　　者　杨天予
学科专业　信号与信息处理
导　　师　张新鹏
完成日期　2023 年 5 月

姓　　名：杨天予　　学号：　20721306

论文题目：语义可控的自然语言隐写技术研究

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：杨天予　　学号：20721306

论文题目：语义可控的自然语言隐写技术研究

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

语义可控的自然语言隐写技术研究

姓　　名：杨天予
导　　师：张新鹏
学科专业：信号与信息处理

上海大学通信与信息工程学院

二〇二三年五月

A Dissertation Submitted to Shanghai University for the Degree of Master in Engineering

Research of Semantic Preserving Linguistic Steganography

Candidate：Tianyu Yang

Supervisor：Xinpeng Zhang

Major：Signal and Information Processing

School of Communication and Information Engineering

Shanghai University

May, 2023

摘　要

自然语言隐写是一种向文本中嵌入机密信息以实现隐蔽通信的安全技术，对国家信息与网络安全有重要意义。现有的自然语言隐写方法主要分为两类：修改式自然语言隐写和生成式自然语言隐写。前者通过修改给定的文本实现机密信息的嵌入，虽能较好地控制载密文本的语义信息，但嵌入量较小；后者利用自然语言模型直接生成载密文本，虽能增大嵌入量，但难以控制文本语义。为了同时实现语义可控和高嵌入效率，本文研究了自然语言隐写新方法，取得的研究成果如下：

1) 针对主流生成式自然语言隐写方法难以控制载密文本语义的问题，本文提出了一种新型的自然语言隐写框架。该框架利用自监督释义技术，通过两次语言转换，以生成式自然语言隐写的方式实现秘密信息的嵌入。基于该框架设计的算法具备生成式自然语言隐写方法的优势，可保证高嵌入效率。同时，由于存在原始文本的约束，其同样具备修改式自然语言隐写方法的优势，能够控制隐写文本与原始文本的语义一致，从而保证了隐蔽通信的安全性。实验表明，基于该框架设计的算法在语义一致性上优于主流的生成式自然语言隐写方法，在嵌入量和嵌入成功率指标上也领先于修改式方法。

2) 针对主流生成式自然语言隐写方法因信息编码的随机性而导致载密文本语义质量低的问题，本文提出了一种基于语义感知编码的自然语言隐写新方法。该隐写编码将语义相似的词汇均匀地映射到不同比特流上，以提升含有所需语义的词汇映射到秘密信息上的可能性。实验表明，基于该隐写编码的生成式自然语言隐写方法可进一步提升隐写文本质量。另外，该编码方式可以减少隐蔽通信所需共享的辅助信息的数量，提高了隐写的隐蔽性。

关键词：自然语言隐写，自然语言处理，语义一致性，信息隐藏

ABSTRACT

Linguistic steganography is an information security technology that embeds secret information into text to achieve covert communication, which is important to the national information and cyber security. The current mainstream linguistic steganographic methods are mainly divided into two categories: modification-based and generation-based method. The former realizes the embedding of secret information by modifying the given text. Although it can control the semantic information of the steganographic text well, the embedding payload is small; the latter uses the natural language model to directly generate the steganographic text. Although it can increase the amount of payload, it is difficult to control the semantics. In order to achieve semantic consistency and large embedding payload at the same time, this thesis studies new steganographic methods. The achievement are as follows:

1) Aiming at the shortage that current generation-based steganographic method cannot achieve semantic consistency, this thesis proposes a new linguistic steganographic framework, which utilizes the self-supervised paraphrasing technology to embed secret information in a generative way through two language transformations. The embedding algorithm designed based on this framework has the advantages of generation-based method, which can ensure high embedding efficiency. At the same time, due to the constraint of the original text, it also has the advantages of the modification-based method, which can maintain the semantic consistency between the steganographic text and the original text to ensure the security of covert communication. Experiments have presented that the embedding algorithm designed based on this framework exceeds the mainstream generation-based method in terms of semantic consistency, and is also ahead of the modification-based method in terms of embedding payload and success rate metrics.

2) Aiming at the problem that the mainstream generation-based embedding method suffers the problem of low semantic quality of steganographic text due to the randomness of information encoding, this thesis proposes a new encoding strategy. This steganographic encoding method evenly maps semantically similar words to different bit streams to enhance the possibility that there is a word with the desired meaning mapped to the secret information. Experiments have shown that the generation-based linguistic steganographic method based on the proposed encoding can further improve the quality of text. In addition, this encoding method can reduce the amount of auxiliary information which is required to share in covert communication, and improves the concealment of steganography.

Keywords: Linguistic Steganography, Natural Language Processing, Semantic Consistency, Data Hiding

目　　录

第一章　绪论

1.1 课题来源

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

1.2 课题研究的目的和意义

隐写术是一种将秘密信息嵌入在信息载体中以实现隐蔽通信的技术。载体可以是多种形式的，例如图像、文本、音频和视频[1-4]。发送方可以通过轻微修改原始的自然载体的方式嵌入秘密信息。所得的含密载体将与原始载体十分相似，所以不会引起第三方的怀疑。在发送方共享了秘密信息提取的规则和辅助信息后，接收方可完整、准确地从含密载体中提取秘密信息。

加密技术同样是安全通信中重要的研究内容之一。在密码学的研究中，真正所要传输的信息称为明文，经过加密后传输的文本称为密文。明文为人类可认知的自然文本，而经过加密后的密文则通常变得晦涩难懂，从而使得第三方无法从密文中直接获得发送方想要表达的真正含义。尽管如此，密文的传输已透露出信道中存在秘密信息的传递，这将引起攻击者的怀疑。

不同于加密技术保护了发送方所传输的文字，隐写术保护了发送方所希望表达的含义。通过自然语言隐写方法，发送方可在自然可读的文本中嵌入秘密信息，而尽管发送方所传输的隐写文本看似正常，但它并非真正想要传输的信息。接收方获取隐写文本后，通过事先约定的提取方法提取秘密信息。由于在传输信道中的文本是自然可读的，自然语言隐写技术降低了被攻击者怀疑的风险。

自然语言隐写也称文本隐写，主要分为两大类：修改式和生成式方法。修改式方法对自然文本进行轻微修改以嵌入秘密信息，所得的隐写文本与原始的自然文本语义相似，但其不足是嵌入量小。生成式方法借助了文本生成技术，直接生成一段含密文本，大幅提升了嵌入量。但生成式方法摒弃了原始文本的约束，故所得的隐写文本看似自然且正常，但无法实现语义可控，留下了潜在的安全隐患。

随着社交网络的进步和发展，每天都有大量文本在社交平台传播。海量文本也为自然语言隐写技术提供了掩护，因隐写文本同样也是自然可读的文本，难以引起其他用户、监测机构或恶意攻击者的怀疑。这样的使用场景减轻了生成式自然语言隐写方法的不足所带来的影响，但若细究隐写文本的质量，则语义可控仍然是自然语言隐写中的一个重要课题。

以社交软件“微博”为例。由于每一个人都会有自己的个性和喜好，所以每一个用户所发布的“微博”总会有类似的特质。若希望以发布的“微博”文字为信息隐藏的载体，并且使用主流的生成式方法作为嵌入策略，则容易生成偏离“微博”使用者特质的文本，从而引起其他人的怀疑。即使对生成文本加以主题限制，若不严格控制语义，也容易生成立场对立的文本。所以，为了使生成式自然语言隐写方法适配社交场景，本文研究了语义可控的自然语言隐写技术，以实现严格的语义控制。

综上所述，对于自然语言隐写，目前的修改式和生成式方法都有明显的优势和劣势。这两类方法也体现了隐写术中存在的一般规律，即嵌入量越大，通信的隐蔽性和安全性越低。尽管一般规律无法打破，但提升隐写术的总体性能是该研究领域的主要目的，即提升固定嵌入量下的隐写文本的质量。针对两类自然语言隐写方法的特性，本文提出了一种结合了修改式和生成式的优势的自然语言隐写方法，旨在既保留修改式方法语义可控的特性，又能具备生成式方法嵌入量大的特点，以提升自然语言隐写的总体性能。

1.3 自然语言隐写相关介绍

隐写领域的早期研究者 Simmons 为解释隐写术提供了一个经典的场景，称为“囚犯问题”[5]。“囚犯问题”指在两个不同的牢房中关押着 Alice 和 Bob 两名囚犯，他们被允许正常的交流，但他们之间的交流需要在狱警 Eve 的监管之下进行。现在 Alice 期望向 Bob 传递越狱相关的信息，他们显然不希望 Eve 从中发现端倪，于是他们将秘密信息嵌入到自然载体中以避免怀疑，从而实现隐蔽通信。Alice 和 Bob 之间组成了一个隐写系统，而狱警 Eve 则代表了检测秘密信息是否存在的监测方。

1.3.1 自然语言隐写基本框架

自然语言隐写基本框架如图 1.1 所示。在该框架中，最为核心的两模块是数据嵌入和数据提取模块。数据嵌入过程由自然语言隐写算法所决定。在数据嵌入过程中，发送方可从数据集中自行挑选合适的自然文本作为原始文本，选择特定的嵌入算法，再使用密钥确定嵌入算法的细节，便可将所需传递的秘密信息嵌入原始文本中。在嵌入阶段结束后，发送方将所得的隐写文本通过公共信道发送给接收方。接收方可根据隐写文本和事先通过安全信道传输的密钥提取秘密信息。

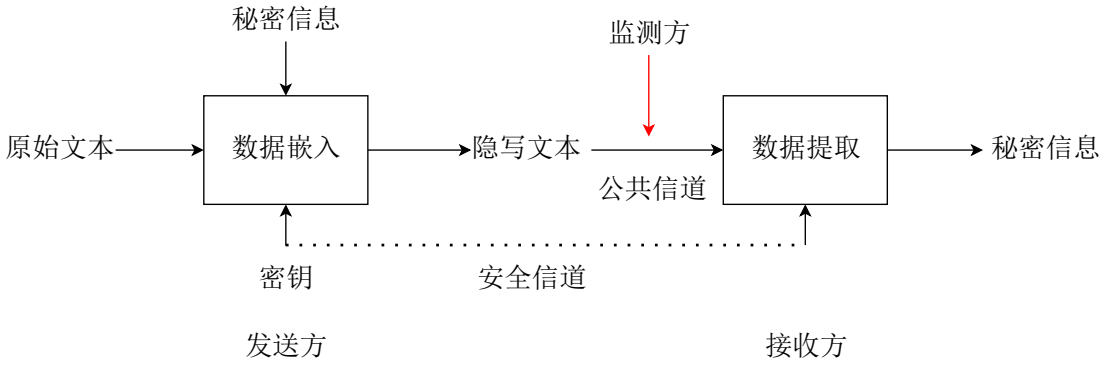

图 1.1　自然语言隐写的基本框架

1.3.2 自然语言隐写的技术难点

随着深度学习和神经网络的发展，使用不同载体的隐写研究都开始大量使用深度学习技术。而在深度学习领域，以图像为载体的研究总是领先于其他载体。对于隐写技术而言，自然语言隐写的发展也落后于图像隐写。尽管文本是人类日常生活中最常用的信息载体，但由于其高度编码和低冗余的特性，使得文本隐写无法简单沿用图像隐写中的技术。下面将通过对比图像和文本在隐写场景下的不同，对自然语言隐写中的难点展开阐述。

1)　文本编码

计算机无法直接识别图像或文字，对于任意信息，第一步都需要将其转化为数值，也即编码操作。对于图像而言，以 32×32 大小的灰度图片为例，图像处理通常以一个像素点为一个编码单元，每一个像素点的颜色可以转化为 0-255 的

像素值，对图像完全编码后可得一个 32×32 的矩阵。然而，对于文本而言，以何种单位作为一个编码单元在自然语言研究初期便广受争议。下面将以英文为例介绍文本编码发展的三个阶段。

a) 以一个字母为一个单元[6]：26 个字母足以组成所有英文单词，这样的编码方式使得每一个编码单元可转化为 0-26 的小数值，但这会导致一段英文文本被切分为超大量片段，细粒度极高，使得神经网络难以训练。尤其在如今自然语言处理着重于长文本处理的趋势下，以一个字母为一个编码单元的文本分割方法的不足被放大，所以该方法逐渐被淘汰。

b) 以一个单词为一个单元[7]：以单词为一个编码单元使得细粒度大幅降低，但世界上英文单词的总数超过了 10^5 的数量级，这使得每一个单元将转化为大数值，从而导致训练模型中的向量维度增大且计算变得复杂，仍将使得神经网络难以训练。考虑到英文单词中有大量的不常用词汇，该做法后演化为仅对常见的单词重新组成一个新的词典，而将不常见的单词统一标注为“<unk>”，意为“unknown”，并将其编为一个统一码，该统一码代表了所有的不常用词汇和不可识别对象。这样的做法有利于神经网络的训练并在自然语言处理领域的各项任务中取得了一定的进展。但这种文本分割方法引起了超出词典范围(Out Of Vocabulary, OOV)的问题，即文本中有大量的单词不存在于新组成的词典中，使得文本出现了大量“<unk>”。含有“<unk>”符号的文本更容易引起人们的怀疑，所以以一个单词为一个编码单元的做法仍是值得被优化的。

c) 以一个“子词”为一个单元[8]：为解决 OOV 问题，目前主流的做法是将一个“子词”作为一个单位。“子词”可以是一个完整的单词，比方说“the”，也可以是部分单词，比方“#ing”。“#”意为“ing”仅是一个完整词的后半部分，若“study”和“#ing”为两个相邻“子词”，则需根据“#”规则将两个“子词”合并为一个完整的单词“studying”。该思想考虑到大多数复杂且出现频率较低的词汇是由简单且出现频率较高的“子词”组成。由此，以“子词”为单位的词典相比以词为单位的相同大小的词典可表示的完整单词数更多。为简便描述，本文将一个文本编码单元统称为“词”。

图像与文本编码的差异也导致了图像与自然语言隐写方法的不同。对于图像

而言，对某一像素点的像素值±1 可代表嵌入比特“1”，保持该像素值不变代表嵌入比特“0”。由于对像素值±1 的操作很难被人类以视觉观察的方式所发现，这样简便的做法也可以认为是不可感知的。然而，对于文本而言，若将一个单词修改为词典中序号相邻的另一个词往往导致文本语义完全改变，甚至使得文本不合语法，极易被监测方怀疑，所以自然语言隐写难以沿用图像隐写中的各种方法，使得自然语言隐写更具挑战性。如图 1.2 所示，相邻编码序号的颜色块几乎没有差异，而相邻编码序号对应的单词除了首字母相同以外没有任何语义上的相关性，所以自然语言隐写无法通过简单对编码序号±1 实现。

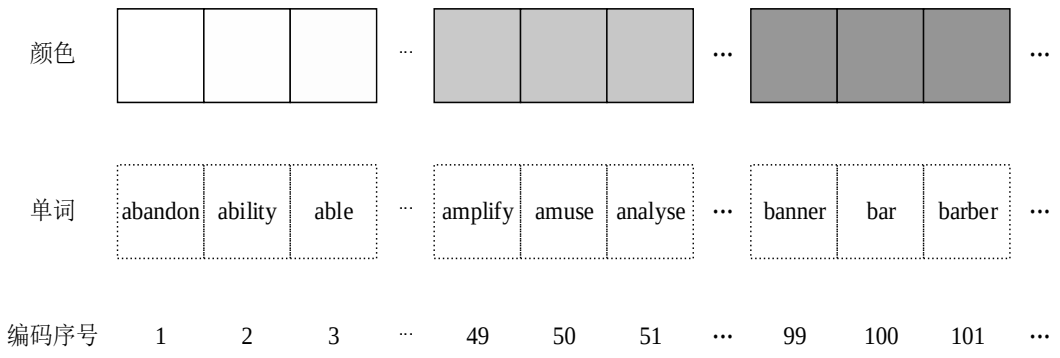

图 1.2　图像与文本相邻编码元素的差异

2) 文本生成

对于生成式自然语言隐写，文本生成是其上游研究领域，文本生成的难点自然影响了生成式自然语言隐写的研究。与图像生成技术可一步生成完整的图像不同，目前文本生成技术仍无法实现一步生成一整段文本，而只能一步生成一个编码单元，通过循环复用的方式实现整段文本的生成，这使得文本生成任务的计算复杂度较高。如图 1.3 所示，文本生成框架主要分为三个部分：编码器、解码器和采样策略，以下将以翻译任务为例，介绍文本生成框架。

编码器用于特征提取，为后续解码器的生成步骤提供约束条件，对于英文-中文翻译任务而言，英文文本即为约束条件，约束条件决定了后续生成内容。解码器用于提供条件转移概率。条件转移概率指在一个生成步骤中，词典中各词作为最终输出的概率值。概率值越大代表模型认为将所对应的词作为最终输出更为

合适。最终的输出将根据条件转移概率和采样策略共同决定。

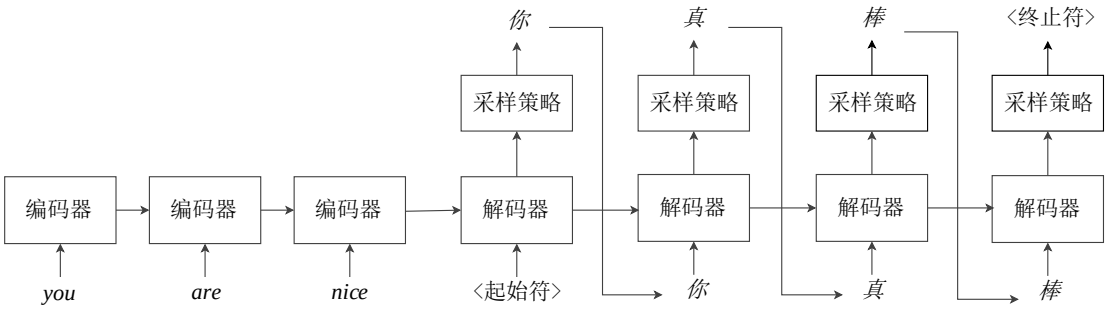

图 1.3　文本生成模型

循环神经网络(Recurrent Neural Network, RNN) [9]是首个广泛应用于自然语言处理的神经网络模型之一，它将文本翻译任务定义为 Seq2Seq(Sequence-to-Sequence)结构，也即从一串序列文本中提取约束特征，从而生成另一段序列文本。由于文字的顺序和位置关系至关重要，输入文本和输出文本都被定义为链式结构，这使得计算机无法并行处理。编码器和解码器的重复使用使得 RNN 结构的计算复杂度较高。尽管后续长短期记忆神经网络(Long Short-Term Memory, LSTM) [10]和门控循环神经网络(Gated Recurrent Unit, GRU) [11]模型通过增加控制门限参数变量的方式使得模型在长文本上有更好的表现，但仍不能改变该类型的模型无法实现并行计算的弊端。

Transformer[12]在 RNN 结构的基础上进行改进，在编码端，Transformer 使用注意力机制[13]和位置向量取代了部分链式结构，在保留了文字的顺序和位置关系的前提下实现了并行特征提取。尽管 Transformer 在编码端的优化为后续自然语言处理的研究做出了突出的贡献，但其解码端仍使用逐词生成的方式，所以无法避免解码器的复用。

随着自然语言处理的研究不断深入，研究表明，人类表达的方式与模型评估的结果不尽相同。人类不仅常常选择文本生成模型中的次优选，还经常选择机器认为概率值较低的选项[14]。于是，为了更好地模拟人类的表达方式，采样策略仍在不断被研究和优化。但目前各类采样规律都仍然不能完美地贴合人类的风格，且主流的采样策略中[15-17]，没有任何一种策略取得了明显的领先地位。

1.3.3 自然语言隐写的衡量指标

自然语言隐写需要权衡多项指标，包括不可感知性、安全性、嵌入量、嵌入时间复杂度和提取时间复杂度。下面对以上指标进行展开阐述。

1) 不可感知性：指隐写文本经过人视觉观察无法被发现是异常的。举例而言，如果一段文本显然不符合语法，则很容易引起怀疑，也便不符合不可感知性的要求。不可感知性指标主要依靠人的主观评价方法进行评估。

2) 安全性：指隐写文本在统计规律特性上与自然文本相近。相比于不可感知性更多是为了躲避人的视觉分析，安全性更多指躲避机器分析。然而，一个隐写算法在不可感知性和安全性上的性能表现往往成正比关系，若由一个隐写算法生成的隐写文本可以在统计规律上接近自然文本，则往往也同样代表了其不容易遭到视觉观察上的怀疑。下面给出了一些常见的机器分析指标：

a) BLEU：双语评估替换(Bilingual Evaluation Understudy, BLEU) [18]最初用于评估机器翻译任务，它统计了机器翻译所得的文本中的文字出现在标准答案文本中的频率，以此体现机器翻译所得文本与标准答案文本的相似度。之后 BLEU 在各类文本生成任务中得到广泛的应用，该指标可用于评估任意具有参考文本的生成文本质量，不再局限于文本翻译任务。

因为主流的生成式自然语言隐写方法摒弃了原始文本的约束，所以 BLEU 并非生成式自然语言隐写方法的主流评估指标。然而，本文所提出的算法是在原始文本的约束下进行生成式隐写，并且采用翻译任务作为框架底座，故本文中采用了 BLEU 指标验证所生成隐写文本与原始文本的相似度。

计算 BLEU 指标时，需将原始文本和隐写文本分别切分成多个长度为 l 的词组块，得原始文本词组块序列 $\boldsymbol{x}=\left(x_1,x_2,\cdots,x_m\right)$ 和隐写文本词组块序列 $\boldsymbol{y}=\left(y_1,y_2,\cdots,y_n\right)$。通常地，由于原始文本与隐写文本的长度不一致，词组块数量 $m\neq n$。为计算 BLEU 指标，首先需计算隐写文本与自然文本的重合比例 P_l，计算方式如公式(1.1)所示。

$$P_l=\frac{\min\left\{h_k\left(\boldsymbol{x}\right),h_k\left(\boldsymbol{y}\right)\right\}}{h_k\left(\boldsymbol{x}\right)} \tag{1.1}$$

其中，$h_k(\boldsymbol{x})$和$h_k(\boldsymbol{y})$分别代表第 k 个词组块出现在原始和隐写文本中的次数。

公式(1.1)存在这样一个不足：即当隐写文本长度较短时，与参考文本的匹配度较高。为避免由文本长度而导致的评分偏向性，BLEU 在最终的评分结果中引入了长度惩罚因子(Brevity Penalty, BP)，其计算方式如公式(1.2)所示。

$$\mathrm{BP}=\begin{cases}1, & l_x > l_y \\ e^{1-\frac{l_x}{l_y}}, & l_x \le l_y\end{cases} \tag{1.2}$$

其中l_x与l_y分别代表原始和隐写文本的长度。

BLEU 指标的最终计算公式如公式(1.3)所示：

$$\mathrm{BLEU}=\mathrm{BP}\times\exp\left(\sum_{l=1}^{N}\frac{1}{N}\log\left(P_l\right)\right) \tag{1.3}$$

