武汉大学国家网络安全学院

信息隐藏项目结题报告

项目名称 基于检索与生成方法结合的文本隐写

指导教师 任延珍

小组成员 张斌龙、赵伯侯、刘竞优

目录

一、 项目介绍

（一）项目内容

Hi-stega 框架 [1] 的秘密消息嵌入过程分为数据信息过程层和控制信息嵌入层。社交网络视为一个巨大的实时更新的语料库 C。上层的主要任务是在社交网络环境中检索秘密消息 m，找到合适的上下文数据载体 m（确保其中尽可能多的单词）。对秘密消息和上下文数据载体进行处理以获得控制信息，控制信息包括索引 i（秘密消息中的单词在上下文数据载体中出现的位置）和未出现的单词 k（命名关键字集合）。

在下层，需要对上层返回的信息进行处理，通过格式协议将其转化为比特流 b。比特流嵌入是在以上下文数据载体作为模型输入生成文本的过程中进行的，生成过程是使用 k 来引导的，以确保秘密消息中可能未被上下文数据载体覆盖的一些单词将出现在生成的隐写文本中。同时，将生成的文本的语义定向到相关方向，增强语义连贯性。Hi-stega 框架算法如下图所示

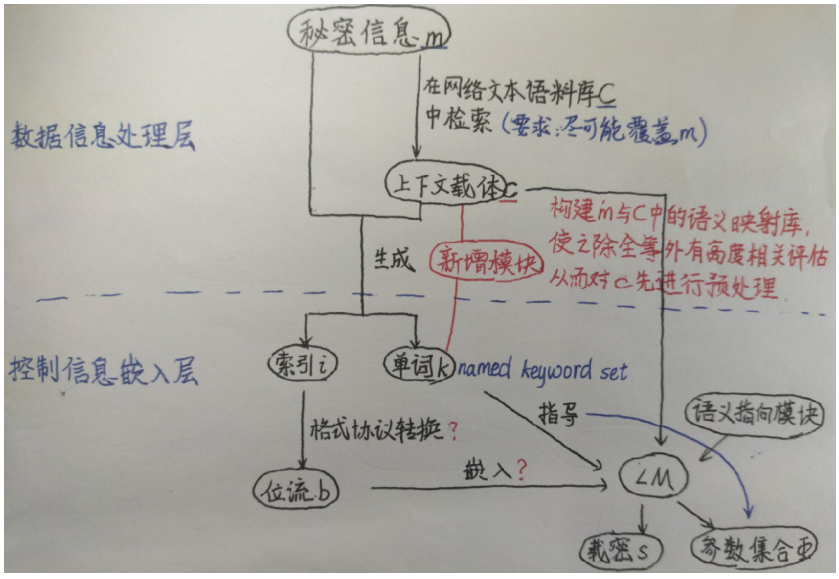


图 1: Hi-stega 框架

（二）预期目标

为了获取具有一定相关性的使用到的数据集为新闻数据集，将新闻的标题作为将要隐写的密文，将新闻的所有段落的内容作为信息隐写语料库 C，最终实现将新闻的标题嵌入到新闻的段落中的一个文本隐写。

（三）小组成员及分工

1.3.1 张斌龙

主要负责部分代码的注释分析，以及代码的修正运行，部分数据的简单提取函数编写。

1.3.2 赵伯侯

主要负责 utility 相关提取关键词代码的注释分析、实验报告编写

1.3.3 刘竞优

encoder，decoder 相关编码解码方法分析、答辩 ppt 制作

二、 相关背景

在信息隐藏领域，现在已经发展出了诸如以图像、音频、视频、文本等一些媒体形式作为载体的隐写方法，但是受传输通道的影响，图像、音频、视频等一些媒体形式的载体在传输过程中可能会失真或者受到噪声影响，在各种载体中，文本隐写术受传输渠道影响较小，具有较高的鲁棒性和实用性，特别适合社交网络场景。