其中，N 为 BLEU 指标的阶数，阶数越大代表 BLEU 更倾向于统计整体一致性。

b) BERTScore：相似地，BERTScore(Bidirectional Encoder Representation Transformer Score) [19]也是验证隐写文本与原始文本相似度的指标。不同的是，考虑到完全不同的文字也可以表达相似的含义，BERTScore 将语义空间的相似度纳入考虑范畴，而不局限于相同文字出现的频率。具体而言，BERT(Bidirectional Encoder Representation Transformer)可以将文本进行特征采集，并量化为语义空间中的高维向量，若语义空间中两向量的相似度高，则可判定两特征向量所代表的文本的语义相似度高。BERTScore 指标的计算公式如公式(1.4)所示。

$$\mathrm{BERTScore}=F_{\mathrm{BERT}}=2\times\frac{P_{\mathrm{BERT}}\times R_{\mathrm{BERT}}}{P_{\mathrm{BERT}}+R_{\mathrm{BERT}}} \tag{1.4}$$

其中，

$$R_{\mathrm{BERT}}=\frac{1}{m}\sum_{\boldsymbol{v}_{x_i}\in\boldsymbol{v}_x}\max_{\boldsymbol{v}_{y_j}\in\boldsymbol{v}_y}\boldsymbol{v}_{x_i}^{\mathrm{T}}\cdot\boldsymbol{v}_{y_j} \tag{1.5}$$

$$P_{\mathrm{BERT}}=\frac{1}{n}\sum_{\boldsymbol{v}_{y_j}\in\boldsymbol{v}_y}\max_{\boldsymbol{v}_{x_i}\in\boldsymbol{v}_x}\boldsymbol{v}_{x_i}^{\mathrm{T}}\cdot\boldsymbol{v}_{y_j} \tag{1.6}$$

其中，$\boldsymbol{v}_x=\left(\boldsymbol{v}_{x_1},\boldsymbol{v}_{x_2},\cdots,\boldsymbol{v}_{x_m}\right)$和$\boldsymbol{v}_y=\left(\boldsymbol{v}_{y_1},\boldsymbol{v}_{y_2},\cdots,\boldsymbol{v}_{y_n}\right)$为原始和隐写文本经词向量化后所得的向量序列，$m$ 和 n 为两序列的长度。

c) PPL：困惑度(Perplexity, PPL) [20]是使用语言模型对隐写文本进行统计分

析的指标，也即验证所提出隐写方法的安全性。隐写文本与原始文本的 PPL 值的差异可反映其统计规律上的相似度。根据公式(1.7)，PPL 值越低代表了隐写文本使用了条件转移概率较高的文字作为最终输出，也代表了隐写文本更为流畅。

$$\mathrm{PPL} = p\left(w_1 w_2 ... w_N\right)^{-\frac{1}{N}} = e^{-\frac{1}{N}\sum_{i=1}^{N}\log p\left(w_i | w_1 w_2 ... w_{i-1}\right)} \tag{1.7}$$

其中，N 为文本长度，w_i 代表文本的第 i 个词，$p\left(w_i \middle| w_1 w_2 ... w_{i-1}\right)$ 为生成 w_i 的条件转移概率。

d) 抗隐写分析能力：隐写分析是针对隐写的对抗技术，目的是检测文本中是否含有秘密信息。对于隐写技术而言，为实现隐蔽通信，应尽可能避免被隐写分析技术检测成功。所以，抗隐写分析能力也是重要的安全性指标之一。

隐写分析可以被视作一个二分类问题，即判断一段文本为自然文本或隐写文本。隐写分析的正确率越接近 50%，则代表了隐写方法的安全级别越高，即监测方无法判断文本属于哪一类，结果仅接近于随机猜测的结果。隐写分析实验通常存在两种不同的假设。第一种假设规定监测方无法获知隐写者使用的具体隐写方法，而第二种假设规定监测方完全掌握了隐写者使用的嵌入策略，但不能获取隐写者使用的密钥信息。在本文中，所有隐写分析实验均采用了第二种假设，并且假设监测方拥有发送方过往的原始-隐写文本数据对，并配以对应的标签，进而对测试集中的文本进行判断。实验表明，即使在对隐写者如此不利的假设下，本文所提出的自然语言隐写方法仍表现出较好的抗隐写分析能力。

在本文的所有实验中，“隐写分析正确率”被简写为“正确率”。尽管隐写分析实验是以监测方的角度进行实验，但该实验实际验证的是自然语言隐写方法的抗隐写分析能力和安全性。

3) 嵌入量：指隐写文本中携带的秘密信息的数量。计算机在处理所有信息时都需要将原始信息转化为二进制比特流，这也意味着二进制比特流可以代表任意的文字信息。所以，在自然语言隐写中，嵌入二进制比特的数量也就代表了嵌入量大小。为了排除文本长度对实验结果的影响，单位文字中嵌入比特数(Bit Per Word, BPW)和单位句子中嵌入比特数(Bit Per Sentence, BPS)是自然语言隐写中的两个常用指标。

BPW 是主要用于生成式自然语言隐写方法的衡量嵌入量的指标。生成式自

然语言隐写方法在逐字生成文本的过程中，可在每一个生成步骤中嵌入秘密信息。由于生成式自然语言隐写方法是将一个词作为一个嵌入单元，所以 BPW 是更为合适的指标。BPS 是主要用于衡量句子级修改式自然语言隐写方法嵌入量的指标，不同于生成式自然语言隐写方法以词为单元嵌入秘密信息，句子级修改式自然语言隐写方法以改变整个句子的表达方式嵌入固定的比特数量，是将整个句子作为一个嵌入单元，所以 BPS 是其更为合适的指标。

由于本文提出的算法是以生成式的方式改变了整个句子的表达方式，本文将其定义为生成式方法，但其实算法与句子级修改式自然语言隐写方法也十分相似。为更好地说明本文所提出的方法在嵌入量上的优越性，本文也将所提出的生成式方法与句子级修改式方法进行对比。为统一指标，BPW 和 BPS 可作如下转化：

$$\mathrm{BPS}=\left[\mathrm{BPW}\times L\right] \tag{1.8}$$

其中，L 为对应的生成式自然语言隐写方法所得的隐写文本的长度，[•]代表四舍五入取整算法。

4) 嵌入时间复杂度和提取时间复杂度：从实际应用的角度出发，嵌入时间复杂度和提取时间复杂度也是两个重要指标。为了更好地模拟人类用户的行为，算法的时间复杂度应尽可能降低。对于发送方而言，嵌入时间复杂度主要分为两个部分：辅助信息构建复杂度和实际嵌入复杂度。而对于接收方而言，提取时间复杂度通常与隐蔽通信双方所需共享的辅助信息量成正比。所需共享的辅助信息越多，代表了提取步骤越复杂，提取秘密信息所需要的时间也越长。

1.4 论文的主要研究内容

1.4.1 主要研究成果

本文研究了语义可控的自然语言隐写方法，针对修改式方法嵌入量小，生成式方法语义不可控的不足提出改进。本文提出的隐写框架通过语言转换的方式，以生成式的方法实现了语义可控，使得基于该框架的隐写算法同时具备语义一致性和高嵌入效率的优势。此外，本文还设计了一种适配该框架的新型生成式隐写编码，使得发送方与接收方所需要共享的辅助信息少于主流生成式隐写编码的要

求。该编码将同义词概念引入隐写编码算法的设计中，解决了在隐写文本生成过程中潜在的劣势选择问题。论文具体内容如下：

1) 提出了一种基于释义技术的自然语言隐写算法。通过分析自然语言隐写方法现状与两类主流方法的特性，设计了一种能够在保证语义一致性的同时，提升自然语言隐写嵌入量的新型隐写框架。与当前自然语言隐写倾向于降低隐写文本与自然文本的统计分布差异的趋势不同，本文通过缩小隐写文本与原始文本语义层面的差异以实现安全性。该隐写框架利用释义技术作为修改式与生成式方法结合的桥梁，并以翻译任务作为释义技术的模型底座。实验表明，基于该框架的自然语言隐写算法所得的隐写文本在语义一致性指标上领先于主流生成式方法，同时，相比修改式方法，该算法在嵌入量上也有明显提升。

2) 提出了一种基于语义感知编码的自然语言隐写算法。当前生成式隐写编码主要采用的编码方式通常可以保证较高的隐写文本质量，但其代价是计算复杂度更高，且隐蔽通信双方需共享更多的辅助信息。随着深度学习模型的不断发展，模型复杂度和参数量持续增大，放大了该类隐写编码的不足。于是，本文设计了一种新型的隐写编码，在保证隐写文本质量的同时，尽可能减少接收方所需的辅助信息。该隐写编码利用同义词关系，将语义相近的词汇均匀地映射到不同二进制比特流上，以提升在文本生成步骤中存在合适语义的词汇映射到秘密信息上的可能性。基于该编码策略的隐写方法可使得接收方仅需通过查询而无需计算提取秘密信息，降低了提取秘密信息的复杂度。

1.4.2 论文结构安排

本文各章内容安排如下：

第一章介绍了自然语言隐写的基本概念及意义，引出了本文的研究目的和研究价值；并通过与图像隐写对比，分析了自然语言隐写的难点；然后介绍了自然语言隐写的主要评估指标。

第二章介绍了主流的自然语言隐写方法，以代表性研究为例，分析了各类方法的优势及不足，并介绍了目前自然语言隐写研究的发展趋势，引出了“语义可控”在生成式自然语言隐写研究的缺失和研究价值。

第三章提出了基于释义技术的自然语言隐写方法。该章节重点介绍了以释义技术为底座的新型语义可控生成隐写框架，该框架以文本生成的方式保证了语义一致性。实验表明，基于该框架设计的算法在安全性和嵌入量等指标上领先主流自然语言隐写方法，验证了算法的有效性。

第四章介绍了基于语义感知编码的自然语言隐写方法。该章节重点介绍了该编码技术设计的背景与目的，阐述了该编码技术的算法细节，并进行了实验分析。实验验证了该算法在复杂度较低的前提下，提升了隐写文本的总体质量。

第五章对本文研究内容进行总结，并对下一步研究进行展望。

第二章　自然语言隐写研究概况

自然语言隐写方法主要分为三类，即修改式自然语言隐写、选择式自然语言隐写和生成式自然语言隐写。其中，修改式自然语言隐写方法最为常见，该方法通过对自然文本进行修改的方式嵌入秘密信息。为了更高的隐写文本质量，选择式隐写方法完全保留了自然文本，而通过在数据库中建立映射关系的方式进行隐蔽通信。随着深度学习的兴起，文本生成模型愈发成熟，通过直接生成隐写文本嵌入秘密信息的生成式自然语言隐写方法成为了当下主流的隐写方法。图 2.1 展示了本章的介绍框架，对于修改式自然语言隐写方法，本章根据具体做法对修改式方法进行了细分。对于生成式自然语言隐写方法，本章根据隐写步骤分成了文本生成框架和隐写编码两个部分，分别与第三章与第四章的研究成果所对应。值得注意的是，文本生成框架和隐写编码都是生成式自然语言隐写的重要部分，两者密不可分且互相影响。

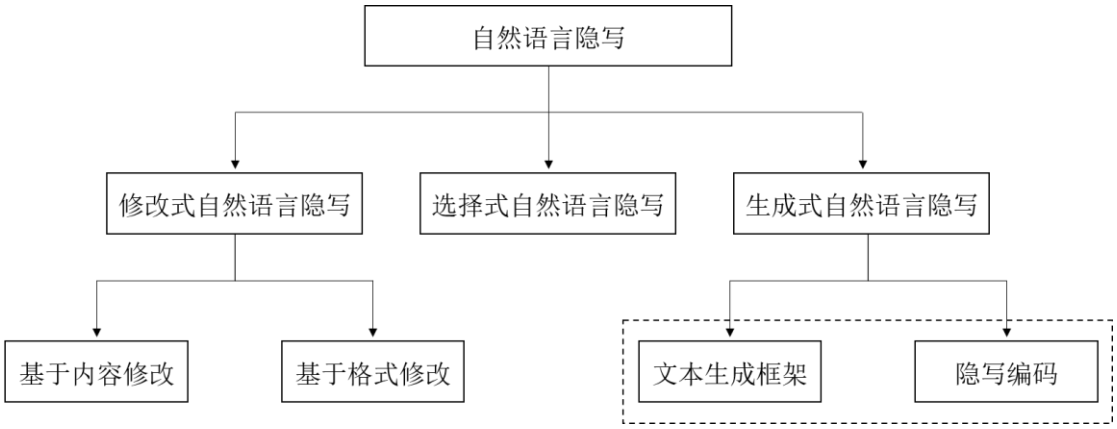

图 2.1　自然语言隐写研究概况介绍框架

2.1 修改式自然语言隐写

修改式自然语言隐写通过对原始文本进行修改实现秘密信息的嵌入。根据修改的对象不同，主要分为两类：基于内容的修改式自然语言隐写和基于格式的修改式自然语言隐写。前者是通过对文本内容的修改嵌入秘密信息，后者则是保留

了原始文本的全部文字，而通过对格式进行修改嵌入秘密信息。修改式方法主要分为两步：第一步构建候选池，第二步对候选池中的每个元素进行编码，从而实现文字向二进制比特的映射。

第二步具体编码方式可采用二叉树编码[21]、矩阵编码[22]、图编码[23]等，这些编码方式可作用于大多数隐写方法，在本章中不多做介绍。本章将着重介绍各类研究方法独特的候选池构造方式。

2.1.1 基于内容的修改式自然语言隐写

基于内容的修改式自然语言隐写方法将原始文本的部分或全部内容替换为携带秘密信息的另一内容，以保证所得的隐写文本和原始文本的语义一致。该方法可根据替换的对象进一步分为词级修改式隐写和句子级修改式隐写方法。前者通过一段文本的一个或几个词嵌入秘密信息，而后者是替换了整段文本。

1) 基于同义词替换的词级修改式隐写方法：同义词替换是词级修改式隐写方法中的一种基本方法。根据 1.3.2 节所述，图像隐写可以对像素值进行±1 进行秘密信息的嵌入，本质是将修改的对象改变至一个相似的对象。类似地，对于自然语言隐写，可以通过引入同义词概念，实现词级修改式自然语言隐写[24-27]。发送方将原始文本中的特定词替换为根据自己所希望发送的秘密信息所对应的同义词，接收方通过共享的同义词词典和映射规则可从隐写文本中提取秘密信息。尽管使用同义词已在一定程度上保证了语义的一致性和连贯性，但由于文字的语义多样性、语法和固定搭配等特性，同义词替换仍难保证隐写文本的质量，存在潜在的安全隐患[28]。

一词多义问题如图 2.2 所示。“court”可译为“法庭”或“球场”，所以“court”会出现在多个同义词集合中。而且不同的同义词集合对“court”编码不同，甚至因为同义词集合大小不同，导致不同集合中“court”映射到的比特长度也不一致，这使得接收方在获取隐写文本后感到困惑，从而无法准确地提取秘密信息。

对于语法问题，以英文文本举例，若将“He runs fast”作为原始文本，并期望将单词“fast”替换为同义词而嵌入秘密信息。“quick”是“fast”的同义词，然而，“fast”是一个形容词与副词同型的英文单词。“He runs fast”中“fast”表

现为副词，而“quick”却是一个形容词。所以尽管“quick”是“fast”的同义词，但若直接进行替换，则隐写文本“He runs quick”不合语法，从而容易引起监测方的怀疑。对于固定搭配问题，以中文文本举例，若将“这是一件好事”作为原始文本，尽管“棒”可以作为“好”的同义词，且词性都为形容词，但“这是一件棒事”并非人们习惯的表达方式，也同样容易引起人们的怀疑。

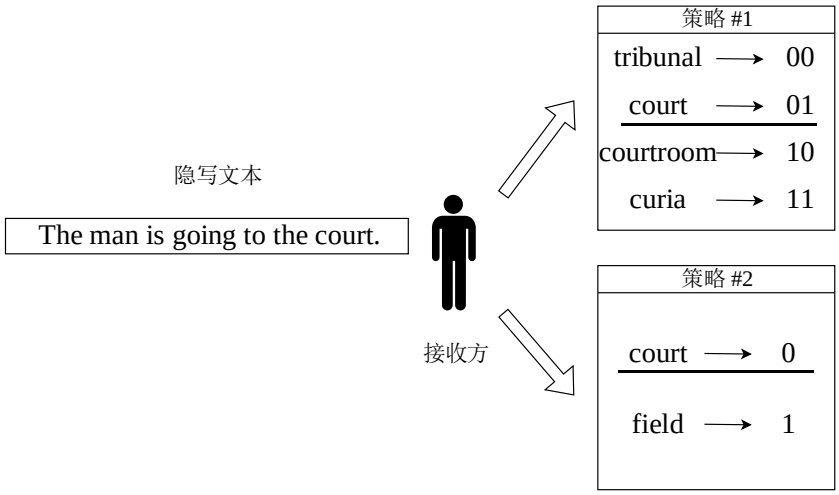

图 2.2　一词多义对基于同义词替换的隐写方法的影响

为更高的隐写文本质量并保证隐写成功率，研究者只能更加精细地设计同义词词典，例如根据句法搭配[29]、单词变换[30]或引入语义关联度量[31]缩减候选词数量。真正严格满足替换规则的同义词数量远少于如 WordNet[32]等原始词典中的同义词数量。如图 2.2 所示，同义词的数量决定了最大的嵌入容量，减少合格同义词的数量使得基于同义词规则的词级修改式自然语言隐写的嵌入量较小。

2) 基于预训练模型的词级修改式隐写方法：由于使用同义词规则难以满足整段文本语义和语法的准确性，随着深度学习的发展，词级修改式自然语言隐写又演化出了基于预训练语言模型修改的分支。

语言模型是一种借助机器学习算法具备计算给定文本中所有词汇的概率分布能力的模型。对于任意的一段文本 $\boldsymbol{x}=\left(x_1,x_2,\cdots,x_N\right)$，其中 N 代表文本的长度，一个语言模型可计算文本中每个词的条件转移概率，整段文本的联合概率为每个词的条件转移概率之积。

预训练语言模型指的是一个语言模型经过大量自然文本的训练具备了处理自然语言问题的基础知识的模型。对预训练语言模型进行针对性专项问题的少量训练之后，就可以在下游任务中具有良好的性能[33-35]。预训练语言模型大致分为两类：填空式和续写式。两种语言模型对条件转移概率的定义不同使得它们在隐写中的应用不同。填空式的语言模型由于其在计算条件转移概率时具有后续的所有文字信息，该模型常用于词级修改式自然语言隐写，而续写式模型在计算条件转移概率时无法获取后续文字，所以该模型更符合一般文本生成的规律，可用于生成式自然语言隐写。填空式语言模型对条件转移概率的定义如公式(2.1)所示，续写式语言模型对条件转移概率的定义将在后文介绍生成式隐写方法时给出。

$$p\left(x_1, x_2, \cdots, x_N\right)=\prod_{i=1}^{N} p\left(x_i \mid x_N, \cdots, x_{i+1}, x_{i-1}, \cdots, x_1\right) \tag{2.1}$$

BERT[36]是填空式预训练语言模型的代表模型。它的主要训练方式之一是将自然文本中的部分词作掩码处理，通过大量训练使得 BERT 可以根据上下文关系恢复出被掩码的词汇。同样地，BERT 在掩码位置的输出也为一个概率分布，训练期望标准答案所占的概率尽可能大，尽管理想中标准答案所占的概率比重应为100%，但实际上无法达成。这是因为符合语境含义的词汇极有可能不是唯一的，一些适合放置在掩码位置的词汇也将占据一定的比重。这便为自然语言隐写提供了修改和嵌入秘密信息的空间。

词级修改式自然语言隐写利用这一特性设计了基于预训练语言模型的替换规则[37]，即掩码预测后使用条件转移概率超过一定门限的词构成候选池。之后的研究者通过吉布斯采样的方式优化了多掩码位置的填充隐写算法[38,39]。基于预训练语言模型的词级修改式自然语言隐写要求接收方需要获得与发送方相同的语言模型计算掩码位置的条件转移概率分布，增加了需要共享的辅助信息。

由于大量自然文本的训练，预训练语言模型可以学习到自然文本语法相关的基础知识，所以通过掩码策略替换规则得到的隐写文本基本满足语法和流畅性的要求。但是，仅根据条件转移概率嵌入秘密信息也有其局限性：预训练语言模型只能保证其语言的流畅性，而不能保证语义准确性。如图2.3 所示，若为“[MASK]”位置选择一个候选词，则发现有大量的候选词具有较高的条件转移概率，这是因

为根据语境，“[MASK]”位置选择一个与职业相关的词汇为优选。仅对该段文本而言，任意与职业相关的词汇都合适，但考虑到一个人可能从事的职业是与其背景有关的，若不考虑具体人的属性或更大范围的语境意义，则有可能导致隐写文本与实际情况不符，这正是因为基于预训练模型修改的方法无法实现语义可控。

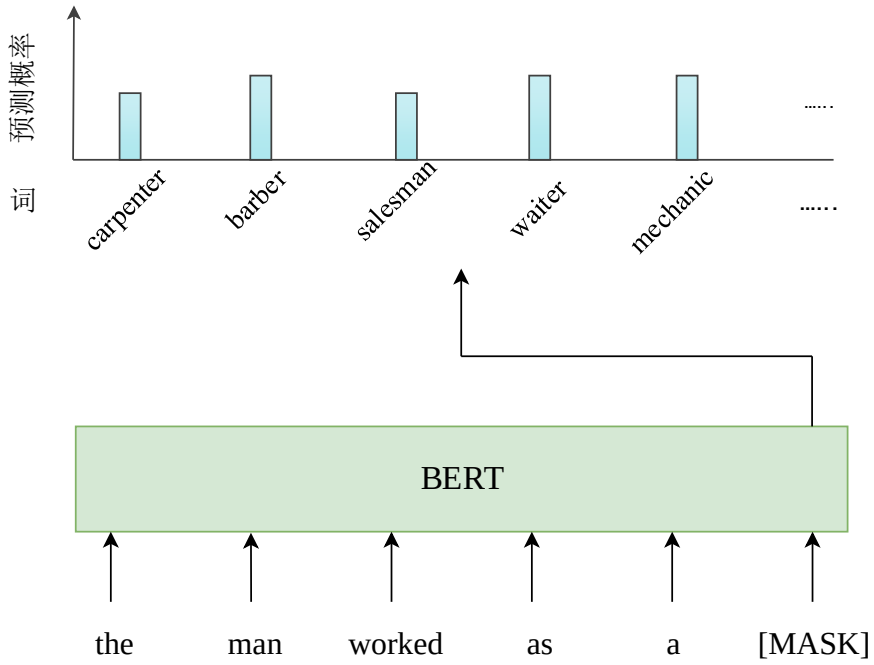

图 2.3　基于预训练模型的词级自然语言隐写

3) 句子级修改式隐写方法：相比于词级修改式方法，句子级修改式自然语言隐写以整个句子为单位，而不是仅对句子的部分内容作修改。这样的做法避免了因部分修改而对句子整体所带来的突兀感。然而，对于词级修改式方法而言，单一文本往往有多个可修改的词，也意味着有多个可嵌入秘密信息的单元，而句子级修改式方法在单一文本中仅有一个可嵌入单元，这使得句子级修改式方法的嵌入容量小于词级修改式的方法。

与基于同义词替换的词级自然语言隐写方法相似，句子级修改式方法也可通过同义句替换的方式嵌入秘密信息。Murphy 等人[40]通过语法分析的方式，对原始文本中的从句部分进行同义替换；Liu 等人[41]使用神经网络提取句法结构树，并进行句法结构同等变换；Chang 等人[42]对文本中的词序重排实现同义句替换。

Alex 等人[43]利用大规模同义短语库重新构建文本；Kermanidis 等人[44]利用复述技术对同等语义的语句进行重写。

总体而言，句子级修改式自然语言隐写方法优势在于该类方法对句子整体进行修改，因此保证了句子整体的语义连贯性，但不足在于该类方法将整个句子作为一个修改单元，相比词级修改式的隐写方法嵌入容量更小。

2.1.2 基于格式的修改式自然语言隐写

通过对文本内容的修改嵌入秘密信息的方式总是不可避免地在一定程度上改变原始文本的语义。基于格式的修改式方法完全保留原始文本中的文字，而通过对原始文本的格式进行修改的方式实现秘密信息的嵌入。该类方法旨在通过完全保留原始的文字内容以实现不可感知性和安全性。由于文本格式与文档属性紧密相关，基于格式的自然语言隐写方法对使用场景的要求更高。

以 Microsoft Office 为例，对于文字属性，可通过对文档中文字的颜色[45]、大小[46]、字体[47]等格式属性嵌入秘密信息；对于段落属性，可通过调整行间距、字间距[48]或增加不可见字符[49]嵌入秘密信息；也可通过文档的一些特殊功能嵌入秘密信息，例如备注功能[50]。

基于格式修改的隐写方法难以通过语义分析识别，但这类方法也有一定的局限性。首先，该类方法仍然向原始文本引入了手工改动，尽管具有一定的不可感知性，但如果监测方仔细审查，仍能从视觉的角度发现异常；其次，依赖于文本格式的隐写方法对文档有更高的约束要求，文档内容若被转移或被其他方式破坏了文档格式，则秘密信息无法准确提取；同时，该类方法仅考虑了语义层面的安全性，使得容易被基于统计分析的检测方法所识别[51, 52]。

2.2 选择式自然语言隐写

修改式自然语言隐写方法通常对自然文本进行修改，而所得修改后的隐写文本的文本质量总会受到不同程度的影响。而选择式隐写方法则避免了这一问题。选择式隐写方法需要发送方与接收方共享一个大型数据库和一套编码规则。发送方只需选择一组可以映射到自己希望发送的秘密信息的元素给接收方，即可实现

秘密信息的传递。

例如，Zhang 等人[53]通过语料库中的词频特性建立映射关系；Long 等人[54]根据语义空间的距离对文本进行编码；Wang 等人[55]根据字符结构构建检索树；Hu 等人[56]通过图文结合的数据库共同实现信息隐藏。

选择式自然语言隐写方法的优势在于发送的载体未经任何修改，仍然是自然的，载体本身难以被监测方识别。然而其劣势在于发送的内容仅限于一个数据库中，若监测方发现传递的消息种类有限，也容易遭到怀疑。所以发送方与接收方不得不共享一个大型的数据库，数量庞大的辅助信息使得该类方法在预先沟通阶段的难度大幅上升。此外，选择式方法不受原始文本约束，仅受数据库约束不能实现精确的语义控制，这使得选择式隐写方法在使用场景上也具有局限性。

2.3 生成式自然语言隐写

生成式自然语言隐写同样不依赖于原始文本，与选择式方法查找一段自然文本不同的是，生成式方法直接生成一段全新的含密文本。随着语言模型在自然语言处理领域的不断进步，使用语言模型生成的文本的连贯性和流畅度都得到了保障。由于生成的文本已经可以接近自然文本的语法规范和流畅性，且生成全新的文本所需的时间少于使用选择式方法从数据库中搜索的时间，所以生成式自然语言隐写方法也可视为对选择式方法的一种改进。

由于文本生成的特性，生成式自然语言隐写方法可在每一个生成步骤中向任意词嵌入秘密信息。不同于修改式中依赖同义词或同义句的替换规则，嵌入要求的降低使得生成式自然语言隐写方法的隐写容量大幅提升。由生成式方法得到的隐写文本的嵌入容量可以达到由其他类型隐写方法所得的隐写文本嵌入容量的数倍。由于生成式自然语言隐写方法的嵌入容量大，相比于其他两类方法对于隐写的前置要求更低，并且由这类方法所得的隐写文本质量普遍较高，所以生成式自然语言隐写已成为近年来自然语言隐写的主要研究方向。

生成式自然语言隐写主要分为两个步骤：第一步确定文本生成框架，第二步建立候选词与秘密信息之间的映射关系，也即编码方式。下面将对这两个步骤分别展开阐述。

2.3.1 生成式自然语言隐写框架

生成式自然语言隐写框架指隐写文本生成框架。实际上，任意的文本生成模型都可用于生成式自然语言隐写模型。尽管隐写框架并不与秘密信息的嵌入产生直接联系，但隐写框架仍然是决定隐写文本质量的一个重要因素。生成式自然语言隐写框架之于生成式隐写方法的重要性可类比于替换规则之于修改式隐写方法的重要性和大型语料库之于选择式方法的重要性。

自回归语言模型常作为生成式自然语言隐写框架的主要模型类型，区别于公式(2.1)，自回归语言模型对条件转移概率的定义如公式(2.2)所示，该定义方式更符合人们撰写文本的习惯，即从左至右依次生成文字。

$$p\left(x_1, x_2, \cdots, x_N\right)=\prod_{i=1}^{N} p\left(x_i \mid x_{i-1}, x_{i-2}, \cdots, x_1\right) \tag{2.2}$$

Alfonso 等人[57]首先提出了使用统计自回归语言模型自动生成隐写文本。该研究收集了大量自然文本，并将数据集中每段文本根据 N-gram 模型切割，将切割后的 N-gram 短语作为一个编码单元，根据秘密信息选择最合适的短语输出。随着 N-gram 数据集质量的提升，之后 Chang 等人[58]进一步利用 Google 构建的 Google N-gram 数据集[59]指导文本生成。

随着文本生成技术的进步，以更高细粒度的文本分割方法取代了 N-gram 分割方法，并将单词取代短语作为一个编码单元。Moraldo 等人[60]引入马尔科夫模型指导隐写文本的生成，该研究使用马尔科夫模型提炼大量自然文本中的低阶转移矩阵，并根据转移矩阵对词典中的各个词汇进行编码。深度学习和神经网络的发展使得生成式自然语言隐写框架的性能更进一步，Yang 等人[61]通过 RNN 模型自动生成隐写文本，选取条件概率较大的词汇构建候选池和映射关系。由于任意的语言模型都可作用于生成式自然语言隐写，大量研究表明使用更先进的自然语言处理技术便可得到质量更高的隐写文本。

然而，生成式自然语言隐写方法也有其弊端，即由生成式自然语言方法所得的隐写文本在统计规律上与自然文本差异较大，使得隐写文本更容易被检测。于是，研究者从统计规律的角度对文本生成模型进行改进。Yang 等人提出了基于变分自编码器(Variational Autoencoder, VAE)[62]和对抗生成网络(Generative

Adversarial Nets, GAN)[63]的隐写框架，在文本生成的步骤中增加统计规律上的约束条件，以进一步保证隐写系统的安全性。实验表明，对隐写文本生成框架增设统计规律的约束条件可有效提升生成式自然语言隐写算法的抗隐写分析性能。

除了统计差异以外，语义不可控是生成式自然语言隐写方法的另一主要不足。相比于修改式方法具备原始文本的限制条件，若生成式方法仅使用简单的文本生成模型而不设置任何约束条件的限制，隐写文本的语义将不受控制。语义不可控的不足使得生成式自然语言隐写方法的应用场景受限。于是，大量研究者尝试了通过改进隐写框架的方式控制语义。

Kang 等人[64]提出了一种通过关键词限制文本语义的生成方式，以实现主题控制的效果；Yang 等人通过将语言模型扩展至 Seq2Seq 结构，以实现对话系统下的生成式隐写[65]，使得回复的文本与历史对话记录所匹配；又尝试在编码端输入知识图谱，以控制文本的主题内容[66]；薛一鸣等人[67]提出了图像描述场景下的生成式自然语言隐写，将图像作为约束条件，生成图像描述文本作为隐写文本。总体而言，这些方法在限制主题内容上取得了一定的进展，但是仍然无法精确地控制语义，仍不能从根本上解决语义不可控对生成式隐写方法使用场景的限制。

2.3.2 生成式自然语言隐写编码

隐写框架对生成式自然语言隐写至关重要，但其本身并不涉及秘密信息的嵌入，而隐写编码则与秘密信息嵌入直接相关。隐写编码是构建文本与秘密信息关系的方式。对于生成式自然语言隐写而言，目前主流的方式是将秘密信息分为多段，并在每一个生成步骤中嵌入部分秘密信息。所以生成式自然语言隐写编码实际构造的是候选词与长度较短的二进制比特流之间的映射关系。

Fang 等人[68]提出了桶编码策略。如图 2.4 所示，桶编码策略将词典中的所有词平均分配到 2^l 个桶中，每一个桶中的所有词汇都映射到长度为 l 的同一比特流，图中 l 取 2。对于每一个隐写步骤，发送方只需从秘密信息所对应的桶中选择输出词。为保证隐写文本的质量，发送方通常选择秘密信息所映射到的桶中具有最大条件转移概率的词汇作为最终输出。这样的做法优势在于编码方式不随文本生成过程而改变，可在文本生成之前完成设置，且接收方提取秘密信息步骤简单。

本文将此类编码方式称为静态编码。然而桶编码策略也有明显劣势，由于桶编码仅依靠随机种子设计映射关系，桶的质量不可控，而若随机过程导致发送方使用了一个质量较低的桶，则会导致隐写文本的质量下降。

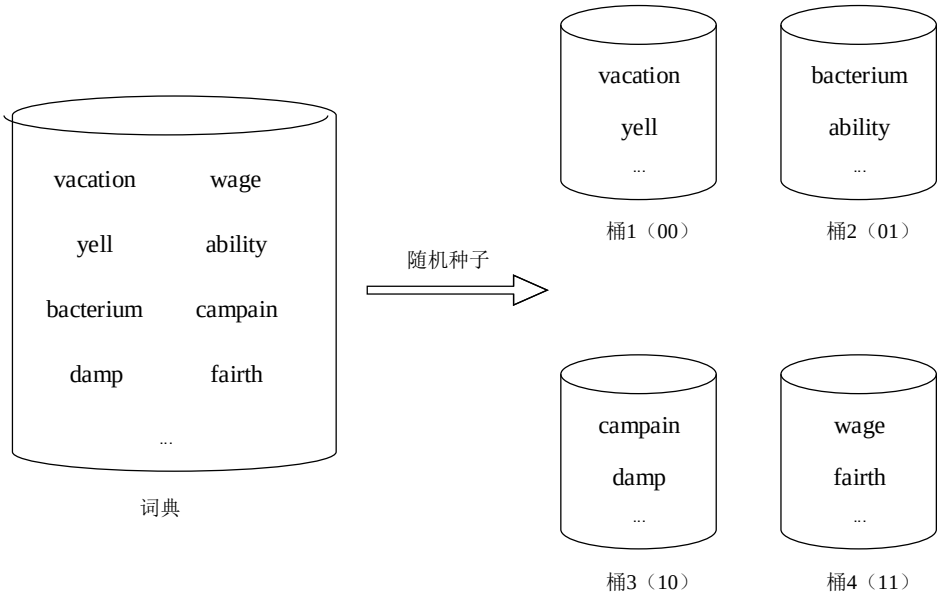

图 2.4　桶编码示意图

Yang 等人[61]使用块编码对文本质量进行改进，对于每一个隐写步骤，每次重新根据条件转移概率前 2^l 的词汇构造候选池，使得每个候选词都携带长度为 l 的二进制比特流。由于每个生成步骤选择的候选词都为该步骤中条件转移概率较高的词，所以可以保证文本质量较高。这类在每一个生成步骤中重新构建映射关系的编码方式在本文中被称为动态编码。

块编码使得所有候选词以相等的概率作为最终的输出，存在进一步优化的空间。尽管候选词的总体质量较高，但仍有高低的区别。语言模型提供了直接的量化指标，所占条件转移概率较高的词汇便是模型认定的更高质量的候选词，所以将这些质量不等的词以相等的概率映射到秘密信息是不合适的。于是研究者们尝试使用可变长的编码方式使得条件转移概率更高的词汇更有可能被选择[69, 70]。

除此之外，Dai 等人[71]提出，若在隐写文本生成过程中总是倾向于使用条件转移概率较高的词汇反而会导致与自然文本的分布产生较大差异。这与文本生成

领域的研究恰好吻合，即自然文本也常包含语言模型显示为低概率的词汇。Dai等人利用KL散度（Kullback-Leibler Divergence）对分布差异进行量化，若KL散度大于预设的阈值，则在该生成步骤将不会嵌入秘密信息，以常规的文本生成的采样策略选择输出词，并将所需嵌入的信息推迟到下一个生成步骤处理。实验表明，这种策略有效提高了自然语言隐写方法的抗隐写分析的能力。

相比于静态编码策略，目前生成式自然语言隐写算法更倾向于使用动态编码。动态编码的确提升了隐写文本的质量，但增加了算法的复杂度。由于每个生成步骤都重新设置映射关系，对于接收方而言，需要通过还原发送方生成文本的全过程提取秘密信息，这大幅增加了接收方提取秘密信息的复杂度。同时，为了接收方可以计算出与发送方一致的条件转移概率，双方必须共享相同的语言模型。语言模型的结构往往十分复杂[72]，将语言模型作为隐蔽通信所需共享的辅助信息将增加隐写过程的成本。总体而言，静态编码与动态编码各有利弊，两种编码方式应受到同等重视。

2.4 本章小结

本章介绍了修改式、选择式和生成式自然语言隐写的概念和代表算法，并分析了各类自然语言隐写方法的优势和不足：修改式方法对自然文本的影响较小，但嵌入容量小；选择式方法不对文本进行修改，文本质量较高，但方法受到发送方与接收方需要共享的数据库的限制；生成式方法嵌入量较大，但由于摆脱了原始文本的约束，造成了使用场景受限。此外，本章还分析了生成式自然语言隐写方法作为目前自然语言隐写领域的主流方法，在隐写框架上存在对语义可控方向研究的缺失，在隐写编码上缺少对静态编码方法的进一步探索，从而引出了本文第三、第四章中针对以上两点的研究。

第三章　基于释义技术的自然语言隐写

3.1 引言

以文本为载体的隐写方法最初以修改式方法为主流，其通过对文本部分或整体的修改，在不明显影响语义的前提下嵌入秘密信息。由于隐写文本与原始文本语义相似，所以认为隐写文本接近自然文本，是一种隐蔽的嵌入方式。然而，这类方法依赖完善的替换规则。例如，同义词替换可以认为是一种对原始文本影响较小的替换规则，但由于文字存在一词多义等特性，容易引起接收方的误解，从而无法正确地提取秘密信息，所以并不是一个完善的规则。对于修改式自然语言隐写方法而言，一个完善的规则必然有严格的条件，从而使得替换的可能性减少，也导致了嵌入容量降低。

为了解决修改式隐写方法嵌入容量小的不足，现有的生成式隐写方法对自然语言隐写基础框架进行了修改。发送方将直接使用文本生成模型生成一段全新的含密文本。与修改式方法最大的不同在于，生成式方法不再受原始文本的约束。由于深度学习和神经网络的发展，文本生成模型的性能不断提升，足以生成高质量的文本，所以隐写文本也可看作是自然的，故认为这类方法也是隐蔽的。生成式方法提升了隐写容量，是目前主流的方法。

对比可知，目前主流的生成式方法没有考虑隐写文本的语义一致性，其生成过程完全不受原始文本的约束。这使得生成式方法在使用场景上受到限制，因为并非任意文本都适合在任意语境和背景下出现。以“博客”的个人介绍场景为例，在这一项存在特定要求的使用场景中，若希望实现隐蔽通信，则需要输出符合人物特性的介绍语句。若隐写文本的内容不符合人物的特性，或完全与人物介绍无关，则容易引起监测方的怀疑。尽管目前有许多关于提升生成式自然语言隐写方法的安全性的研究，但这些研究倾向于缩小隐写文本和自然文本的统计分布差异，而缺少关于语义控制的相关研究。

修改式自然语言隐写方法语义可控，但嵌入容量小；生成式方法嵌入容量大，却不能控制语义。通过分析两类自然语言隐写方法的特性，促成了本章的研究，

即基于释义技术的自然语言隐写。该研究旨在同时满足较大的嵌入容量和语义的高度一致性。

3.2 相关技术简介

3.2.1 Seq2Seq 模型

Seq2Seq 模型，即 Sequence-to-Sequence 模型，是用于将一个序列映射至另一个序列的一类模型。由于一段文本可以视作一个序列，所以 Seq2Seq 模型常用于文本生成任务。输入一段序列作为约束条件，通过 Seq2Seq 模型可生成一段新序列。一个 Seq2Seq 模型由一个编码器和一个解码器共同组成，编码器对输入序列 $\boldsymbol{x}=\left(x_1,x_2,\cdots,x_{L_x}\right)$ 进行特征提取，其中 L_x 为输入序列的长度，x_i 为输入序列中的第 i 个元素。依据编码器所提取的特征，解码器逐步生成输出序列。解码器一次仅处理输出序列中的一个元素，对于第 t 个生成步骤，编码器仅输出一个概率分布 $\boldsymbol{p_t}=\left(p_{t,1},p_{t,2},\cdots,p_{t,m}\right)$，其中 m 为所有可能生成的元素的数量。以文本生成任务为例，m 即为词典的大小。$p_{t,i}$ 代表在生成文本中第 t 个元素时，词典中所有可能被选择的元素中的第 i 个元素作为最终输出的概率，概率值越大代表该元素更加适合作为该步骤的最终输出元素，但最终输出元素 y_t 需由采样策略决定。将所有元素拼接后，可得完整的输出序列 $\boldsymbol{y}=\left(y_1,y_2,\cdots,y_{L_y}\right)$。通常地，$L_x\neq L_y$，即输入序列和输出序列的长度通常不一致。一个 Seq2Seq 模型可以由各类的神经网络结构组成，如 LSTM[73]。

3.2.2 翻译技术

翻译是自然语言处理中的基础任务之一。翻译的目的是将一种语言的文本转化至另一种语言，并保持两个不同语言的文本语义一致。传统翻译技术主要运用了统计机器翻译系统或人工制定的语言规则[74, 75]。随着深度学习的发展，目前主流的翻译技术通常以神经网络模型为底座[76, 77]。翻译任务是适合使用 Seq2Seq 模

型的任务之一，以中译英为例，中文文本即为 Seq2Seq 模型中的输入序列，而英文文本是 Seq2Seq 的输出序列。

3.2.3 释义技术

释义技术将给定文本转化至同一语言下的另一文本，并保持两段文本语义一致。由于目前的自然语言处理问题主要依靠深度学习算法实现，而基于深度学习的各类算法都需要大型数据集以供训练模型，释义技术常用作自然语言处理的数据增强技术，以扩充训练数据量，从而提升各类方法在自然语言处理基础任务上的表现，例如信息提取和问答任务[78, 79]。

释义技术与自然语言隐写也密切相关。得到一段与原始文本语义一致而表达略有不同的文本是释义技术和修改式自然语言隐写方法的共同目标。同时由于释义生成技术以文本生成技术为底座，所以释义技术同时具有修改式和生成式自然语言隐写方法的特性，可以视作为两者之间的桥梁，这也启迪了本章中的研究。然而，大多数释义技术建立在特殊数据集的基础上，例如提问和图片描述[80, 81]。为了更广阔的应用场景，本章使用了一种自监督的释义生成模型作为语义可控生成隐写框架，这使得任意文本都可以作为隐写框架中的原始文本。

3.2.4 桶编码

Fang 等人[68]提出的桶编码策略是首个适用于生成式自然语言隐写方法的编码策略。该策略将词典中的所有词均匀分配到 2^l 个桶中，每个词都映射到唯一确定的一个桶，而每个桶对应一段二进制比特流。该方法的映射关系不随文本生成过程改变，便于接收方提取秘密信息。

然而，后续的生成式自然语言隐写编码研究倾向于使用动态编码的方式。动态编码根据文本生成过程中的概率分布动态地更新映射关系，由于动态编码只使用每个生成步骤中条件转移概率较大的词汇组成候选池，其有效地提升了基于动态编码的自然语言隐写方法所得的隐写文本质量。但是动态编码的代价是通信双方需要共享文本生成模型，且接收方需要通过逐词计算提取秘密信息。随着自然语言处理模型不断增大，使用动态编码的代价也将不断提高。

本章提出的框架理论上可作用于各类生成式自然语言隐写编码技术。然而，由于本章设计的语义可控生成框架相比于主流生成式方法更加复杂，若使用动态编码方式，则通信双方需要共享的内容过多。所以在本章提出的框架下，静态编码是更适合的选择。

3.2.5 Transformer 模型

Transformer 是自然语言处理领域最著名的模型之一，也是本章方案设计中采用的基础模型。尽管本章内容着重于整体语义可控生成框架，而非底座模型本身。但为在后文中更好地阐述，本章将简要介绍 Transformer 模型的核心模块，即注意力机制。

如图 3.1 所示，Transformer 也属于 Seq2Seq 结构，整体模型的左侧为编码端，而右侧为解码端。不同的是，Transformer 利用注意力机制，通过矩阵运算的方式实现了并行计算。所谓注意力机制，即是用数值量化文本中两个词汇的相关性。举例而言，“the students have done their homework”一句中，“students”、“have”和“their”应是高度相关的，他们以名词复数形式为纽带互相关联。同时“have”和“done”也因一般现在时的语法结构而高度相关。一段文本中任意两个不同的词汇总会产生不同程度的关联，而注意力机制正是利用这一种关系构造了一种新的运算模式。

注意力机制的引入使得 Transformer 的计算效率大幅提升，同时也使该模型具备分析文本中距离较远的词汇间关系的能力。下面将展开介绍图 3.1 中编码端的多头注意力机制（Multi-Head Attention）的计算方式，图中其余注意力机制原理也与之相似，不再展开进行阐述。

第一步：输入向量化后文本 $\boldsymbol{v}_x=\left(\boldsymbol{v}_{x_1},\boldsymbol{v}_{x_2},\cdots,\boldsymbol{v}_{x_N}\right)$，其中 N 为输入的长度。

第二步：为每一个输入单元初始化三个不同的变量，即 $\boldsymbol{w}^{\mathrm{q}}$、$\boldsymbol{w}^{\mathrm{k}}$ 和 $\boldsymbol{w}^{\mathrm{v}}$ 以构造查询向量 $\boldsymbol{q}$、键向量 $\boldsymbol{k}$ 和值向量 $\boldsymbol{v}$，这些变量随着模型的训练不断迭代优化。以查询向量为例，计算公式如下：

$$\boldsymbol{q}_i=\boldsymbol{v}_{x_i}\cdot\boldsymbol{w}_i^{\mathrm{q}},\quad \forall i\in\{1,2,\cdots,N\} \tag{3.1}$$