目前为止较为热门的文本隐写方法有三种，分别为基于修改的方法、基于检索的方法和基于生成的方法。

基于修改的方法 [2] 通过词汇修改文本内容来嵌入信息。这些方法通常具有较低的嵌入率并且倾向于改变词频，这使得它们更容易受到隐写分析技术的影响。

基于检索的方法 [3] 首先对文本语料库中的样本进行编码，然后通过将样本与秘密消息映射来选择相应的句子进行传输。这些方法需要对文本语料库进行编码和构建，容量相对较低。然而，由于文本载体本身没有改变，基于检索的方法很难被现有的隐写分析方法检测到。

使用神经网络生成文本 [4]。这些方法可以在一定程度上改善嵌入有效负载，但生成的文本在语义上是随机的，导致对某些隐写分析方法无效。基于生成的方

法的灵活性，包括适应任何上下文和任何场景的能力，尚未得到充分探索和利用。

文本隐写领域中，目前为止较为新颖的研究框架是使用 Hi-stega 进行文本隐写。

三、 项目内容

（一）主要框架

实现 Hi-Stega 层次隐写方法的主要程序伪代码如下所示

算法 1 秘密信息嵌入算法

输入: 某个输入秘密消息 m、社交网络文本语料库 C、语言模型 LM
输出: 隐写文本 stego、隐写参数 ϕ

function DataInformationProcessing(m,c)
    在 C 中检索，找到 c 可以覆盖 m 中尽可能多的单词；
    在 c 中记录这些词的索引 i =（i1,i2,...,ip）；
    if $m_i \in m \ and \ m_i \notin c$ **then**
        将 mi 添加到关键字集 k
    end if
    return 控制信息 k、i 和数据载体 c
end function
function Control Information Embedding(k,i,c)
    通过格式协议将 i 转换为比特流 b
    while 生成过程并未结束 **do**
        $W_j \leftarrow LM(c, k, b)$
        if $w_j \in k$ **then**
            从 k 中移除 w_j
            将索引 j 添加到参数集 Φ 中；
        end if
    end while
    return 隐写文本 stego、隐写参数 ϕ
end function

（二）数据预处理

3.2.1 数据清洗

由于原始数据集中并不是所有的字段在实验中都需要，所以需要对原始数据集进行清洗和预处理等工作。通过调用 TweetClean.py 代码、checker 函数对数据中存在的表情符号进行情感分析、缩写词进行同义展开、与语义无关的字符的删除等数据清洗工作，方便模型进行识别与读取

3.2.2 提取主体数据字典

该部分的功能在 news_Search.py 中进行实现，主要是将经过清洗之后的原始数据集 test.data 中存在的标题数据以及相关的内容进行提取，得到 test_title.txt 文件。从而获得到将要隐写到载体中的密文信息然后需要对原始数据集中需要用到的数据进行筛选与提取，将提取得到的各项参数以 jsonl 格式保存到 test_data_clean_for_test_all.jsonl 文件中。