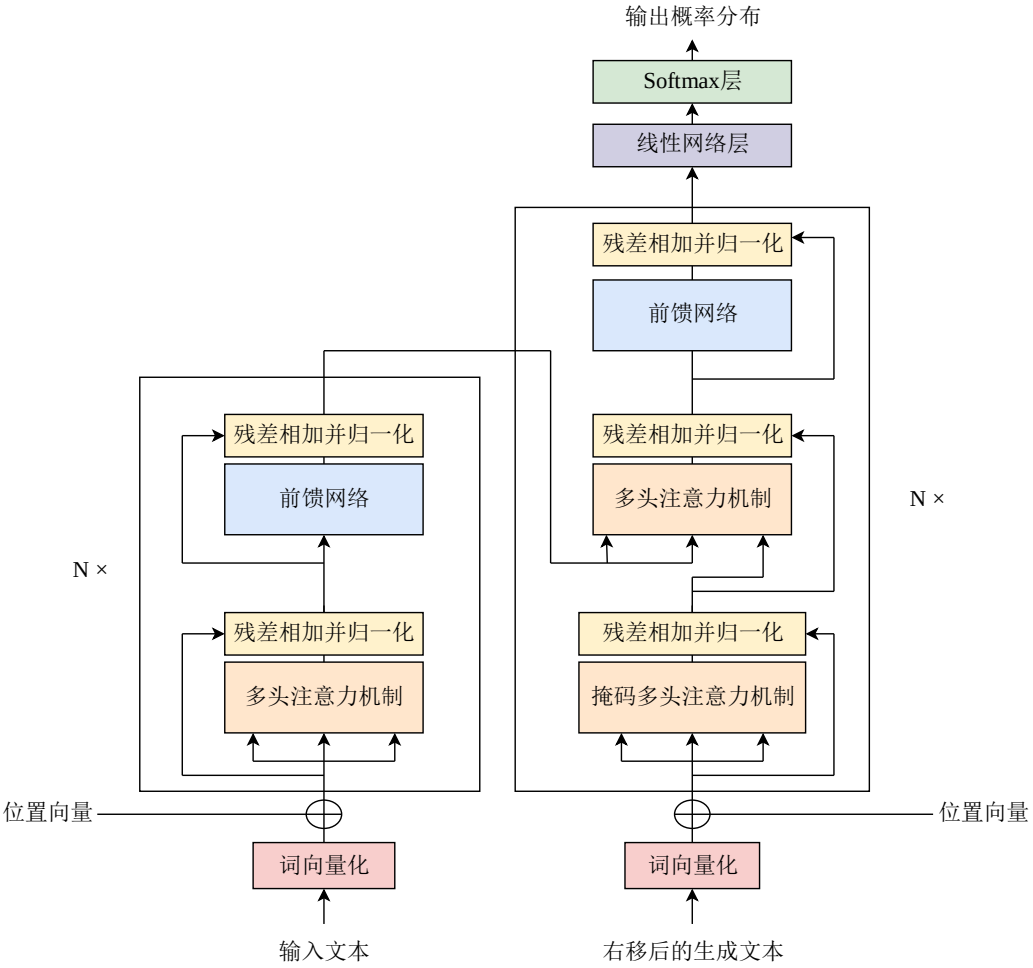

图 3.1　Transformer 模型框架图

第三步：计算分数 $s_{i,j}$。$s_{i,j}$ 为后续计算提供第 i 个和第 j 个输入单元之间的权重值大小，计算方式如公式(3.2)所示。

$$s_{i,j} = softmax\left(\frac{\boldsymbol{q}_i \cdot \boldsymbol{k}_j^{\mathrm{T}}}{\sum_{n=1}^{N} \boldsymbol{q}_i \cdot \boldsymbol{k}_n^{\mathrm{T}}}\right) \tag{3.2}$$

其中 $\boldsymbol{k}_j$ 代表第 j 个输入的键值向量。

第四步：计算单步输出特征向量，该值代表由注意力加权后的特征提取结果，计算公式如下所示：

$$\boldsymbol{z}_i = \sum_{j=1}^{N} s_{i,j} \times \boldsymbol{v}_j, \quad \forall i \in \{1, 2, \cdots, N\} \tag{3.3}$$

第五步：拼接所有的输出特征向量，得最终提取的特征矩阵。

以上的说法为拆分后的底层逻辑，在实际计算中，所有的计算都以矩阵运算的方式进行。例如公式(3.1)可改写为：

$$\boldsymbol{Q} = \boldsymbol{v}_x \cdot \boldsymbol{W}^{\mathrm{Q}} \tag{3.4}$$

其中，$\boldsymbol{W}^{\mathrm{Q}}$ 是由所有 $\boldsymbol{w}^{\mathrm{q}}$ 组成的矩阵，所得 $\boldsymbol{Q}$ 为所有查询向量所组成的查询矩阵。

正是因为所有进程都可以通过矩阵运算并行实现，Transformer 的计算效率极高。多头注意力机制在注意力机制的基础上重复操作数次，以收集更完善的特征信息。然而，这种并行运算方式仍然不能使得模型在解码端一次性输出完整的文本，而仍然需要逐字生成，使得文本生成相对而言仍是一个计算量较高的任务。这也是图 3.1 所示的解码端使用掩码多头注意力机制(Masked Multi-Head Attention)的原因：由于输出端仍是逐字生成，所以在训练第 *i* 个词时，需要屏蔽自第 *i*+1 个词起的所有后续文字。

3.3 方案设计

为实现高嵌入效率且保持语义一致性，该方案通过基于语言转换的释义技术嵌入秘密信息。如图 3.2 所示，隐写框架包含三个阶段，即语言编码、语言解码和秘密信息提取。消息发送方负责语言编码和解码阶段，而秘密信息提取由接收方完成。语言编码的目的是将原始英文文本转换至德文文本，两段文本以不同语言表述但所表示的含义相同。在语言转换的过程中，德文文本被视为英文文本的内在语义信息。对于语言解码阶段，德文文本被解码至英文，所得英文文本与原始给定文本和中间文本语义一致。隐写文本发送后，接收方将利用密钥和其他辅助信息从隐写文本中提取秘密信息。语言编码和语言解码共同组成了语言转换系统，而秘密信息在语言解码阶段进行嵌入。

3.3.1 语义可控生成隐写框架

该框架通过释义生成的方式实现自然语言隐写。如上文所介绍，为更广阔的应用场景，以实现将任意的文本作为原始给定文本，该框架使用了自监督释义生

成方法取代了使用特殊的释义数据集训练的方法。该框架将两种相似的语言作为原始语言和中间语言，即英文与德文，本章将该模型命名为 En2Ge2En（English-to-German-to-English）。这种操作称为语言转换（Pivot Translation）。

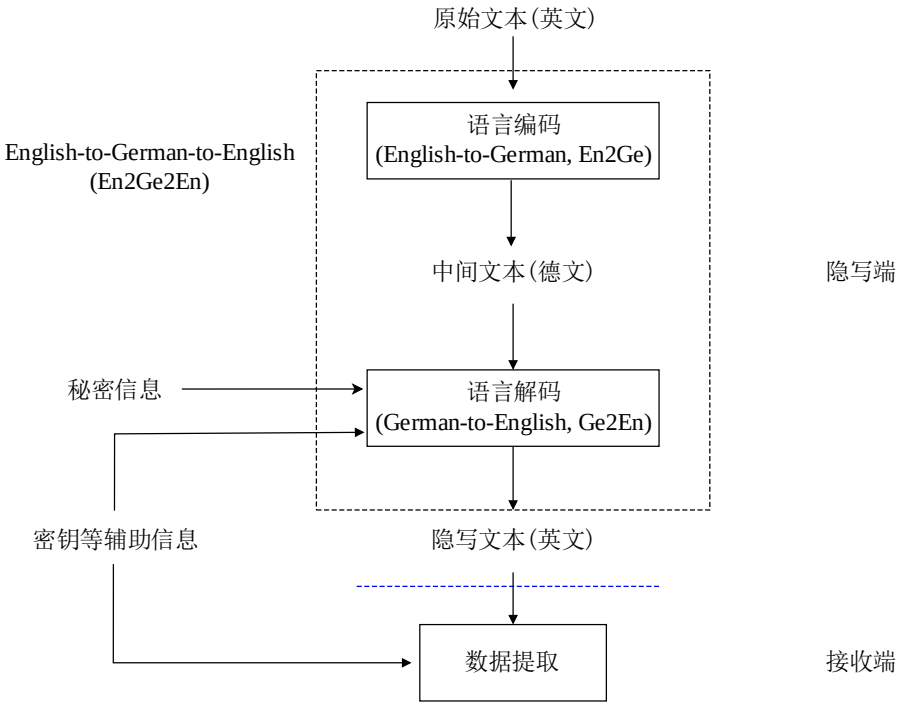

图 3.2　语义可控生成隐写框架图

使用英文-德文作为原始-中间语言对的原因主要有以下三点：

1) 语言转换的目的：语言转换是一种通过中间语言，生成在原始语言下与原始文本语义相似文本的方法。为了在框架中实现语义一致性，隐写文本与原始文本的差异需要被尽可能地缩小。根据 Federmann 等人[82]在语言转换领域的关键发现之一，使用相似的语言会对语言转换系统引入较少的语义影响，所以使用英文与德文这对起源接近的语言更适合该隐写框架。

2) 隐写的可实现性：尽管使用相似的语言对语言转化在语义上的影响较小，但是这也会引起所生成的文本欠缺多样性。然而，多样性并不是隐写问题需要考虑的因素。对于发送方而言，在秘密信息嵌入阶段引入多样性是一个简单的问题。因为发送方只需要提升嵌入量便可同时引入多样性，嵌入量增大代表了文本受到了更多修改，相当于引入多样性。然而，若期望在秘密信息嵌入阶段实现更强的语义一致性，则嵌入量必然收到损失。因此，对于隐写问题而言，语义一致性是

核心问题，这也意味着在语言转换中应使用更接近的语言。

3) 数据集的可信度：本章实验采用了 WMT2016 En2Ge 数据集。该数据集是翻译任务下的常用数据集，所以通过该数据集所得到的实验结果相比于使用其他语言数据集更具信服力。

如图 3.2 所示，En2Ge2En 模型使得任意的英文文本可以转换至德文文本，并保持语义一致性，之后德文文本又可以通过模型生成全新的英文文本。所得的全新英文文本与原始英文文本和德文文本语义都是相近的。值得注意的是，若不在语言解码阶段引入隐写编码，所得的英文文本将不携带任何的秘密信息。也就是说，无论嵌入秘密信息与否，通过 En2Ge2En 模型所得的英文文本都可以认为与原始文本的语义相似。

模型结构和模型训练并非隐写问题的主要研究内容，所以下面只简单阐述 En2Ge2En 模型的结构和训练策略。En2Ge2En 模型由两个结构完全相同的 Transformer 构成，Transformer 的细节可详见 Vaswani 等人的研究[16]。在介绍 En2Ge2En 模型的训练步骤之前，需要事先指出的是，En2Ge2En 模型的训练与秘密信息的嵌入互相独立。

En2Ge2En 模型由两部分 En2Ge 和 Ge2En 组成，每一部分都是一个基础翻译模型。在训练 En2Ge 模型时，采用原始数据集中的英文-德文文本对进行训练，而在训练 Ge2En 模型时，需要创建新数据集进行训练。这是因为在实际使用中，经过 En2Ge 模型生成的文本已与原始训练集中的德文存在差异，为了消除这种分布差异，在对 Ge2En 模型的训练过程也需使用由 En2Ge 模型生成的德文文本。因此，需要使用训练完成的 En2Ge 模型将所有原始数据集中的英文文本翻译至新的德文文本，之后使用新的德文文本取代原始数据集中的德文文本，并保留原始英文文本组成新的数据集。除了数据集使用上的差异，En2Ge 与 Ge2En 模型的训练过程是完全相同的。图 3.3 展示了使用不同数据集训练两模型的过程。这是一种自监督方法，该方法比使用特殊数据集的释义技术具有更广阔的应用场景。

3.3.2 数据嵌入

在训练 En2Ge2En 模型完成之后，即可通过该模型嵌入秘密信息。与传统修

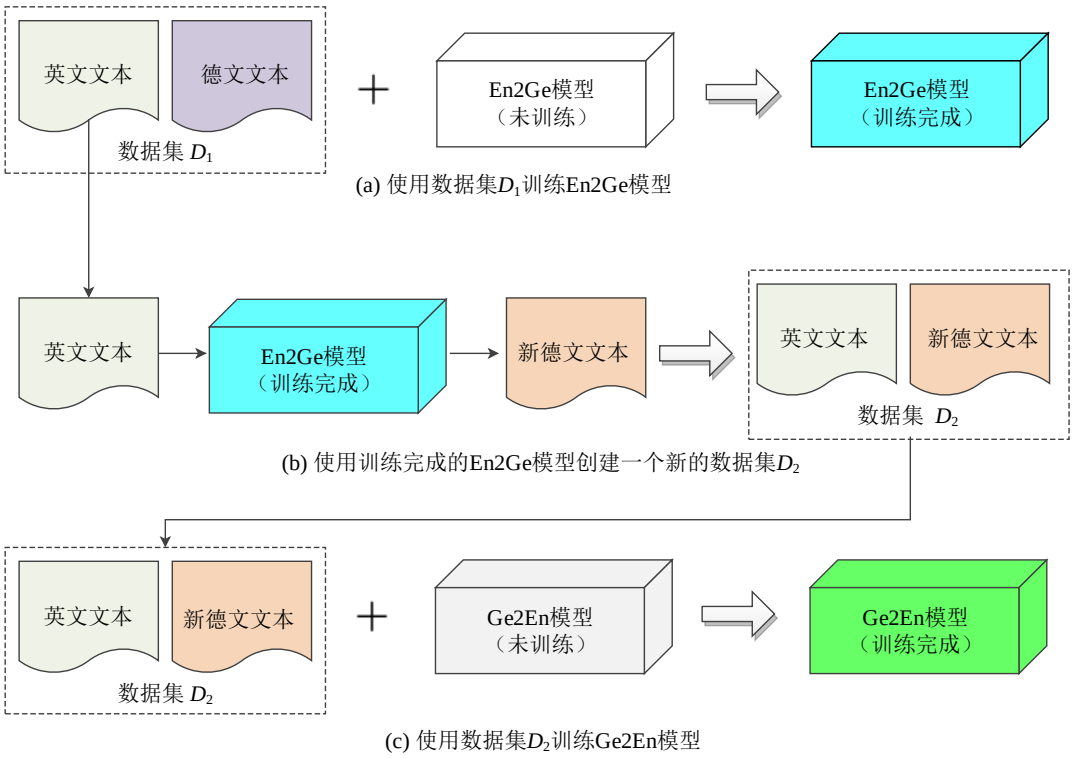

图 3.3　使用不同数据集训练 En2Ge 和 Ge2En 模型

改式方法不同的是，该框架以文本生成的方式改写了整段文本，生成式的嵌入方式保证了较大的嵌入量。如前文所介绍，En2Ge2En 模型可以逐词生成全新的文本，在拼接所有词汇之后可得完整的隐写文本。针对数据嵌入阶段，本章将隐写文本命名为 $\boldsymbol{y}=(y_1,y_2,\cdots,y_n)$，其中 n 为隐写文本的长度，y_i 为隐写文本中第 i 个词。值得注意的是，本章中的隐写文本 $\boldsymbol{y}$ 默认包含文本终止符。本章将英文词典命名为 $V=\{v_1,v_2,\cdots,v_m\}$，其中 m 是词典大小，v_i 代表词典中的第 i 个词。

对于文本生成任务，模型不会直接生成一个词，而是首先由解码器给出一个概率分布。对于 Ge2En 模型，其将由 En2Ge 模型生成的德文文本作为输入，在生成隐写文本中第 i 个词时，Ge2En 模型中的解码器为词典 V 中每个词附上概率值 $p_{i,j}$，其中 $j\in\{1,2,\cdots,m\}$，词典中所有词在该生成步骤中所占的概率值总和为 1，其数学式如下所示：

$$\sum_{j=1}^{m}p_{i,j}=1,\quad \forall i\in\{1,2,\cdots,n\} \tag{3.5}$$

概率值代表词作为该位置的最终输出的适合程度。如果总是采用概率值最高的词作为最终输出便可以获得一段质量较高的文本，其采样策略即为 top-1 策略，其数学公式可表示为：

$$y_i = \arg\max_{v_j \in V} p_{i,j} \tag{3.6}$$

概括地说，无论框架中具体使用的模型结构和采样策略，序列文本 $\boldsymbol{y} = (y_1, y_2, \cdots, y_n)$ 可由 Ge2En 模型生成。此处忽略了秘密信息、嵌入策略、密钥和辅助信息，而仅对文本生成问题进行讨论。根据不同的采样策略，公式中的 y_i 可不必强制选择概率值最高的词汇。部分采样策略允许在选定 y_i 时可构建出一个候选池，表示候选池中的每个元素都可作为最终选择。这类策略常用于文本生成，因为若采用 top-1 策略，则对于给定输入，模型将总是输出相同的文本，这使得生成文本缺乏多样性。如 top-k 策略中，y_i 将从概率值最高的 k 个词中选择，尽管其余元素的概率不是最大，但以一定的文本质量为代价，该策略提供了多样性。

这种思想也同样适用于自然语言隐写，候选池为隐写提供了冗余。发送方对候选池中的元素进行编码，并根据秘密信息从候选池中选择可以正确映射到所需二进制比特流的元素作为最终输出，则可以在文本生成的过程中嵌入秘密信息。于是，需要对公式(3.6)进行改写，改写后所得的公式(3.7)所表示的隐写策略为：总是选择符合映射关系的词中概率值最大的元素。

$$y_i = \arg\max_{v_j \in V, f(v_j)=b_i} p_{i,j} \tag{3.7}$$

其中，b_i 代表在第 i 个位置所需要传输的二进制比特流，$b_i \in \{0,1\}^{l_i}$，l_i 是该二进制比特流的长度，f 为词典中元素和二进制比特流的映射关系。

若选定 $y_i = v_j$，且 v_j 被映射至 b_i，则 y_i 已携带所需嵌入的秘密信息。为了提取秘密信息的可实现性，需要求 $f^{-1}(y_i) = f^{-1}(v_j) = b_i$。主流的生成式方法在生成每一个词的过程中都嵌入秘密信息，但经实验发现，若要求高质量的文本，不得不进一步降低嵌入量。于是该框架除了单次嵌入的比特流长度 l 以外，还引入了嵌入步长 s 以更好地控制嵌入量，这使得秘密信息均匀地分布在整个文本段落中，

从而提高了隐蔽性。于是，对于文本生成的采样策略，该框架中实际采用的是公式(3.6)和公式(3.7)的混合策略，两种策略的使用频率比例为$(s-1):1$，本章所提出算法的数据嵌入流程伪代码详见算法 3.1。

算法 3.1 数据嵌入过程的伪代码

输入：原始文本 ***c***，训练完成的 En2Ge2En 模型 ***M***，秘密信息 ***b***，词典 V，密钥 k，映射关系 f

输出：隐写文本 ***y***

1：根据 k 设置一个嵌入步长 $s \geq 1$

2：建立一个空序列 ***y***，并设置索引 $i=1$

3：**while** 需要生成一个词 **do**

4：　根据 ***M***，***c***，和 $(y_1, y_2, \cdots, y_{i-1})$ 获取概率分布 p_i

5：　**if** $(i-1) \bmod s \neq 0$ **or** ***b*** 已经嵌入完成 **then**

6：　　根据公式(3.6)确定 y_i

7：　**else**

8：　　根据已经嵌入的二进制比特流(如果存在)和 ***b*** 确定 b_i

9：　　根据公式(3.7)确定 y_i

10：　**end if**

11：　将 y_i 添加至 ***y***，并令 $i=i+1$

12：**end while**

13：**return** ***y***

对于算法 3.1 中的第 8 行，b_i 的长度为 l。它是由需要嵌入的完整秘密信息 ***b*** 和已经嵌入完成的秘密信息共同决定。举例而言，若 ***b***＝“010111000”且“010”已经嵌入完成，则下一次嵌入步骤中应嵌入的秘密信息为“111”。算法假设 l 总能够均分 ***b***，由于可以对二进制比特流的末尾添加“0”，这种假设是始终成立的。如果一段文本无法完整地嵌入整段秘密信息，则可以通过发送多段隐写文本的方式实现完整秘密信息的嵌入。

为了更好地说明数据嵌入步骤，本章在图3.4和图3.5中提供了简明的例子。

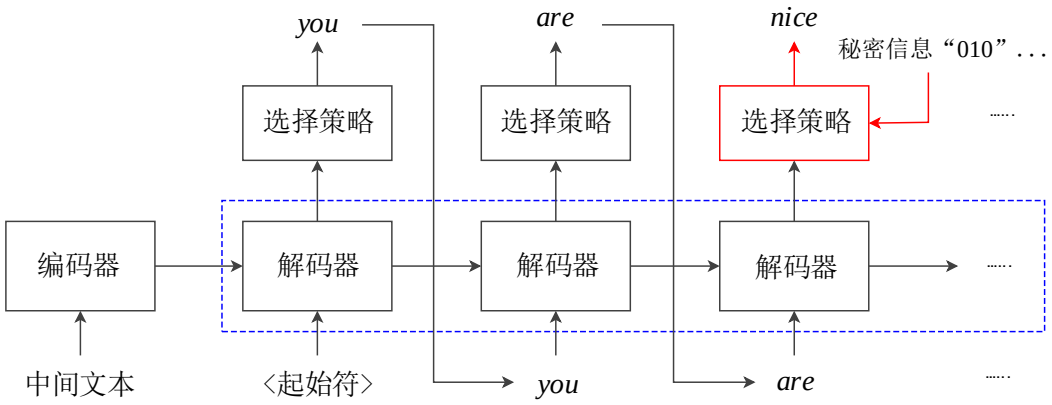

图 3.4　数据嵌入阶段示意图

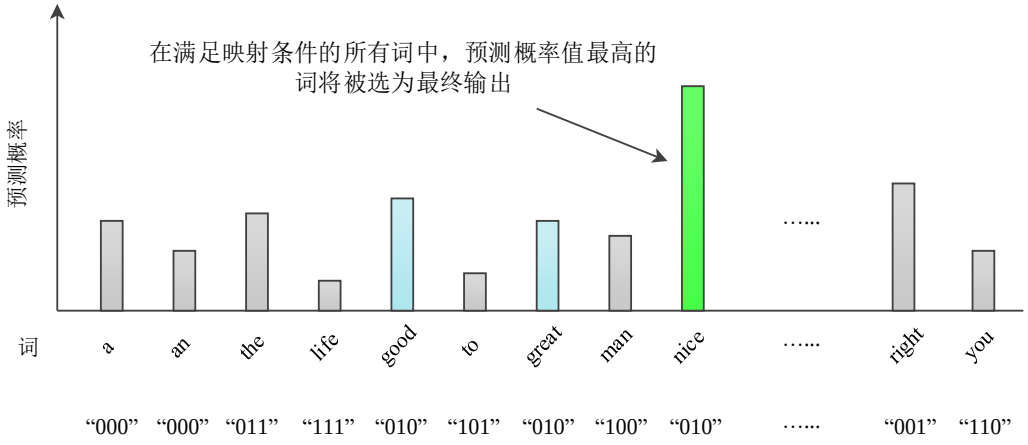

图 3.5　图 3.4 中“选择策略”模块的示意图

在图 3.4 中，秘密信息“010”在生成第三个词的步骤中被嵌入至“nice”一词中，这也代表了接收方可以通过必要的辅助信息从“nice”一词中提取出“010”。发送方选择“nice”一词的步骤如图 3.5 所示。根据前文所述，首先，词典 V 中的所有词都通过映射关系 f 与二进制比特流所对应，其次 Ge2En 模型在文本生成环节中输出了概率分布，使得词典 V 中的每一个词都有一个与之对应的概率值。最后，通过比较所有能够正确映射的词的概率值大小，选择其中概率值最大的词为最终输出，如图 3.5 所示，“nice”一词为映射至“010”的所有词中概率值最大的。以这种方式，一个单一的数据嵌入步骤已经完成，通过不断地重复这一步骤，就可以逐步嵌入完整的秘密信息。

根据公式(3.7)，在确定了采样策略之后，剩余的关键问题在于如何设计映

射关系f。任何生成式隐写编码理论上都可以作用于该框架，但由于 En2Ge2En 模型的复杂度较高，使用静态编码可以降低通信双方需要共享的辅助信息，是更加符合该框架的编码，所以本章采用桶编码进行仿真。由于桶编码构建较简单且并非本章节的核心内容，以下给出算法 3.2 以说明桶编码的构造过程。

算法 3.2 构建桶编码的伪代码

输入：原始英文数据集 $D_{1,0}$，词典 V，密钥 k，嵌入比特流长度 l

输出：桶 $\{V_0,V_1,\cdots,V_{2^l-1}\}$，即映射关系 f

1：初始化 $V_0=V_1=\cdots=V_{2^l-1}=\varnothing$

2：计算词典的大小 $m=|V|$

3：标记词典中的所有词为“未处理”

4：**while** $i\times 2^l<m$ **do**

5：　确定 $n_s=\min\{m-i\times 2^i,2^l\}$

6：　根据 k 从“未处理”的词中随机选择 n_s 个词

7：　根据 k 从 $\{V_0,V_1,\cdots,V_{2^l-1}\}$ 随机选择 n_s 个桶

8：　根据 k 将 n_s 个词放入 n_s 个桶中，并要求每个桶都放入且只放入一个词

9：　将该 n_s 个词标记为“已处理”

10：　$i=i+1$

11：**end while**

12：**return** $V_0,V_1,\cdots,V_{2^l-1}$

3.3.3 数据提取

对于接收方而言，从隐写文本中提取秘密信息是简单的。图 3.2 中所需的辅助信息包括映射关系 f 和所有预先确定的参数，如 s 和 l。所有的这些辅助信息可以通过共享的密钥控制。换句话说，通过预先共享的密钥，接收方首先获取嵌

入步长以确定接受到的文本中哪些位置是嵌入秘密信息的。再通过映射关系，接收方可以从隐写文本中确定所有的二进制比特流。最后接收方拼接所有的二进制比特流，便可获得完整的秘密信息。值得注意的是，映射关系等同于一个编码本，也可以提前由通信双方共享。

3.4 实验结果分析

3.4.1 实验设置

实验部分采用 WMT2016 英文-德文翻译数据集训练 En2Ge 模型。该数据集由 4.5×10^6 对文本和 3.3×10^4 个词组成。官方数据集已经将所有数据分成了三个独立的子集，即训练集、验证集和测试集。为了使实验部分更有说服力，实验扩充了测试集的大小。我们随机抽取了训练集中的部分文本对，并将其移入测试集。所以所抽取的文本对将不会参与到模型的训练过程中，但它们将被用作测试样本以展示所提出方法的性能。在 WMT2016 数据集中，原始的测试集包括了 3000 个文本对，通过上述操作，测试集的大小被扩充至 10000 个文本对。值得注意的是，即使从训练集中抽取了 7000 个文本对，但这对模型的训练影响是极小的，因为原始训练集中的文本对数量远大于这个数量级。

在 En2Ge 模型训练完成之后，将训练集和验证集中的英文文本通过 En2Ge 模型翻译至德文，并将原始英文文本和所生成的德文文本组成新的数据集以训练 Ge2En 模型。这是为了消除在后续的隐写过程中原始德文文本和 En2De 模型所生成的德文文本之间的统计差异。

如 3.3.1 节所述，实验分别使用两个相同的 Transformer 模型训练 En2Ge 和 Ge2En 模型。实际训练使用了自然语言工具 Fairseq。Fairseq 为模型训练提供了便利，该工具内置了大量常用的模型和配置参数，这节省了大量本章所提出的方法在模型训练上的时间。对于 Transformer 的参数设置，实验采用了原论文中所使用的所有参数。为了评估翻译模型的性能，实验采用 beam-search 采样方法，并设置 beam size 为 4，之后选取所有候选文本中最佳的文本作为输出文本，最后引入 BLEU 指标评估所生成的文本。实验结果显示，En2Ge 和 Ge2En 模型所

生成文本的平均 BLEU 值分别达到了 27.83 和 51.10。这表示两个翻译模型的性能与目前主流方法的性能接近。值得注意的是，训练 En2Ge 和 Ge2En 模型与嵌入秘密信息无关，是常规的针对翻译任务的模型训练。

需要强调的是，语言转换是自然语言隐写的一种框架。任何的语言和模型都可应用到这个框架之中。数据集和模型的质量必然影响了后续的隐写文本的质量，所以采用常用的语言和先进的数据集是更为推荐的。在本章的仿真实验中采用了英文-德文的数据集和 Transformer 模型作为示例，是因为该数据集和模型都是自然语言研究中常用的，以此增强实验的说服力。

在隐写分析实验中，实验采用了预训练模型 BERT 作为基础模型并使用自然-隐写文本对进行微调。实验将上述测试集中的英文文本通过 En2Ge2En 模型得到 10000 个隐写文本，与原始文本组成 10000 对自然-隐写文本数据对。再将该 10000 对文本分为三部分，6000 对文本作为隐写分析实验中的训练集，1000 对文本作为验证集，3000 对文本作为测试集。该隐写分析方案假设监测方获取了部分自然-隐写文本对和配对的标签，并分析在该不利条件下，所提出方法的抗隐写分析能力。对于 BERT 模型的训练，使用的学习率为 10^{-6}，采用 Adam 优化器[83]，批量计算大小设置为 32，并训练 60 个循环。

对于每一次隐写分析实验，由于深度学习算法的随机性，实验重复十次模型训练。每一次模型训练中，采用验证集中表现最好的模型在测试集上的分析结果，最后收集十次分析数据，以平均值±标准差的方式给出最终结果。值得注意的是，尽管关于隐写分析的实验是以监测方的角度进行分类判断，结果指标为正确率，但实际该实验反应的是各自然语言隐写方法的抗隐写分析能力。

由于下一章所提出的基于新型感知编码的自然语言隐写方法可视作为本章算法的一个改进，为方便区分，在实验部分，我们将本章提出的算法称为 PT+Bins（Pivot Translation+Bins coding）。

3.4.2 参数设置对算法的影响

为了展示算法中参数设置对所提出方法的影响，表 3.1 展示了不同设定下的隐写文本的质量。在表 3.1 中，以 $l=2$ 为例，意为将词典中的所有词分配至 $2^l=4$

个桶中。对于 BLEU、BERTScore 和 PPL 指标，实验收集了所有隐写文本的对应指标并取平均值作为最终的结果。观察表 3.1 可知，在大多数情况下，不同的参数会导致不同的 BPW，也存在不同的参数设置也会得到相同的 BPW 值的特殊情况，即当 $s = l$ 时，代表隐写文本中每 s 个词嵌入 l 比特的秘密信息，BPW 值恒为 1。对于一个固定的 l，当 s 增大时，秘密信息将被更为平均地分摊到整段文本中，这将有效地提升文本质量。相反地，对于一个固定的 s，当 l 增大时，隐写文本的质量将降低，这是因为隐写文本承载了更多的秘密信息，也代表了原始文本被更大幅度地修改了。表 3.1 所展示的实验结果也恰符合隐写研究中的一般规律，即隐写文本质量和嵌入量之间存在权衡关系。总体而言，通过微调所提出方法中的参数，发送方可以在嵌入量和文本质量之间可以达到一个较好的平衡。特别地，注意到将 top-1 策略作为文本生成策略时，即使在不嵌入任何的秘密信息的情况下，隐写分析正确率仍然高于 50%。这说明自然文本与隐写文本的差异不仅来源于隐写编码，也来源于文本生成框架。

表 3.1　参数设置对 PT+Bins 的影响

参数	BPW	BLEU	BERTScore	PPL	正确率
s =∞, l = 1	0	47.60	0.9535	1.219	0.6422±0.0183
s = 3, l = 1	0.33	21.85	0.9155	2.545	0.8351±0.0076
s = 2, l = 1	0.50	16.02	0.9034	3.315	0.8638±0.0090
s = 3, l = 2	0.67	12.38	0.8942	3.922	0.8915±0.0137
s = 1, l = 1	1.00	6.86	0.8782	6.563	0.9105±0.0060
s = 2, l = 2		6.70	0.8776	6.042	0.9176±0.0068
s = 3, l = 3		7.02	0.8768	5.724	0.9167±0.0061
s = 2, l = 3	1.50	2.84	0.8555	9.775	0.9522±0.0037
s = 1, l = 2	2.00	1.51	0.8440	16.327	0.9778±0.0043
s = 1, l = 3	3.00	0.38	0.8235	33.103	0.9977±0.0043

通过使用 t-SNE[84]算法，图 3.6 提供了隐写文本和自然文本在统计规律上的差异。在该可视化过程中，BERT 被用于提取每一个自然文本和隐写文本特征，所得的隐向量即代表了整个文本的语义，这一概念与 BERTScore 评估方法的概念一致。之后，通过 t-SNE 算法对所有的高维向量降至二维，并用橙色的点代表隐写文本，蓝色的点代表自然文本。如图 3.6 所示，随着 BPW 降低，更多两种颜色的点重合在一起，这代表了隐写文本与自然文本更为接近，也即在 BPW 较低时，该方法具有更高的安全性。

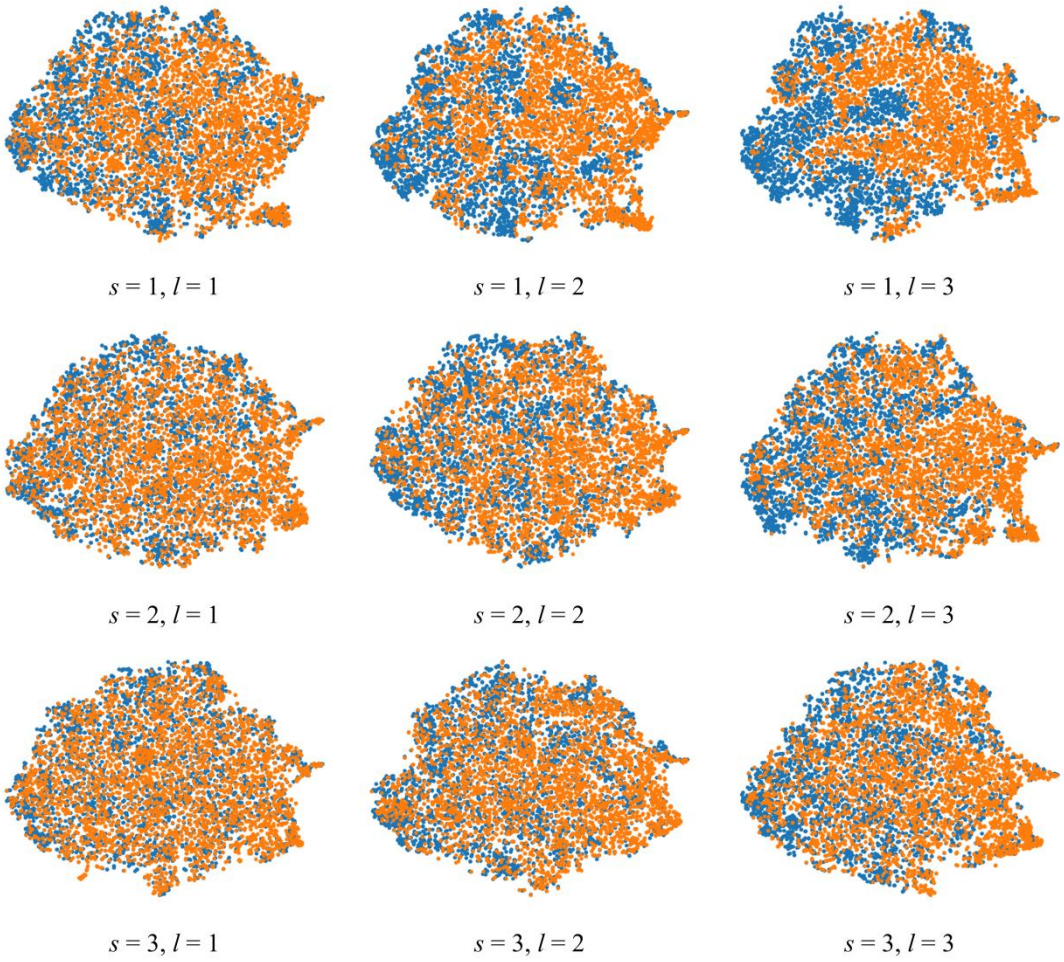

图 3.6　通过 t-SNE 算法可视化隐写和自然文本差异

表 3.2 提供了一些由本章所提出方法生成的隐写文本例子。与表 3.1 所揭示的规则相符的，当 BPW 升高，不仅各项机器指标性能将下降，隐写文本在视觉上也更容易被认为是不合语法的或是不流畅的。尽管这些携带了更多秘密信息的句子更容易被检测，但这些 BPW 较大的设置仍然是有意义的。因为文本质量不是自然语言隐写中的唯一指标，当发送方需要躲避较为严格的检测时，他可以选择一个 BPW 较小的设置。然而，在其他情况，发送方可以自由选择 BPW 较大的设置以追求更高的嵌入量。尤其在社交网络隐蔽通信的使用场景下，由于社交平台上的海量信息掩盖了隐写文本的存在，所以通常允许 BPW 值较大的设定。

注意到 PT+Bins 生成的大多数隐写文本在前半部分与原始文本的语义十分相似，却没有及时终止文本生成，而后续生成的文本与原始文本的语义有较大的差异。这是因为在桶编码下，将文本生成中最关键的特殊符号之一“<eos>”也

随机分配到一个桶中。“<eos>”意为“end of sentence”，在文本生成中代表一段文本的结束符号。例如，一段生成的文本“How are you ? <eos>”实际代表着“How are you ?”。“<eos>”的出现代表了文本生成过程的中止，否则文本将继续生成新的词汇。对“<eos>”的分配方式使得 PT+Bins 方法出现了上述不足。当一段文本应当结束时，“<eos>”必然是整个词典中条件转移概率最大的元素，然而“<eos>”只存在于一个桶中，这使得其仅有 $1/2^l$ 的概率被选中，从而导致 PT+Bins 方法在绝大部分情况下都会出现不恰当续写的情况。表 3.2 中，$s = 1, l = 1$ 设置下的隐写文本在观感质量上高于同 BPW 的其他文本，这便是因为该文本在需要结束的时刻，“<eos>”恰好映射到所需传输的二进制比特流。显然为了更好的文本质量，这一种偶然出现的情况应成为一种可控的常态情况。为解决这一问题，我们在第四章所提出的新型隐写编码中进行了改进。

表 3.2　使用 PT+Bins 算法生成的隐写文本例子

原始文本	Numerous panels are assessing the proposals , posing questions to the researchers .			
参数	BPW	隐写文本	BERTScore	PPL
$s = \infty, l = 1$	0	Numerous panels assess the proposals and ask questions to the researchers .	0.9619	1.311
$s = 3, l = 1$	0.33	A large number in panels assess the proposals and pose questions to researchers .	0.9291	3.395
$s = 2, l = 1$	0.50	A large number of panel evaluates the proposals and asks questions for researchers , i.e. , for the first part , it is possible to create a picture of the future .	0.9041	6.232
$s = 3, l = 2$	0.67	Many panels assess The Proposaries and ask questions to the Explorers in question . In the first part , the researchers are interested to learn more and to learn to understand the problem .	0.9025	7.090
$s = 1, l = 1$	1.00	A great many panel assess what has been put forward . We also ask questions to the researchers .	0.9194	10.749
$s = 2, l = 2$		Many panels evaluate the solutions and ask questions with regard from the researchers . This is why we work on specific research projects .	0.9001	15.320

表 3.2(续) 使用 PT+Bins 算法生成的隐写文本例子

原始文本	Numerous panels are assessing the proposals , posing questions to the researchers .			
参数	BPW	隐写文本	BERTScore	PPL
$s = 3, l = 3$	1.00	Various panels assess which proposals are forthcoming and ask question to the researchers . A number of these questions are to be answered to the researchers : the first analysis is to be carried out to the best advantage .	0.8971	10.924
$s = 2, l = 3$	1.50	Various panels assess the solutions and put questions down to those researchers as well to be their own points . A number will be presented to you in detail . and the research staff of the Committee on the Prosecutions Group will come up early .	0.8897	25.287
$s = 1, l = 2$	2.00	Many of labels assess the proposals and submit issues to researchers to decide whether they can do just that ? We will be happy with what happens , too . & nbsp ; We can do our best , with the right help , and to do the rest !	0.8688	34.021
$s = 1, l = 3$	3.00	Various panels are evaluators as to their proposed features , placing questions for researchers to consider for the future .. There you can get in question ! ! " . . , > The answer for this issue was how to connect the communication platform and the communication	0.8648	104.331

3.4.3 与主流方法对比

本章提出了一种基于释义技术的自然语言隐写框架，该框架将原始文本通过两次语言转换生成全新的含密文本，以文本生成的方式达成语义一致的目的。所以，基于该框架的自然语言隐写方法同时具有修改式和生成式隐写方法的特点。于是，实验部分同时将所提出的算法与修改式和生成式方法作比较。

主流的修改式方法主要依靠替换规则或者哈希运算嵌入秘密信息，这可能导致一个问题，即当原始文本中不存在符合要求的替换规则或不能正确地映射到所

需传递的二进制比特时，隐写是无法实施的。若发送方选择的原始文本不存在可供隐写的冗余，发送方不得不重新寻找新的原始文本直至满足隐写条件。

Wilson 等人[85]提出了一种基于 Paraphrase DataBase[86]和哈希运算的一种修改式自然语言隐写方法，在对比实验中，该方法简称为 PPDB。该方法通过释义数据库对原始文本进行部分替换，再将替换后的完整文本映射至二进制比特流。该方法的嵌入成功率理论上无法达到 100%，且仅在嵌入的秘密信息足够少时，该方法的成功率才可能无限接近 100%。然而，当所需嵌入的秘密信息量增大时，该隐写方法的成功率将急剧下降。隐写成功率上的不足也揭露了修改式自然语言隐写的最大问题，即嵌入容量小。

表 3.3 将 PT+Bins 与 PPDB 方法作对比。然而，PPDB 为句子级修改式自然语言隐写方法，无论句子长短，都映射至预先确定的固定长度的二进制比特流，故采用 BPS 进行量化更为合适。为了展示两方法在嵌入量相同时，隐写性能的差异，该实验中引入了公式(1.8)，具体做法如下：

1) 设置参数s和l，使用PT+Bins生成隐写文本，并统计所有隐写文本的长度，可得$\{(c_1,l_1),(c_2,l_2),\cdots,(c_{10000},l_{10000})\}$。其中$c_i$为数据集中第$i$个原始文本，$l_i$为根据原始文本$c_i$使用PT+Bins生成的隐写文本的长度。

2) 引用公式(1.8)，得$\{(c_1,b_1),(c_2,b_2),\cdots,(c_{10000},b_{10000})\}$，其中$b_i=\left[\frac{s}{l}\times l_i\right]$。即确认了对于PPDB方法而言，每一个原始文本$c_i$所需映射到的二进制比特流长度$b_i$，以保证每个原始文本在两种不同方法下的秘密信息嵌入量相同。

3) 根据$\{(c_1,b_1),(c_2,b_2),...,(c_{10000},b_{10000})\}$作为原始文本和目标嵌入比特数，使用PPDB方法生成隐写文本。

4) 分别收集使用两方法生成的隐写文本，并进行后续比较。

如表 3.3 所示，尽管在大部分设定和指标下，PPDB 方法领先于 PT+Bins，但 PPDB 方法在隐写成功率上存在明显的不足。PPDB 方法所得的隐写文本质量不随 BPW 上升而下降，因为其替换规则与嵌入量无关。然而，PPDB 方法的隐写成功率较低，这也侧面反映了传统修改式方法在嵌入容量上的不足。PT+Bins 方法可以采用任意的文本作为原始文本，所以隐写成功率恒为 100%，但 PPDB

方法依赖于替换规则和哈希映射，所以对于无法替换或无法映射到秘密信息对应的比特流上的文本，无法嵌入秘密信息，故隐写成功率无法到达 100%。由于修改式自然语言隐写方法在嵌入秘密信息时不使用语言模型，所以在该对比中不引入 PPL 指标。表 3.3 验证了所提出方法相对于修改式方法的优越性，即隐写容量更大，同时能在语义一致性上保持较高的水平。由于本章提出的自然语言隐写方法以文本生成的方式嵌入秘密信息，所以之后的实验还将所提出的方法与主流生成式自然语言隐写方法作对比。

表 3.3　PT+Bins 与 PPDB 方法对比

BPW	隐写方法	成功率	BLEU	BERTScore	正确率
0.33	PPDB	93.07%	**85.32**	**0.9753**	0.8383±0.0095
	PT+Bins	**100%**	21.85	0.9155	**0.8351±0.0076**
0.50	PPDB	62.07%	**84.64**	**0.9777**	**0.8466±0.0014**
	PT+Bins	**100%**	16.02	0.9034	0.8638±0.0090
0.67	PPDB	42.12%	**86.06**	**0.9648**	**0.8396±0.0038**
	PT+Bins	**100%**	12.38	0.8942	0.8915±0.0137
1.00	PPDB	15.75%	**83.94**	**0.9803**	**0.8548±0.0107**
	PT+Bins	**100%**	6.86	0.8782	0.9105±0.0060
1.50	PPDB	3.63%	**83.46**	**0.9761**	**0.8412±0.0076**
	PT+Bins	**100%**	2.84	0.8555	0.9522±0.0037
2.00	PPDB	0.60%	**85.49**	**0.9648**	**0.8416±0.0029**
	PT+Bins	**100%**	1.51	0.8440	0.9778±0.0043
3.00	PPDB	0.11%	**85.36**	**0.9576**	**0.8541±0.0061**
	PT+Bins	**100%**	0.38	0.8235	0.9977±0.0043

主流生成式方法不受原始文本的约束，而是直接生成一个全新的文本。与修改式方法可以将原始文本与隐写文本进行质量对比不同，主流生成式方法所生成的隐写文本没有对应的质量参照，这使得对隐写文本的质量评估产生困难。为解决该困难，在之后与生成式方法的比较中，实验将隐写文本与模型所生成的不含秘密信息的文本作比较，即在文本生成过程中，一直使用公式(3.6)作为采样规则，将条件转移概率最大的词作为最终输出，也即 top-1 策略。实验将使用该方式所生成的不携带任何秘密信息的文本视作自然文本。由于主流生成式方法无原始文本作为模型输入，所以无法使用语义转换框架。为公平比较，对比实验使用了 WMT2016 数据集中所有的英文文本对预训练模型 GPT-2 进行微调，并在微调后的 GPT-2 模型上使用主流的生成式编码，以生成隐写文本进行比较。

为在嵌入量相同的情况下对比隐写文本质量，表 3.4 所示的实验对所有方法引入了相同的 s 和 l。实验对 WMT2016 En2De 数据集中的英文文本统计了出现频率为前 10000 的句首词作为 GPT-2+FLC 和 GPT-2+Bins 方法的句首词。这是因为如果采用 top-1 策略且不设置句首词，GPT-2 模型所生成的文本将是固定的，最终将会获得 10000 个相同的自然文本。为引入多样性，并保证隐写分析模型的可训练性和有效性，实验引入了 10000 个不同的句首词引导主流的生成式方法生成不同的文本，所设置的句首词不含任何的秘密信息。如表 3.4 所示，PT+Bins 方法在多数设定下领先了主流的生成式方法，由 PT+Bins 方法所生成的隐写文本

表 3.4　PT+Bins 与主流生成式方法对比

参数	BPW	隐写方法	BLEU	PPL	BERTScore	正确率
$s=3, l=1$	0.33	GPT-2+FLC	10.03	12.213	0.8839	**0.7610±0.0208**
		GPT-2+Bins	3.79	31.962	0.8055	0.9996±0.0012
		PT+Bins	**21.90**	**2.545**	**0.9175**	0.7683±0.0109
$s=2, l=1$	0.50	GPT-2+FLC	10.30	14.877	0.8755	**0.8267±0.0044**
		GPT-2+Bins	3.50	50.053	0.8088	0.9989±0.0005
		PT+Bins	**16.02**	**3.315**	**0.9057**	0.8769±0.0214
$s=3, l=2$	0.67	GPT-2+FLC	6.45	15.419	0.8614	**0.8487±0.0118**
		GPT-2+Bins	2.45	58.133	0.8073	0.9991±0.0004
		PT+Bins	**12.38**	**3.922**	**0.8964**	0.8669±0.0113
$s=1, l=1$	1.00	GPT-2+FLC	3.62	35.273	0.8596	0.9257±0.0190
		GPT-2+Bins	2.22	69.947	0.7845	0.9994±0.0007
		PT+Bins	**6.86**	**6.563**	**0.8805**	**0.9217±0.0164**
$s=2, l=2$		GPT-2+FLC	3.14	40.304	0.8541	**0.9178±0.0056**
		GPT-2+Bins	2.51	71.801	0.7907	0.9994±0.0010
		PT+Bins	**6.70**	**6.042**	**0.8800**	0.9324±0.0019
$s=3, l=3$		GPT-2+FLC	3.15	41.865	0.8536	0.9419±0.0091
		GPT-2+Bins	2.19	68.115	0.7809	0.9996±0.0011
		PT+Bins	**6.77**	**5.724**	**0.8787**	**0.9368±0.0144**
$s=2, l=3$	1.50	GPT-2+FLC	1.23	64.462	0.8435	0.9606±0.0168
		GPT-2+Bins	1.10	95.4863	0.7800	0.9993±0.0014
		PT+Bins	**2.70**	**9.775**	**0.8576**	**0.9562±0.0050**
$s=1, l=2$	2.00	GPT-2+FLC	0.77	83.374	0.8383	0.9839±0.0146
		GPT-2+Bins	0.58	118.84	0.7716	0.9998±0.0003
		PT+Bins	**1.51**	**16.327**	**0.8464**	**0.9762±0.0037**
$s=1, l=3$	3.00	GPT-2+FLC	0.18	120.961	**0.8269**	**0.9872±0.0089**
		GPT-2+Bins	0.13	142.701	0.7628	0.9997±0.0002
		PT+Bins	**0.38**	**33.103**	0.8258	0.9956±0.0054

不仅能够更好地保持语义特性，并且更加流畅，这证实了所提出的方法相对于主流的生成式方法的优越性。

3.4.4 时间复杂度分析

本章所提出的隐写方法时间复杂度主要集中于模型训练阶段。然而，模型训练可以通过离线方式完成且不是本章的重点内容。所以时间复杂度比较实验聚焦于隐写阶段的时间损耗。该实验使用 Intel(R) Xeon(R) Gold 5118 CPU@2.30GHz 作为 CPU, 并使用 NVIDIA TITAN RTX 24GBx1 作为 GPU。

整个隐写过程可以分为两个阶段：辅助信息读取阶段和数据嵌入阶段。前者又可以细分为模型读取阶段和映射关系构建阶段。需要注意的是，辅助信息读取在整个隐写过程只需进行一次，读取完毕的模型和映射关系可以被重复使用，直至隐写完全结束。对于数据嵌入阶段的时间统计，实验分别统计了使用各个方法生成一段隐写文本的平均时间消耗作为最终展示的数据，结果如表 3.5 所示。

表 3.5　PT+Bins 时间复杂度分析(秒)

隐写方法	辅助信息获取		数据嵌入
	模型读取	构造映射关系	
PPDB	/	212.318	0.091
GPT-2+FLC	9.225	~0	0.284
GPT-2+Bins	9.093	0.043	0.285
PT+Bins	27.061	0.056	0.593

PPDB 方法不需要使用语言模型且替换和哈希映射相比于文本生成速度更快，但是其需要花费大量时间加载映射关系。由于映射关系只需要被读取一次，所以 PPDB 方法在需要生成多段隐写文本时仍然是一个可行的方法。GPT-2+FLC 和 GPT-2+Bins 都属生成式方法，由于两者使用的模型相同，所以理论上所需的模型加载时间应相同，但在实际使用中，模型读取的时间损耗总会有轻微的变化。对于 GPT-2+FLC 方法，其不需要提前构造映射关系，而是在文本生成的过程中临时构建候选池，而 GPT-2+Bins 方法只是对词典中的所有词进行随机分配，所以两者在映射关系读取阶段的时间消耗可忽略不计。PT+Bins 方法采用 Seq2Seq2Seq 框架，更复杂的框架也意味着更高的模型复杂度。由于该方法所使用的模型参数量大于 GPT-2 模型参数量，所以本章所提出的方法的确需要更多

的时间。然而，以一定的时间复杂度为代价换取更高的文本质量是有价值的，因为安全性和嵌入量才是自然语言隐写中的核心要素。

3.5 本章小结

本章分析了修改式和生成式自然语言隐写方法的优缺点，提出了一种基于释义技术的语义可控生成隐写框架。通过该框架可以实现以文本生成的方式保证隐写文本与原始文本的语义相似，故而结合了修改式和生成式自然语言隐写方法的优势。通过生成式隐写编码，基于该框架的自然语言隐写方法可以实现较高的隐写效率；通过控制语义一致性，基于该框架的自然语言隐写方法保证了隐写过程的隐蔽性和安全性。实验结果显示，该方法相较于主流修改式方法具有更大的嵌入容量，并保证了隐写成功率为 100%；相较于主流生成式方法在语义一致性上有较大提升。

第四章　基于语义感知编码的自然语言隐写

4.1 引言

生成式自然语言隐写方法主要分为两个部分：隐写框架和隐写编码。第三章提出了一种基于新型隐写框架的自然语言隐写方法。而第四章将介绍所提出的语义感知编码。本章的研究动机与上一章相同，即旨在实现语义一致性的同时保证高嵌入效率。不同的是，本章所提出的算法可视作上一章算法的改进方案，目的在于进一步提高隐写文本质量。

现如今，主流的生成式自然语言隐写编码为动态编码。动态编码根据每一个生成步骤所输出的概率分布建立映射关系，其在一定程度上提升了隐写文本的质量，但提高了接收方提取秘密信息的复杂度。由于每一个生成的词所对应的映射关系不相同，动态编码要求接收方获得与发送方一致的模型，且通过使用模型计算概率分布的方式获取映射关系。这大幅提高了双方需要共享的辅助信息的数量，也降低了秘密信息的提取效率。

自然语言处理模型的总体趋势是模型的大小和计算复杂度不断上升，这放大了动态编码的不足。尤其是当隐写方法需要构建比较复杂的模型结构时，动态编码反而更不实用。以第三章所提出的生成式隐写框架和对话场景为例。由于分词器、词典、密钥和隐写编码算法只需提前沟通一次，发送方若使用静态编码，则通信双方即可实现重复地隐蔽通信。然而，若发送方使用动态编码，则需要为接收方额外提供每一次所发送隐写文本的对应原始文本，额外的共享内容将更容易引起监测方的怀疑，使得隐蔽通信的安全性受到损害。表 4.1 给出了静态编码与动态编码对所需共享的辅助信息的要求。于是，本章内容聚焦于静态编码设计，以追求在所需共享的辅助信息较少的前提下，提升隐写文本的质量。

表 4.1　基于静态编码和动态编码的算法对辅助信息的要求

静态编码	动态编码
分词器、词典、密钥、隐写编码算法	分词器、词典、密钥、隐写编码算法、 原始文本、语言模型

桶编码是静态编码中最具代表性的算法之一。在桶编码中，桶的质量决定了最终生成的隐写文本质量。然而，由于该编码方式是通过随机过程将词典中的所有词映射到二进制比特流，桶的质量过度依赖密钥和随机性。也就是说，若随机过程恰巧生成了质量较高的桶，所得的隐写文本质量就较高。反之，随机过程也容易导致算法生成质量较低的隐写文本。

如图 4.1 所示，“___”位置需要一个意为“故意地”的词，并要求该词映射到二进制比特“1”。不幸的是，由于随机分桶策略，所有表示“故意地”的词都放入了映射至“0”的桶中，这导致在如图所示的“___”位置上，将没有合适的选择，这使得所生成的隐写文本质量必然较低。为了解决上述问题，本章提出了新型隐写编码技术，即语义感知编码。该编码可视作是在桶编码基础上的一种改进。语义感知编码利用同义词技术，将语义相同的词汇平均分配到不同的桶中，以避免如图 4.1 所示的现象。

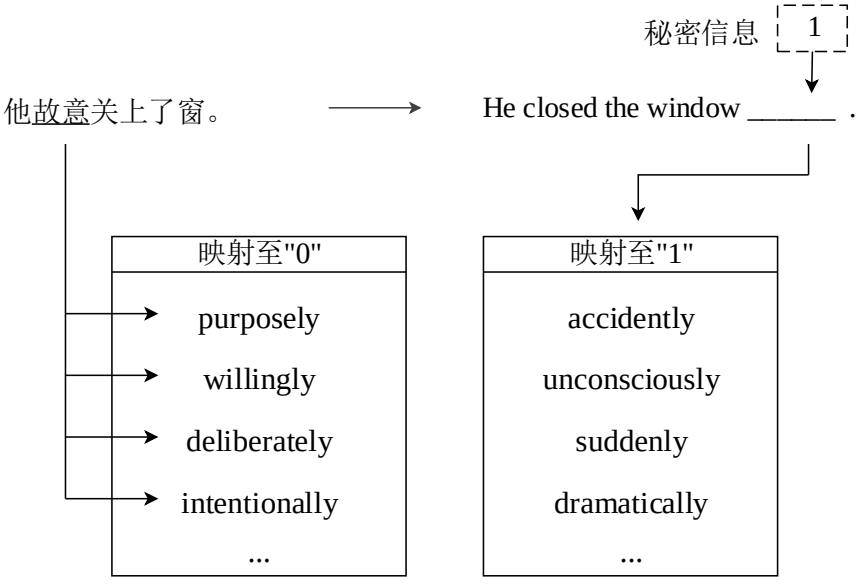

图 4.1　桶编码中的随机过程导致生成的隐写文本质量较低

4.2 相关内容简介

4.2.1 基于同义词替换的隐写

同义词之间的语义极为相似，所以同义词替换是修改式自然语言隐写的常用方法之一。它基于同义词词典和编码策略将原始文本中的部分词汇替换成对应的

同义词。例如，“see”和“look”可视作一对同义词，分别对应二进制比特“0”和“1”。“see”可被用于替换“look”以嵌入秘密信息“0”。

为防止部分替换词汇导致整段文本不流畅，当使用同义词替换时，通常会将一个大型语料库作为补充材料，例如 Google N-gram data[59]，以统计替换前后的文本段落出现在大型语料库的频率。这种做法通常会将所替换的位置附近的词汇组成一个样本，若该同义词替换后的样本出现在语料库的频率与替换前的频率接近，则可以认为该样本是流畅的。若替换后样本的出现频率极低，则通常意为这段文本是不流畅的，隐写者需要取消该替换，还原至原文本并继续探索下一个潜在的嵌入位置。

4.2.2 分词策略

对于英文文本生成任务，将世界上所有的英文词汇全部纳入词典中是不现实的，因为词汇的数量过多，将会导致模型训练复杂度过高。相反的，如果仅将一些常用词纳入词典，则将会有很多词汇无法被表示。同时，尽管 26 个字母足以表示所有的英文单词，但这将会导致文本的细粒度太小。例如，“I like machine learning”仅有四个单词，但如果以字符为一单位，则该文本将会被切分为 23 个编码单元(包含空格)，这也使得模型训练更加困难。

字节对编码(Byte Pair Encoding, BPE)是一种解决上述 OOV 问题的分词方案。它将数据集中的文本切割成字节对，并根据字节对在数据集中出现的频率将部分字节对合并。通过不断地合并字节对，以形成一个新的固定大小的词典。该分词策略介于字符级和单词级之间，使用该策略生成的词典中存在大量的词根和完整单词。常用词以完整单词的形式出现在词典中，不常用的词汇则可由词典中的一些常用的词根组合而成。

该策略已经成为自然语言处理的主流预处理方法。第三章所提出算法的仿真中也使用了该分词策略。但与上一章不同的是，由于第四章所提出的新型隐写编码使用了同义词相关的技术，这与分词策略存在一定冲突。因为仅有完整的词才可能具有同义词，分词策略将会减少词典中同义词关系的数量。所以，本章的实验部分将评估两者对该隐写编码的影响。

4.3 方案设计

生成式自然语言隐写可分为两个模块：隐写文本生成框架和隐写编码。不同于上一章着重于前者的研究和介绍，本章的方案设计将重点放在隐写编码上。由于语义感知编码仍然是为了实现语义可控，并且该编码实际是对上一章所设计的算法的一种改进措施，所以在本章方案设计中将略去两算法的共同部分的介绍，而聚焦于映射规则f的设计。

语义感知编码的设计主要有以下三个目的：1)改进上一章实验中发现的文本生成过度延伸的问题；2)针对图 4.1 所示的情况，所设计的编码方式需将语义相近的词汇均匀地映射到不同比特流上；3)使得提取秘密信息所需的辅助信息尽可能少。为了实现以上的目标，该方案首先对文本生成终止符单独处理，以使得文本生成过程在恰当的时刻终止；其次，该方案在桶编码的基础上引入同义词概念以提高合适语义的词汇映射到秘密信息上的可能性。由于该编码属于一种静态编码方式，所以其所需的共享辅助信息少于目前主流的动态编码方式。

4.3.1 生成终止符处理

根据表 3.2 的实验结果及分析，语义感知编码首先对“<eos>”进行单独处理。为了防止文本生成过程无法在恰当的时刻停止，语义感知编码新建一个桶V_{2^l}，该桶有且仅有一个特殊符号“<eos>”，并规定若“<eos>”为任何文本生成步骤中输出概率最大的符号，则直接终止文本生成过程，无论该步骤是否需要嵌入秘密信息及嵌入何种秘密信息。

Fang 等人[68]提出过类似的桶编码变种策略以提升隐写文本的质量，该策略将大量的常用词汇分配至V_{2^l}中，并认为这些常用词是不可替代的。然而，这样的做法大幅降低了嵌入量，由于这些词出现的频率远高于其他词汇，所以文本中有大量的词将不再携带任何秘密信息。为了更好地权衡隐写文本质量和嵌入量，在本章中，算法规定仅有“<eos>”一个符号不携带任何秘密信息。相比于将“<eos>”视为一个普通词并不进行特殊处理的做法，本章所提出的语义感知编码在嵌入量上也有轻微的降低，但在实际操作中，这种降低可以忽略不记。由

于“<eos>”是文本生成的结束符号，故该符号实际存在于所有文本中，其真实嵌入量的计算方式如公式(4.1)所示。

$$\mathrm{BPS}=\frac{l\times N-1}{s} \tag{4.1}$$

其中 l 为每次嵌入的比特流长度，s 为嵌入的步长，N 为隐写文本的长度。

“<eos>”不再携带秘密信息。然而生成特殊符号“<eos>”所在的步骤是一个需要嵌入秘密信息的步骤的概率仅有 s^{-1}，且文本生成所得的往往是一段较长的文本。所以对于一段文本整体而言，其携带的秘密信息可以认为是几乎没有下降的，其数学表达式如公式(4.2)所示。所以本章的实验部分将使用近似 BPW 值作为衡量本章所提出算法的嵌入量的指标。

$$\mathrm{BPW}=\frac{l\times N-1}{s\times N}\approx\frac{l}{s} \tag{4.2}$$

与桶编码类似地，语义感知编码将设置 2^l+1 个独立的桶，l 为预先确定的所需嵌入的单位二进制比特流的长度，数学表达式如公式(4.3)所示。

$$V=\bigcup_{i=0}^{2^l}V_i \quad \text{and} \quad V_i\cap V_j=\varnothing,\quad \forall 0\le i\ne j\le 2^l \tag{4.3}$$

在完成对“eos”的处理后，“eos”将在后续处理中被忽略，算法将聚焦于处理其他词和桶之间的映射关系。值得注意的是，在本章所提出的算法中，除了“eos”以外，词典 V 中的所有词汇都携带秘密信息，其数学表达式如公式(4.4)所示：

$$f:\ V\to\bigcup_{i=0}^{\infty}\{0,1\}^i \tag{4.4}$$

对于任意的 $i\in\{0,1,\cdots,2^l-1\}$，根据序号 i，V_i 中的所有的词都映射到相同的二进制比特流。例如，当 $l=4$，所有在 V_3 中的词都映射到二进制比特流“0011”。对于接收方，当其收集到含有“eos”的隐写文本 $\boldsymbol{y}=(y_1,y_2,\cdots,y_n)$ 后，即可通过合并 $\left(f^{-1}(y_1),f^{-1}(y_2),\cdots,f^{-1}(y_{n-1})\right)$ 获取秘密信息。需要特别说明的是，对于大多数生成式隐写编码而言，由于其“eos”携带秘密信息，但“eos”不会在隐

写文本中直接显示，所以当接收方获取文本后需自行在文本末尾补上“eos”才视为获得完整的隐写文本。例如，假设接收方收到的隐写文本为“How are you？”，接收方在秘密信息提取前需要自行将隐写文本补充为“How are you？<eos>”。然而，对于本章所提出的算法，由于所有隐写文本的最后一个字符必定是“eos”，且“eos”不含任何秘密信息，所以接收方可以直接跳过对隐写文本最后一个字符 y_n 的处理。为了与其他自然语言隐写方法的描述统一，本章中隐写文本 $\boldsymbol{y}$ 仍然指一段包含了“eos”的文本。

4.3.2 语义均分

语义感知编码将词在数据集中的统计特性纳入考虑范围中。具体而言，算法先统计词典中每个词在数据集中出现的频率，并通过将词典中的所有词根据频率降序排列 $\left\{v_{p_1}, v_{p_2}, \cdots, v_{p_m}\right\}$，该集合满足 $h\left(v_{p_1}\right) \geq h\left(v_{p_2}\right) \geq \cdots \geq h\left(v_{p_m}\right)$，其中 $h\left(v_{p_i}\right)$ 指原始词典中第 i 个词出现在数据集中的频率。对于除了“<eos>”以外的任意 v_{p_i}，算法将其同义词集合定义为 C_{p_i}，也代表了 C_{p_i} 中所有元素都可以视作 v_{p_i} 的替换词，且满足语义不变。

对于 C_{p_i} 中的所有未处理元素，算法将其随机分配到桶中。通过有序地处理 $\left\{C_{p_1}, C_{p_2}, \cdots, C_{p_m}\right\}$，可以完成携带秘密信息的所有桶的构建，即 $\left\{V_0, V_1, \cdots, V_{2^l-1}\right\}$，再添加预先处理完成的 V_{2^l}，便可以最终完成映射关系 f 的构建，详细的伪代码可见算法 4.1。

为了更直观地说明语义感知编码的构建原理，图 4.2 提供了一个示例。在图 4.2(a)中，四个词 v_1, v_2, v_3, v_4 已经按照在数据集中的出现频率排列完成，即满足不等式 $h\left(v_1\right) \geq h\left(v_4\right) \geq h\left(v_2\right) \geq h\left(v_3\right)$。假设 l =1，则每个词都将被放入 V_0 桶或 V_1 桶，分别映射至二进制比特“0”和“1”。在图 4.2(b)中，v_1 已经被分配至 V_1，而 v_2 被分配至 V_0。在图 4.2(c)中，算法仅对 C_4 中未处理元素进行分配，也即仅将 v_4 分配至 V_1。相似的进程也在图 4.2(d)和图 4.2(e)中体现。如图 4.2(e)所示，

算法 4.1 构建语义感知编码的伪代码

输入：原始数据集 $D_{1,0}$，词典 V，密钥 k，嵌入比特流长度 l

输出：语义感知桶 $\{V_0, V_1, \cdots, V_{2^l}\}$，即映射关系 f

1：根据 $D_{1,0}$ 确定 $h = \{h(v_1), h(v_2), \cdots, h(v_m)\}$

2：通过降序排列 h，得 $h' = \{h(v_{p_1}), h(v_{p_2}), \cdots, h(v_{p_m})\}$，该有序集合满足不等式 $h(v_{p_1}) \geq h(v_{p_2}) \geq \cdots \geq h(v_{p_m})$（若存在多个词在数据集中的频率相同，则根据密钥 k 决定排列的先后顺序）

3：初始化 $V_0 = V_1 = \cdots = V_{2^l} = \varnothing$

4：标记 V 中所有词为“未处理”

5：设置 $V_{2^l} = \{\langle \text{eos} \rangle\}$，并标记“<eos>”为“已处理”

6：**for** $i = 1, 2, \cdots, m$ **do**

7：　确定同义词集 $C_{p_i} \subset V$

8：　收集 C_{p_i} 中所有的未处理词组成 C'_{p_i}，其个数表示为 $|C'_{p_i}|$

9：　**while** $|C'_{p_i}| > 0$ **do**

10：　　确定 $n_s = \min\{|C'_{p_i}|, 2^l\}$

11：　　根据 k 从 C'_{p_i} 随机选择 n_s 个元素

12：　　根据 k 从 $\{V_0, V_1, \cdots, V_{2^l-1}\}$ 随机选择 n_s 个元素

13：　　根据 k 将 n_s 个词放入 n_s 个桶中，并要求每个桶都放入且只放入一个词

14：　　将该 n_s 个词标记为“已处理”并移出 C'_{p_i}

15：　**end while**

16：**end for**

17：**return** $V_0, V_1, \cdots, V_{2^l}$

最终v_2和v_3映射至“0”，而另外两个词v_1和v_4映射至“1”。若发送方希望嵌入的秘密信息为“0”，那么他将根据概率分布使用v_2和v_3其中之一作为该步骤的最终输出。相反地，若发送方希望嵌入的秘密信息为“1”，那么他将使用v_1和v_4其中之一作为该步骤的最终输出。

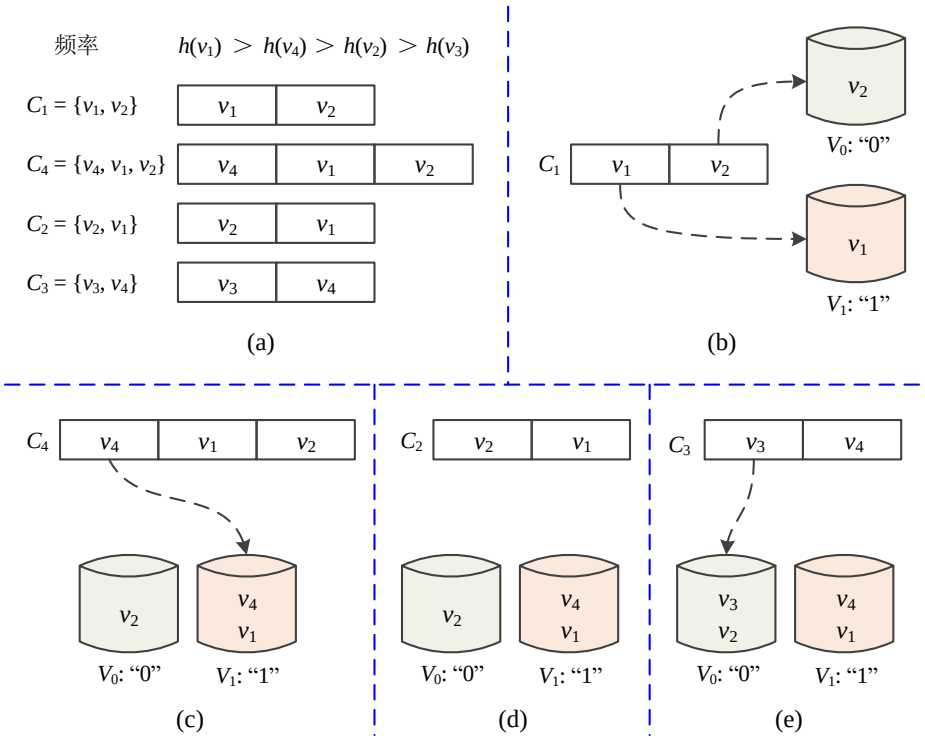

图 4.2　语义感知编码构造示例

4.3.3 同义词集合构造

算法 4.1 给出了语义感知编码的总体构造流程。下面将对于算法 4.1 的第 7 行获取同义词集合C_{p_i}的方式进一步补充阐述。本章的算法通过 WordNet[32] 确定C_{p_i}。WordNet 含有 1.1×10^5 个同义词集合，每一个集合中的词都具有相同的含义。由于一词多义现象，一个词可能出现在多个集合中。对于任意的一个词v_{p_i}，算法收集它的所有同义词作为C_{p_i}。通过简单直接收集所有同义词的方法，C_{p_i}中可能包含多种同义词集合，且每个同义词集合代表一种语义，所

以严格地说，C_{p_i}中的所有元素不完全是语义一致的。这种构建方法是可以进一步被优化的，我们将在未来的研究中继续探索。

为确定C_{p_i}，可以通过同义词集合构建一个同义词图。在同义词图中，每一个词是一个独立的节点，而同义词关系为连接这些词的边。因此C_{p_i}可以通过简单收集所有v_{p_i}的相邻节点获得。在获取C_{p_i}后，算法将对C_{p_i}中未被处理的元素逐个分配，这一过程所需的时间复杂度为$O\left(\left|C_{p_i}\right|\right)$。如果所有的$C_{p_i}$已经确定，则处理词典中的所有词的时间复杂度为$O\left(\sum_{i=1}^{m}\left|C_{p_i}\right|\right)=O(km)$，其中$k<<m$，代表一个同义词集合中具有的平均元素数量。

4.4 实验结果与分析

由于语义感知编码是针对图 4.1 所示的情况而设计的编码，旨在提高隐写文本的语义一致性。将该隐写编码作用于其他不强调语义一致性的生成式隐写文本模型上则效果不明显，所以本章实验仍然沿用了第三章所提出的隐写框架。与第三章实验不同的是，本章的实验主要研究的是所提出的新型编码对整体性能的影响，而不是隐写文本生成框架。本章实验将语义感知编码作用于语义可控生成自然语言隐写框架的方法称为 PT+SaBins（Pivot Translation+Semantic-aware Bins coding）。PT+SaBins 是针对上一章中提出的 PT+Bins 的一种改进方案，具体体现在使用了语义感知编码替代了桶编码。于是，本章节的实验部分将着重单独分析语义感知编码所带来的性能提升。本章的实验设置、数据集和参数若无特别说明，皆与上一章实验部分设定相同。

4.4.1 与桶编码对比

如表 4.2 所示，语义感知编码在所有情况、所有指标下性能都领先于桶编码，这是由于语义感知编码充分考虑了词的语义和分布特性，并且对于文本生成的结束词进行了单独的设置。在表 4.2 中，PT+Bins 与 PT+SaBins 分别简称

为 Bins 与 SaBins。

由于语义感知编码将同义词平均地分配到了所有桶中，使得发送方在实际隐写过程中，选择到更恰当的词的机会更大，这使得该编码方式在 BLEU 和 BERTScore 两个指标上表现优异。由于语义相似的词总是拥有相似的概率值，所以在该过程中，语义感知编码实际也将不同的词按条件转移概率的大小进行了平均分配，在一定程度上使得不同的桶中概率分布更为相似，这使得在最终选定输出词的时候，那些对应条件转移概率更大的词更容易被选择，减轻了随机性带来的不良影响，这也使得该编码在 PPL 指标上的表现更为优异。不仅如此，抗隐写分析能力是更具综合性的指标，在抗隐写分析能力上的表现对比也表明了语义感知编码是一种优于桶编码的编码方式。

表 4.2　语义感知编码与桶编码对比

参数	隐写方法	BLEU	BERTScore	PPL	正确率
$s=\infty, l=1$	-	47.60	0.9535	1.219	0.6422±0.0183
$s=3, l=1$	Bins	21.85	0.9155	2.545	0.8351±0.0076
	SaBins	**25.92**	**0.9213**	**2.282**	**0.8111±0.0047**
$s=2, l=1$	Bins	16.02	0.9034	3.315	0.8638±0.0090
	SaBins	**19.10**	**0.9083**	**3.096**	**0.8458±0.0043**
$s=3, l=2$	Bins	12.38	0.8942	3.922	0.8915±0.0137
	SaBins	**14.87**	**0.8999**	**3.616**	**0.8834±0.0050**
$s=1, l=1$	Bins	6.86	0.8782	6.563	0.9105±0.0060
	SaBins	**8.58**	**0.8840**	**6.019**	**0.8960±0.1000**
$s=2, l=2$	Bins	6.70	0.8776	6.042	0.9176±0.0068
	SaBins	**8.59**	**0.8835**	**5.530**	**0.9092±0.0038**
$s=3, l=3$	Bins	7.02	0.8768	5.724	0.9167±0.0061
	SaBins	**8.82**	**0.8799**	**5.710**	**0.9086±0.0055**
$s=2, l=3$	Bins	2.84	0.8555	9.775	0.9522±0.0037
	SaBins	**3.90**	**0.8620**	**9.637**	**0.9485±0.0038**
$s=1, l=2$	Bins	1.51	0.8440	16.327	0.9778±0.0043
	SaBins	**1.88**	**0.8491**	**15.665**	**0.9560±0.0030**
$s=1, l=3$	Bins	0.38	0.8235	33.103	0.9977±0.0043
	SaBins	**0.43**	**0.8252**	**28.557**	**0.9786±0.0036**

表 4.3 提供了由本章所提出算法生成的隐写文本。与表 3.2 对比可知，在语义感知编码中对于“<eos>”的单独设置是有效的。表 4.3 中所展示的隐写文本在表达完原始文本中所包含的语义后便及时中止，并没有过多延伸其他的内容。由上一章的分析可知，不恰当的延续文本生成过程使得所延伸的内容与

表 4.3　使用 PT+SaBins 算法生成的隐写文本例子

原始文本	In fantastic weather , 214 cyclists came to Illmensee to take on the circuit , over the hills and around the lake .			
参数	BPW	隐写文本	BERTScore	PPL
$s=\infty$, $l=1$	0	When the weather was fine , 214 cyclists arrived in Illmensee to take the route over the hills and the lake .	0.9579	1.361
$s=3$, $l=1$	0.33	In fine weather 214 cyclists arrived in Illmensee to take the route over the hills and the lake .	0.9571	1.891
$s=2$, $l=1$	0.50	In fine weather , we had 214 bike rikers coming to Ill lake , to pick up the route across the hills and the lake .	0.9223	5.167
$s=3$, $l=2$	0.67	With fine weather , 214 bicyclists flocked to Lake Illmen to take over the course across the hills and the lake , and the trip was made in the winter .	0.9187	6.478
$s=1$, $l=1$	1.00	In fine weather , 2fourteen bike riders arrived to Ill lake , in order for its way across hill and Lake .	0.8852	10.694
$s=2$, $l=2$		With beautiful wear conditions 2fourone bike drivers were arriving inside Illa , to take track all the distance around hills plus See , where you will be surprised by the quality service of all the bikeers .	0.8825	14.854
$s=3$, $l=3$		When the weather became nice , the 214 touring cybermen came into Illmenus lake to assume the route all over the summits and along the lake itself .	0.8971	12.331
$s=2$, $l=3$	1.50	When the local weather occurred , 214 local cyriders arrived in Hillside Illa to undertake the distance over top of both the sides of hills and of the river lake to take them .	0.8734	23.008
$s=1$, $l=2$	2.00	With beautiful wear conditions 2fourone bike drivers were arriving inside Illa , to take track all the distance around hills plus See , where you will be surprised by the quality service of all the bikeers .	0.8725	53.846
$s=1$, $l=3$	3.00	When a fine climate hit 215 cylindarles would visit Easters , as the path crossed either lake Ille and its mountain \#be .	0.8535	173.32

原始文本所表达的含义不同，而如表 4.3 所示，在语义感知编码中，这个问题被成功解决了。

语义感知编码在各个方面领先桶编码，但同时它也符合隐写中的一般规律，即嵌入的秘密信息越多，隐写文本的质量越低。但由于参数是可变的，所以发送方可以自行选择合适的嵌入量以匹配实际使用的场景。若发送方对安全性的要求较高，可灵活选择使用较低的 BPW 设定；若发送方利用的信道环境对安全性要求较低，则可以使用较高的 BPW。

4.4.2 与主流方法在抗隐写分析性能上的对比

安全性是隐写方法的重要指标之一。所以本章额外使用了其他的隐写分析方法以进一步说明语义感知编码的优越性。表 4.4 与 4.5 所示的实验增加了 TS-RNN[87]和 TS-CSW[88]两种不同的隐写分析技术以验证所提出方法的优越性。从两表所示的结果可以发现，尽管由于使用的隐写分析模型和方法不同导致了不同的隐写分析结果，但是实验揭示的总体规律不变，即本章所提出的自然语言隐写算法总体优于其他主流方法。其中，表 4.4 中的实验在 BPW=0.33 的限制条件下进行，具体操作方法在 3.4.3 节已有具体说明。

表 4.4　PT+SaBins 与修改式方法在多种隐写分析技术下的正确率对比

隐写方法	Fine-tuned BERT	TS-RNN	TS-CSW
PPDB	0.8383±0.0095	0.5665±0.0015	0.6139±0.0194
PT+SaBins	**0.8111±0.0047**	**0.5268±0.0124**	**0.5527±0.0045**

值得注意的是，PT+SaBins 在两表中使用相同设定的隐写分析产生的结果是不同的。这是因为主流的生成式方法不受原始文本的约束，所以无法使用隐写文本与原始文本进行隐写分析。与生成式方法对比的实验将模型生成的不含秘密信息的文本视为自然文本，并将其与隐写文本进行比较。而在与修改式方法的比较中，由于所提出方法与主流方法都存在原始文本的参照，所以不需要采用这种代替的方法。由于比较对象的不同，导致了 PT+SaBins 在两表中使用相同隐写分析方法的结果是不同的。这也与第三章的实验部分设置一致。特别地，由表 4.5 与表 3.4 对比可知，通过引入语义感知编码，本章所提出的隐写方法在抗隐写分析能力上得到了显著的提升。

除此之外，可以发现 Fine-tuned BERT 在隐写分析上的能力超过 TS-RNN 和 TS-CSW。这是因为隐写分析模型的结构不同，相比于以 RNN 和 CNN 为基础的两种隐写分析模型，BERT 模型更为复杂，具有的参数量更多，使得它分析的结果更为精确。表 4.5 所示的实验结果说明 PT+SaBins 相比于主流方法更容易躲过早期的隐写分析手段，也侧面验证了由 PT+SaBins 方法生成的隐写文本具有更高的安全性。尤其是当 BPW = 0.33 时，由本章所提出的隐写方法所生成的隐写文本在 TS-RNN 和 TS-CSW 模型隐写分析的结果接近于 50%，近似于随机猜测的结果。

表 4.5　PT+SaBins 与生成式方法在多种隐写分析技术下的正确率对比

隐写方法	参数	BPW	Fine-tuned BERT	TS-RNN	TS-CSW
GPT-2+FLC			0.7610±0.0208	0.6608±0.0195	0.6900±0.0099
GPT-2+Bins	$s = 3, l = 1$	0.33	0.9996±0.0012	0.9225±0.0041	0.9308±0.0031
PT+SaBins			**0.7358±0.0206**	**0.5449±0.0133**	**0.5582±0.0157**
GPT-2+FLC			0.8267±0.0044	0.7350±0.0264	0.7741±0.0049
GPT-2+Bins	$s = 2, l = 1$	0.50	0.9989±0.0005	0.8333±0.0248	0.8841±0.0026
PT+SaBins			**0.7959±0.0019**	**0.6788±0.0119**	**0.6755±0.0161**
GPT-2+FLC			0.8487±0.0118	0.7425±0.0017	0.7733±0.0177
GPT-2+Bins	$s = 3, l = 2$	0.67	0.9991±0.0004	0.9258±0.0069	0.9308±0.0033
PT+SaBins			**0.8042±0.0115**	**0.7104±0.0154**	**0.7445±0.0005**
GPT-2+FLC			0.9257±0.0190	0.8550±0.0164	0.8941±0.0197
GPT-2+Bins	$s = 1, l = 1$		0.9994±0.0007	0.9486±0.0069	0.9841±0.0264
PT+SaBins			**0.9246±0.0194**	**0.7936±0.0042**	**0.7953±0.0218**
GPT-2+FLC			0.9178±0.0056	0.8575±0.0096	0.8833±0.0314
GPT-2+Bins	$s = 2, l = 2$	1.00	0.9994±0.0010	0.9247±0.0102	0.9725±0.0017
PT+SaBins			**0.8991±0.0017**	**0.7903±0.0085**	**0.8136±0.0243**
GPT-2+FLC			0.9419±0.0091	0.8450±0.0146	0.8901±0.0268
GPT-2+Bins	$s = 3, l = 3$		0.9996±0.0011	0.9353±0.0215	0.9733±0.0146
PT+SaBins			**0.9257±0.0081**	**0.8035±0.0128**	**0.8641±0.0148**
GPT-2+FLC			0.9606±0.0168	0.8916±0.0016	0.9175±0.0128
GPT-2+Bins	$s = 2, l = 3$	1.50	0.9993±0.0014	0.9192±0.0025	0.9875±0.0218
PT+SaBins			**0.9595±0.0177**	**0.8843±0.0216**	**0.9101±0.0054**
GPT-2+FLC			0.9839±0.0146	0.9391±0.0346	0.9466±0.0189
GPT-2+Bins	$s = 1, l = 2$	2.00	0.9998±0.0003	0.9910±0.0049	0.9908±0.0220
PT+SaBins			**0.9568±0.0050**	**0.9201±0.0181**	**0.9217±0.0314**
GPT-2+FLC			0.9872±0.0089	0.9641±0.0248	0.9700±0.0156
GPT-2+Bins	$s = 1, l = 3$	3.00	0.9997±0.0002	0.9950±0.0016	0.9941±0.0246
PT+SaBins			**0.9850±0.0113**	**0.9555±0.0074**	**0.9683±0.0086**

4.4.3 与 Common-token 策略对比

Fang 等人[68]提出了一种基于桶编码的变种 Common-token 策略以提升隐写文本质量。该策略旨在将一些常用的词汇配置在一个特殊的桶中，以避免这些词汇在需要时恰好无法映射到所需的秘密信息，但代价是这些常用词汇将不再携带秘密信息。由于这些词汇更为常用，导致该方法的嵌入量大幅降低。

该策略与语义感知编码都可视为是桶编码的变种，而且这两种方法对桶编码的改动是互相冲突的，所以我们在表 4.6 中提供了两种方法的对比。实验统计了

表 4.6　语义感知编码与 Common-token 的对比

隐写方法	参数	BPW	BLEU	PPL	BERTScore	正确率
GPT-2+Bins +Common-token	$s=3, l=1$	0.29	4.16	31.189	0.8250	0.9797±0.0016
PT+SaBins		**0.33**	**44.07**	**2.282**	**0.9426**	**0.7358±0.0206**
GPT-2+Bins +Common-token	$s=2, l=1$	0.34	5.99	48.350	0.8188	0.9878±0.0117
PT+SaBins		**0.50**	**18.88**	**3.096**	**0.9098**	**0.7959±0.0019**
GPT-2+Bins +Common-token	$s=3, l=2$	0.59	3.73	50.097	0.8122	0.9822±0.0027
PT+SaBins		**0.67**	**22.73**	**3.616**	**0.9139**	**0.8042±0.0115**
GPT-2+Bins +Common-token	$s=1, l=1$	0.72	2.57	67.820	0.8007	0.9694±0.0124
PT+SaBins		**1.00**	**8.59**	**6.019**	**0.8858**	**0.9246±0.0194**
GPT-2+Bins +Common-token	$s=2, l=2$	0.61	2.69	70.625	0.8063	0.9678±0.0217
PT+SaBins		**1.00**	**8.59**	**5.530**	**0.8859**	**0.8991±0.0017**
GPT-2+Bins +Common-token	$s=3, l=3$	0.46	2.72	67.929	0.8073	0.9406±0.0150
PT+SaBins		**1.00**	**8.76**	**5.710**	**0.8817**	**0.9257±0.0081**
GPT-2+Bins +Common-token	$s=2, l=3$	0.74	1.28	90.319	0.7921	0.9694±0.0019
PT+SaBins		**1.50**	**3.90**	**9.637**	**0.8640**	**0.9595±0.0177**
GPT-2+Bins +Common-token	$s=1, l=2$	1.26	0.69	109.450	0.7840	0.9983±0.0007
PT+SaBins		**2.00**	**1.89**	**15.665**	**0.8516**	**0.9568±0.0050**
GPT-2+Bins +Common-token	$s=1, l=3$	1.63	0.24	132.031	0.7712	0.9978±0.0008
PT+SaBins		**3.00**	**0.43**	**28.557**	**0.8275**	**0.9850±0.0113**

WMT2016 En2Ge 英文数据中的前 1000 个最常用的词汇作为“common tokens”，其余设置与表 4.5 中的 GPT-2+Bins 设置相同。如表 4.6 所示，尽管 Common-token 策略在一定程度上提升了隐写文本的质量，但仍然在一个较低的水平。在表 4.6 的所有设定下，本章所提出的方法在几乎没有损失嵌入量的情况下，所得的隐写文本质量仍然超出了对比方法。对于 Common-token 策略而言，随着分桶策略中桶的数量上升，嵌入量的损失更加严重。这是因为随着桶的数量上升，单个桶的质量下降，使得嵌入策略更倾向于使用不含秘密信息的桶中的词。该实验进一步体现出了本章所提出的方法的优越性。

4.4.4 “子词”策略对编码性能的影响

语义感知编码将同义词纳入了编码设计中，而分词策略将一个完整的词分成多个“子词”，这使得分词策略对该编码方式产生了影响，即使用“子词”策略将使得词典中的同义词对大幅减少。实验发现，若以完整的单词作为一个单元，则词典中的同义词关系为 2.6×10^4 对，然而，若使用“子词”策略，则同义词关系下降为 9.3×10^3 对。但由于“子词”策略仅将不常用的词进行拆分，所以实际对隐写文本的质量影响并不严重。

然而，若不使用“子词”策略，将对自然语言隐写的安全性产生严重影响。由于词典中都是完整的词却又不能包含所有的词，这将导致大量的英文单词无法被表示而使用“<unk>”代替，如表 4.7 所示。这些文本更容易引起怀疑，影响了隐蔽通信的安全性。此外，由于大量的词汇无法被表示，算法的抗隐写分析性能进一步下降，如图 4.3 所示，在不同的参数设定中，使用“子词”策略的抗隐写分析结果都优于不使用“子词”策略的结果。由实验可知，尽管分词策略减少了同义词关系的数量，但其带来的优势仍超过了由同义词数量损失引起的劣势。

表 4.7　PT+SaBins 在使用和不使用“子词”策略下的文本质量对比

原始文本	There are now 201 cardinals .
中间文本(非“子词”)	Es gibt jetzt 201 <unk> .
中间文本(“子词”)	Es gibt jetzt 201 Kardinäle .
隐写文本(非“子词”)	There are now <unk> <unk> .
隐写文本(“子词”)	There are now the 201 cardinals .

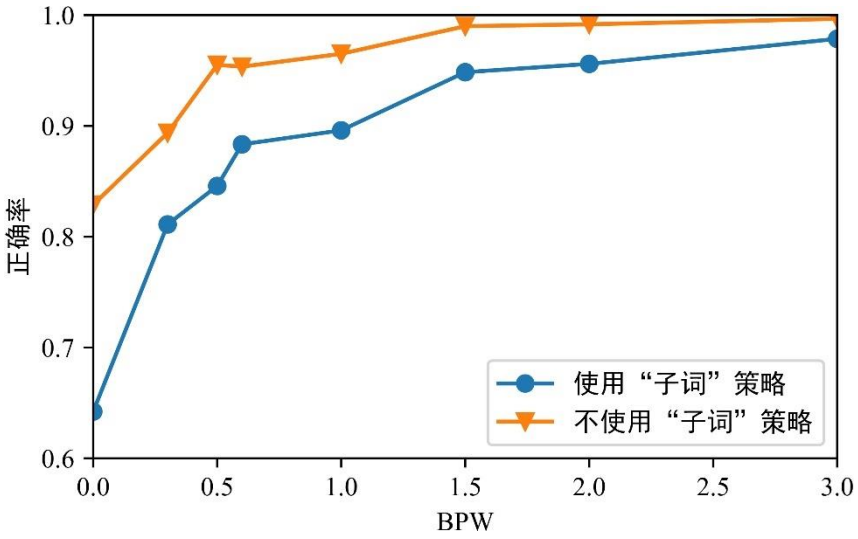

图 4.3　PT+SaBins 在不同分词策略下的隐写分析正确率对比

4.4.5 时间复杂度对比

语义感知编码引起的时间复杂度变化如表 4.8 所示。PT+Bins 和 PT+SaBins 采用了相同的自然语言隐写框架，而采用了不同的编码方式，这使得两者在读取模型和数据嵌入时的理论时间损耗是相同的。类似于实验中 GPT-2+FLC 和 GPT-2+Bins 的结果，相同模型在实际操作中的运算时间仍有轻微的差异。两方法不同点在于隐写编码技术，由于 PT+SaBins 是由语义可控生成隐写框架和语义感知编码共同组成，所以相比于 PT+Bins，需要消耗更多的时间在映射关系的建立，因为读取同义词关系并根据算法 4.1 构造桶是需要额外的时间的。

表 4.8　PT+SaBins 时间复杂度分析(秒)

隐写方法	辅助信息获取		数据嵌入
	模型读取	构造映射关系	
PPDB	/	212.318	0.091
GPT-2+FLC	9.225	~0	0.284
GPT-2+Bins	9.093	0.043	0.285
PT+Bins	27.061	0.056	0.593
PT+SaBins	27.656	26.155	0.649

但需要补充的是，在整个隐写过程中，映射关系只需构造一次便可以重复利用，所以在传输多个隐写文本的情况下，PT+SaBins 在总体耗时上是与 PT+Bins 相近的。并且以一定的映射构建时间为代价换取更高的隐写文本质量是有价值的，因为随着隐写文本质量的上升，隐蔽通信的过程将更难以被察觉。

4.5 本章小结

不同于前一章在隐写框架上的改良，本章聚焦于自然语言隐写的另一关键因素，即隐写编码。本章提出了基于语义感知编码的自然语言隐写方法，是对上一章算法的一种改良。语义感知编码旨在解决桶编码由于随机性而导致的隐写文本质量较低的情况，以进一步提升隐写文本的质量。实验表明，语义感知编码通过将语义相似的词均匀地分配到所有桶的方式提升了隐写文本的流畅性、语义一致性和安全性。除此之外，语义感知编码对隐蔽通信双方所需共享的辅助信息要求较低，使得该算法的隐蔽性较高，且更便于接收方提取秘密信息。

第五章　结论与展望

5.1 结论

自然语言隐写是将秘密信息嵌入看似普通的文本以实现隐蔽通信的技术，是网络信息安全的一个重要的研究分支。目前主流的自然语言隐写方法主要分为两个大类：修改式和生成式。修改式方法将一段自然文本作为原始文本，并通过对其整体或部分的修改嵌入秘密信息，该类方法可以较好地保持原始文本的语义，从而不容易引起监测方的怀疑，但其不足是嵌入容量较小。生成式方法利用文本生成语言模型，直接生成一段全新的文本，摆脱了原始文本的约束，从而实现了更大的嵌入量，但该类方法由于不存在原始文本约束，所以生成的隐写文本的内容和语义难以控制，更难适配实际的应用场景。

于是，本文提出了新型自然语言隐写方法，旨在同时实现语义一致性和高嵌入效率。该方法以生成式隐写方法为出发点，在其基础上实现语义一致性。生成式隐写方法主要分为两个步骤：首先需确定一个文本生成框架，然后在模型生成文本的过程中，使用隐写编码嵌入秘密信息。本文分别针对以上两个步骤提出了新型语义可控生成隐写框架和语义感知编码技术，具体研究内容如下：

1) 本文提出了新型的语义可控生成隐写框架。针对主流生成式和修改式自然语言隐写方法的不足，该框架运用释义技术，通过两次翻译过程，使得原始文本和最终生成的文本语义接近，从而实现了语义一致性。由于该框架以文本生成任务为基座，故基于该框架的自然语言隐写方法将具有生成式隐写方法高嵌入容量的优势。通过实验可知，该方法相比修改式方法具有更大的嵌入容量，而相比生成式方法能够更好地保留语义特性。

2) 本文还提出了一种新型的语义感知编码技术。针对主流的编码技术因信息编码的随机性导致载密文本语义质量低的问题，所提出的语义感知编码利用同义词关系，将语义相似的词均匀地映射到不同的二进制比特流上，以提升存在合适语义的词汇与秘密信息匹配的可能性。此外，该编码使得文本生成总是在恰当的位置停止，从而避免了过度延伸的隐写文本对总体质量的破坏。语义感知编码

相比主流隐写编码减少了隐蔽通信双方所需共享的辅助信息，提升了隐写实现的隐蔽性。由于语义感知编码目的同样在于保证隐写文本的语义一致性，所以将语义可控生成框架和语义感知编码结合可以使得隐写算法更好地控制语义。

5.2 展望

尽管实验结果表明了本文所提出的方法保持了较高的隐写文本质量和隐写效率，但在框架和编码细节上仍可以继续优化。

1) 简化语义可控生成框架的结构：在本文中，该框架采用了 Seq2Seq2Seq 结构以实现语义可控，其模型复杂度较高。若能够改用端到端的单一模型实现释义生成技术，则可以大幅降低时间复杂度。此外，该框架使用了翻译技术，而翻译文本相比于自然文本更难收集，所以若能改用单 Seq2Seq 模型，则可以摆脱翻译技术的约束，在数据集上有更多的选择空间。

2) 优化语义感知编码的同义词关系构造：语义感知编码利用了同义词技术进行语义平均分配，但由于自然语言中一词多义等特殊情况，同义词技术至今仍不完善，语义感知编码也不能实现严格地在语义层面进行均分。可以考虑从两个方向进行改进：(a)对生成文本进行句法分析，通过词性进一步约束同义词集合；(b)语义相似的词具有相似的词向量，可借用预训练模型以词向量为特征构建映射关系以代替同义词策略。

参考文献

[1] WU H, SHI Y, WANG H, et al. Separable reversible data hiding for encrypted palette images with color partitioning and flipping verification[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2016, 27(8): 1620-1631.

[2] 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.

[3] YANG Z, DU X, TAN Y, et al. Aag-stega: Automatic audio generation-based steganography[J]. arXiv preprint arXiv:1809.03463, 2018.

[4] CHEN Y, WANG H, WU H, et al. Adaptive video data hiding through cost assignment and STCs[J]. IEEE Transactions on Dependable and Secure Computing, 2019, 18(3): 1320-1335.

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

[6] BOJANOWSKI P, JOULIN A, MIKOLOV T. Alternative structures for character-level RNNs[J]. arXiv preprint arXiv:1511.06303, 2015.

[7] CHURCH K W. Word2Vec[J]. Natural Language Engineering, 2017, 23(1): 155-162.

[8] SENNRICH R, HADDOW B, BIRCH A. Neural machine translation of rare words with subword units[C]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, August 7-12, 2016, Berlin, Germany. Stroudsburg: ACL, 2016: 1715-1725.

[9] ZAREMBA W, SUTSKEVER I, VINYALS O. Recurrent neural network regularization[J]. arXiv preprint arXiv:1409.2329, 2014.

[10] CHENG J, DONG L, LAPATA M. Long short-term memory-networks for machine reading[J]. arXiv preprint arXiv:1601.06733, 2016.

[11] CHUNG J, GULCEHRE C, CHO K H, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling[J]. arXiv preprint arXiv:1412.3555, 2014.

[12] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017: 5998-6008.

[13] LUONG M T, PHAM H, MANNING C D. Effective approaches to attention-based neural machine translation[J]. arXiv preprint arXiv:1508.04025, 2015.

[14] HOLTZMAN A, BUYS J, DU L, et al. The curious case of neural text degeneration[J]. arXiv preprint arXiv:1904.09751, 2019.

[15] CACCIA M, CACCIA L, FEDUS W, et al. Language gans falling short[J]. arXiv preprint arXiv:1811.02549, 2018.

[16] PAULUS R, XIONG C, SOCHER R. A deep reinforced model for abstractive summarization[J]. arXiv preprint arXiv:1705.04304, 2017.

[17] FAN A, LEWIS M, DAUPHIN Y. Hierarchical neural story generation[C]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, July 15-20, 2018, Melbourne, Australia. Stroudsburg: ACL, 2018: 889-898.

[18] PAPINENI K, ROUKOS S, WARD T, et al. Bleu: a method for automatic evaluation of machine translation[C]//Proceedings of the 40th annual meeting of the Association for Computational Linguistics, July 6-12, 2018, Philadelphia, USA. Stroudsburg: ACL, 2018: 311-318.

[19] ZHANG T, KISHORE V, WU F, et al. Bertscore: Evaluating text generation with bert[J]. arXiv preprint arXiv:1904.09675, 2019.

[20] ZHOU X, PENG W, YANG B, et al. Linguistic steganography based on adaptive probability distribution[J]. IEEE Transactions on Dependable and Secure Computing, 2021, 19(5): 2982-2997.

[21] CHIANG Y L, CHANG L P, HSIEH W T, et al. Natural language watermarking using semantic substitution for Chinese text[C]//Proceedings of the 2nd International Workshop on Digital Watermarking, October 20-22, 2003, Seoul, Korea. Berlin: Springer, 2003: 129-140.

[22] 杨潇,李峰,向凌云.基于矩阵编码的同义词替换隐写算法[J].小型微型计算机系统, 2015, 36(06):1296-1300.

[23] CHANG C, CLARK S. Practical linguistic steganography using contextual synonym substitution and a novel vertex coding method[J]. Computational Linguistics, 2014, 40(2): 403-448.

[24] TOPKARA M, TASKIRAN C M, DELP III E J. Natural language watermarking[C]// Proceedings of the Security, Steganography, and Watermarking of Multimedia Contents VII, January 17-20, 2005, San Jose, USA. Bellingham: SPIE, 2013: 441-452.

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

[26] HUO L, XIAO Y. Synonym substitution-based steganographic algorithm with vector distance of two-gram dependency collocations[C]//Proceedings of the 2nd IEEE International Conference on Computer and Communications, October 14-17, 2016, Chengdu, China. Piscataway: IEEE, 2016: 2776-2780.

[27] CAN G, XINGMING S, YULING L, et al. An improved steganographic algorithm based on synonymy substitution for chinese text[J]. Journal of Southeast University (Natural Science Edition), 2007, 37(S1): 137-140.

[28] 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 VII, January 15, 2006, San Jose, USA. Bellingham: SPIE, 2006: 97-105.

[29] WANG F, HUANG L, CHEN Z, et al. A novel text steganography by context-based equivalent substitution[C]//Proceedings of the 2013 IEEE International Conference on Signal Processing, Communication and Computing, August 5-8, 2013, Kunming, China. Piscataway: IEEE, 2013: 1-6.

[30] RIZZO S G, BERTINI F, MONTESI D. Content-preserving text watermarking through unicode homoglyph substitution[C]//Proceedings of the 20th International Database Engineering & Applications Symposium, July 11-13, 2016, Montreal, Canada. New York: ACM, 2016: 97-104.

[31] ZHENG X, HUANG L, CHEN Z, et al. Hiding information by context-based synonym substitution[C]//Proceedings of the 8th International Workshop on Digital Watermarking, August 24-26, 2009, Guildford, UK. Berlin: Springer, 2009: 162-169.

[32] MILLER G A. WordNet: a lexical database for English[J]. Communications of the ACM, 1995, 38(11): 39-41.

[33] CLARK K, LUONG M T, LE Q V, et al. Electra: Pre-training text encoders as discriminators rather than generators[J]. arXiv preprint arXiv:2003.10555, 2020.

[34] RAFFEL C, SHAZEER N, Roberts A, et al. Exploring the limits of transfer learning with a unified text-to-text transformer[J]. The Journal of Machine Learning Research, 2020, 21(1): 5485-5551.

[35] LIU Y, OTT M, GOYAL N, et al. Roberta: A robustly optimized bert pretraining approach[J]. arXiv preprint arXiv:1907.11692, 2019.

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

[37] UEOKA H, MURAWAKI Y, KUROHASHI S. Frustratingly easy edit-based linguistic steganography with a masked language model[J]. arXiv preprint arXiv:2104.09833, 2021.

[38] ZHENG X, FANG Y, WU H. General framework for reversible data hiding in texts based on masked language modeling[C]//Proceedings of the 24th IEEE International Workshop on Multimedia Signal Processing, September 26-28, 2022, Shanghai, China. Piscataway: IEEE, 2022: 1-6.

[39] 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.

[40] MURPHY B, VOGEL C. Statistically constrained shallow text marking: techniques, evaluation paradigm, and results[C]//Proceedings of the Security, Steganography, and Watermarking of Multimedia Contents IX, January 28, 2007, San Jose, USA. Bellingham: SPIE, 2007: 363-371.

[41] LIU Y, SUN X, WU Y. A natural language watermarking based on Chinese syntax[C]//Proceedings of the 2005 International Conference on Natural Computation, August 27-29, 2005, Changsha, China. Berlin: Springer, 2009: 958-961.

[42] CHANG C, CLARK S. The secret's in the word order: Text-to-text generation for linguistic steganography[C]//Proceedings of the 24th International Conference on Computational Linguistics, December 8-15, 2012, Mumbai, India. Stroudsburg: ACL, 2012: 511-528.

[43] WILSON A, KER A D. Avoiding detection on twitter: embedding strategies for linguistic steganography[J]. Electronic Imaging, 2016, 2016(8): 1-9.

[44] KERMANIDIS K L. Hiding secret information by automatically paraphrasing modern Greek text with minimal resources[C]//Proceedings of the 22nd IEEE International Conference on Tools with Artificial Intelligence, October 27-29, 2010, Arras, France. Piscataway: IEEE, 2010: 379-380.

[45] TANG X, CHEN M. Design and implementation of information hiding system based on RGB[C]//Proceedings of the 3rd IEEE International Conference on Consumer Electronics, Communications and Networks, November 20-22, 2013, Xianning, China. Piscataway: IEEE, 2013: 217-220.

[46] ALI A A. New text steganography technique by using mixed-case font[J]. International Journal of Computer Applications, 2013, 62(3): 6-9.

[47] ZHOU X, WANG S, ZHOU N, et al. An erasable watermarking scheme for exact authentication of Chinese Word documents[C]//Proceedings of the 3rd IEEE International Congress on Image and Signal Processing, October 16-18, 2010, Yantai, China. Piscataway: IEEE, 2010: 1156-1160.

[48] KURIBAYASHI M, WONG K S. Improved DM-QIM watermarking scheme for PDF document[C]//Proceedings of the 18th International Workshop on Digital Watermarking, November 2-4, 2020, Chengdu, China. Cham: Springer, 2020: 171-183.

[49] LIU F, LUO P, MA Z, et al. Security secret information hiding based on hash function and invisible ASCII characters replacement[C]//Proceedings of the 2016 IEEE Trustcom/BigDataSE/ISPA, August 23-26, 2016, Tianjin, China. Piscataway: IEEE, 2016: 1963-1969.

[50] LIU Y, SUN X, LIU Y, et al. MIMIC-ppt mimicking-based steganography for microsoft power point document[J]. Information Technology Journal, 2008, 7(4): 654-660.

[51] SUI X G, LUO H. A steganalysis method based on the distribution of space characters[C]. Proceedings of the 2009 IEEE International of Communications, Circuits and Systems, June 25-38, 2006, Guilin, China. Piscataway: IEEE, 2006: 54-56.

[52] ZHONG S, FANG X, LIAO X. Steganalysis Against Equivalent Transformation Based Steganographic Algorithm for PDF Files[C]//Proceedings of the 2009 International Symposium on Information Processing, August 21-23, 2009, Huangshan, China. Finland: Academy Publisher, 2009: 75.

[53] ZHANG J, XIE Y, WANG L, et al. Coverless text information hiding method using the frequent words distance[C]//Proceedings of the 3rd International Conference on Cloud Computing and Security, June 16-18, 2017, Nanjing, China. Cham: Springer, 2017: 121-132.

[54] LONG Y, LIU Y. Text coverless information hiding based on word2vec[C]//Proceedings of the 4th International Conference on Cloud Computing and Security, June 8-10, 2018, Haikou, China. Cham: Springer, 2018: 463-472.

[55] WANG K, GAO Q. A coverless plain text steganography based on character features[J]. IEEE Access, 2019, 7: 95665-95676.

[56] HU Y, LI H, SONG J, et al. MM-stega: multi-modal steganography based on text-image matching[C]//Proceedings of the 6th International Conference on Artificial Intelligence and Security, July 17-20, 2020, Hohhot, China. Singapore: Springer, 2020: 313-325.

[57] MUÑOZ A, GALLARDO J C, ÁLVAREZ I A. Improving N-Gram linguistic steganography based on templates[C]//Proceedings of the 2010 IEEE International Conference on Security and Cryptography, July 26-28, 2010, Athens, Greece. Piscataway: IEEE, 2010: 209-212.

[58] CHANG C, CLARK S. Linguistic steganography using automatically generated paraphrases[C]//Proceedings of the 2010 Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, June 2-4, 2010, Los Angeles, USA. Stroudsburg: ACL: 2010: 591-599.

[59] MICHEL J B, SHEN Y K, AIDEN A P, et al. Quantitative analysis of culture using millions of digitized books[J]. Science, 2011, 331(6014): 176-182.

[60] MORALDO H H. An approach for text steganography based on markov chains[J]. arXiv preprint arXiv:1409.0915, 2014.

[61] YANG Z, GUO X, CHEN Z, et al. RNN-stega: Linguistic steganography based on recurrent neural networks[J]. IEEE Transactions on Information Forensics and Security, 2018, 14(5): 1280-1295.

[62] YANG Z, ZHANG S, HU Y, et al. VAE-Stega: linguistic steganography based on variational auto-encoder[J]. IEEE Transactions on Information Forensics and Security, 2020, 16: 880-895.

[63] YANG Z, WEI N, LIU Q, et al. GAN-TStega: Text steganography based on generative adversarial networks[C]//Proceedings of the 18th International Workshop on Digital Watermarking, November 2-4, 2019, Chengdu, China. Cham: Springer, 2020: 18-31.

[64] KANG H, WU H, ZHANG X. Generative text steganography based on LSTM network and attention mechanism with keywords[J]. Electronic Imaging, 2020, 2020(4): 291-1-291-8.

[65] YANG Z, ZHANG P, JIANG M, et al. Rits: Real-time interactive text steganography based on automatic dialogue model[C]//Proceedings of the 4th International Conference on Cloud Computing and Security, June 8-10, 2018, Haikou, China. Cham: Springer, 2018: 253-264.

[66] YANG Z, GONG B, LI Y, et al. Graph-Stega: Semantic controllable steganographic text generation guided by knowledge graph[J]. arXiv preprint arXiv:2006.08339, 2020.

[67] 薛一鸣,周雪婧,周小诗等.基于图像描述的文本信息隐藏[J].北京邮电大学学报,2018,41(06):7-13.

[68] FANG T, JAGGI M, ARGYRAKI K. Generating steganographic text with LSTMs[J]. arXiv preprint arXiv:1705.10742, 2017.

[69] ZIEGLER Z, DENG Y, RUSH A M. 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, November 3-7, 2019, Hong Kong, China. Stroudsburg: ACL, 2019: 1210-1215.

[70] SHEN J, JI H, HAN J. Near-imperceptible neural linguistic steganography via self-adjusting arithmetic coding[J]. arXiv preprint arXiv:2010.00677, 2020.

[71] DAI F Z, CAI Z. Towards near-imperceptible steganographic text[J]. arXiv preprint arXiv:1907.06679, 2019.

[72] SANH V, DEBUT L, CHAUMOND J, et al. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter[J]. arXiv preprint arXiv:1910.01108, 2019.

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

[74] OSBORNE M. Statistical machine translation[M]. Cambridge: Cambridge University Press, 2009.

[75] BROWN P F, DELLA PIETRA S A, Della Pietra V J, et al. The mathematics of statistical machine translation: Parameter estimation[J]. Computational Linguistics, 1993, 19(2): 263-311.

[76] CHO K, VAN MERRIËNBOER B, BAHDANAU D, et al. On the properties of neural machine translation: Encoder-decoder approaches[J]. arXiv preprint arXiv:1409.1259, 2014.

[77] LUONG M T, PHAM H, MANNING C D. Effective approaches to attention-based neural machine translation[C]//Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, September 17-21, 2015, Lisbon, Portugal. Stroudsburg: ACL, 2015: 1412-1421.

[78] SHINYAMA Y, SEKINE S. Paraphrase acquisition for information extraction[C]// Proceedings of the 2nd International Workshop on Paraphrasing, July 11, 2003, Sapporo, Japan. Stroudsburg: ACL, 2003: 65-71.

[79] DONG L, MALLINSON J, REDDY S, et al. Learning to paraphrase for question answering[C]//Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, September 9-11, 2017, Copenhagen, Denmark. Stroudsburg: ACL, 2017: 875-886.

[80] FADER A, ZETTLEMOYER L, ETZIONI O. Paraphrase-driven learning for open question answering[C]//Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, August 4-9, 2013, Sofia, Bulgarian. Stroudsburg: ACL, 2013: 1608-1618.

[81] LIN T, MAIRE M, BELONGIE S, et al. Microsoft coco: Common objects in context[C]// Proceedings of the 13th European Conference on Computer Vision, September 6-12, 2014, Zurich, Switzerland. Cham: Springer, 2014: 740-755.

[82] FEDERMANN C, ELACHQAR O, QUIRK C. Multilingual whispers: Generating paraphrases with translation[C]//Proceedings of the 5th Workshop on Noisy User-generated Text, November 4, 2019, Hong Kong, China. Stroudsburg: ACL, 2019: 17-26.

[83] KINGMA D P, BA J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.

[84] BELKINA A C, CICCOLELLA C O, ANNO R, et al. Automated optimized parameters for T-distributed stochastic neighbor embedding improve visualization and analysis of large datasets[J]. Nature Communications, 2019, 10(1): 5415.

[85] WILSON A, KER A D. Avoiding detection on twitter: embedding strategies for linguistic steganography[C]//Proceedings of the 2016 Media Watermarking, Security, and Forensics, February 14-18, 2016, San Francisco, USA. Oxford: Ingenta, 2016: 1-9.

[86] PAVLICK E, RASTOGI P, GANITKEVITCH J, et al. PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification[C]//Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, July 26-31, 2015, Beijing, China. Stroudsburg: ACL, 2015: 425-430.

[87] 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.

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

作者在攻读硕士学位期间公开发表的论文

[1] **Yang T,** Wu H, Yi B, et al. Semantic-preserving linguistic steganography by pivot translation and semantic-aware bins coding[J]. IEEE Transactions on Dependable and Secure Computing, 2023. (CCF A 类期刊)

作者在攻读硕士学位期间所作的项目

[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

在本篇论文完成之际，我想由衷感谢在硕士研究生阶段关心和帮助过我的人。

首先，我想感谢我的导师张新鹏教授。张老师总是对研究方向有预见性的看法和独到的见解，也为我在分析和解决问题的思路上树立了榜样。此外，随着研究的深入和对研究课题理解的加深，我更加意识到张老师所提供的实验室环境和计算资源来之不易。优越的研究条件也使得我能够更加专注于科研工作，并更加高效地对课题的各种可能进行探索和尝试。

其次，我想感谢同样为我的科研工作提供极大帮助的吴汉舟老师。吴老师在我的研究方向选择上提供了针对性的建议，也对我的研究课题给予了耐心的指导和多处细节上的建议，在方案设计细节和论文撰写等方面不断提出优化的措施。我在硕士期间所取得的成果也离不开吴老师的付出和努力。

再次，我想感谢实验室的易标师兄。师兄与我研究方向相近，在与师兄对课题的探讨中，我也更加快速地对核心问题有了更深刻的理解。正因为师兄的无私分享，让我在课题研究上少走了不少弯路。

之后我想感谢实验室的魏诗语、唐雄、郑晓燕、柳琦云等同门和其他朋友们。尽管大家的研究方向和内容不尽相同，但大家在科研工作和日常生活中互相建议、互相鼓励、共同进步，使得平日的研究氛围更加轻松愉快。

然后也感谢我的父母，感谢父母一如既往地给予我在物质和精神层面上的支持并尊重我做出的各个决定。

最后感谢各位评审专家和老师，感谢您在百忙之中审阅这篇论文！