在本次实验中原作者给定的数据集有三个，复现过程中为了节省时间采用了较小的一个 test.data 文件进行训练。

3.2.3 数据提取结果

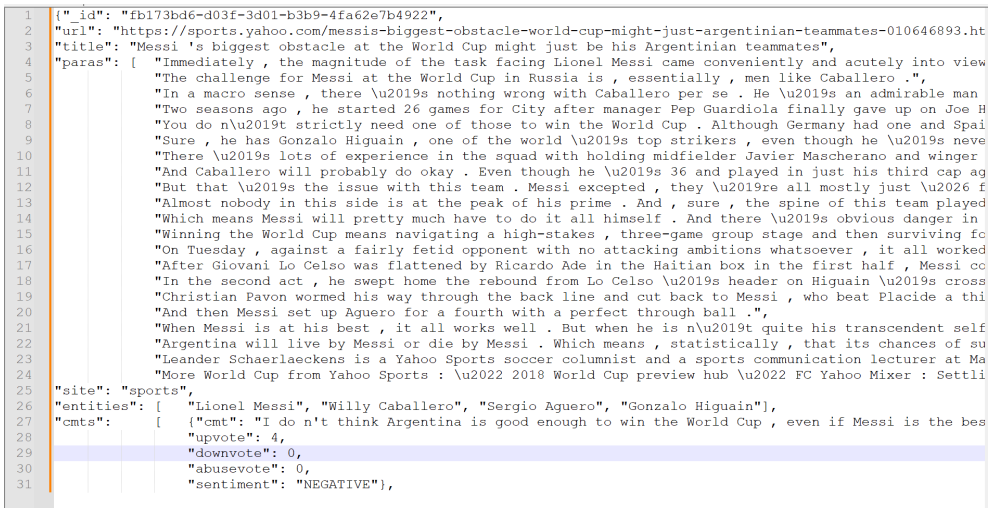
在运行预处理程序的过程中会出现由于数据集中混杂了一些其他编码的数据，导致分词时代码会报错，报错信息如下图所示

```
Traceback (most recent call last):
  File "D:\Hi-Stega\code\dataProcess\news_Search.py", line 425, in <module>
    search_news_fast_json(text=text)
  File "D:\Hi-Stega\code\dataProcess\news_Search.py", line 391, in search_news_fast_json
    tokenize_input = tokenizer.tokenize(final_line['paras'][0])
TypeError: string indices must be integers
```

图 2: 预处理过程报错信息

因为不是正常的字符。所以我们的处理只能是裁剪原始数据集。

在本次实验中所使用的原始数据集的结构如下图所示



```
{"_id": "fb173bd6-d03f-3d01-b3b9-4fa62e7b4922",
"url": "https://sports.yahoo.com/messis-biggest-obstacle-world-cup-might-just-argentinian-teammates-010646893.ht
"title": "Messi 's biggest obstacle at the World Cup might just be his Argentinian teammates",
"paras": [  "Immediately , the magnitude of the task facing Lionel Messi came conveniently and acutely into view
            "The challenge for Messi at the World Cup in Russia is , essentially , men like Caballero .",
            "In a macro sense , there \u2019s nothing wrong with Caballero per se . He \u2019s an admirable man
            "Two seasons ago , he started 26 games for City after manager Pep Guardiola finally gave up on Joe H
            "You do n\u2019t strictly need one of those to win the World Cup . Although Germany had one and Spai
            "Sure , he has Gonzalo Higuain , one of the world \u2019s top strikers , even though he \u2019s neve
            "There \u2019s lots of experience in the squad with holding midfielder Javier Mascherano and winger
            "And Caballero will probably do okay . Even though he \u2019s 36 and played in just his third cap ag
            "But that \u2019s the issue with this team . Messi excepted , they \u2019re all mostly just \u2026 f
            "Almost nobody in this side is at the peak of his prime . And , sure , the spine of this team played
            "Which means Messi will pretty much have to do it all himself . And there \u2019s obvious danger in
            "Winning the World Cup means navigating a high-stakes , three-game group stage and then surviving fo
            "On Tuesday , against a fairly fetid opponent with no attacking ambitions whatsoever , it all worked
            "After Giovani Lo Celso was flattened by Ricardo Ade in the Haitian box in the first half , Messi co
            "In the second act , he swept home the rebound from Lo Celso \u2019s header on Higuain \u2019s cross
            "Christian Pavon wormed his way through the back line and cut back to Messi , who beat Placide a thi
            "And then Messi set up Aguero for a fourth with a perfect through ball .",
            "When Messi is at his best , it all works well . But when he is n\u2019t quite his transcendent self
            "Argentina will live by Messi or die by Messi . Which means , statistically , that its chances of su
            "Leander Schaerlaeckens is a Yahoo Sports soccer columnist and a sports communication lecturer at Ma
            "More World Cup from Yahoo Sports : \u2022 2018 World Cup preview hub \u2022 FC Yahoo Mixer : Settli
"site": "sports",
"entities": [   "Lionel Messi", "Willy Caballero", "Sergio Aguero", "Gonzalo Higuain"],
"cmts":     [   {"cmt": "I do n't think Argentina is good enough to win the World Cup , even if Messi is the bes
                 "upvote": 4,
                 "downvote": 0,
                 "abusevote": 0,
                 "sentiment": "NEGATIVE"},
```

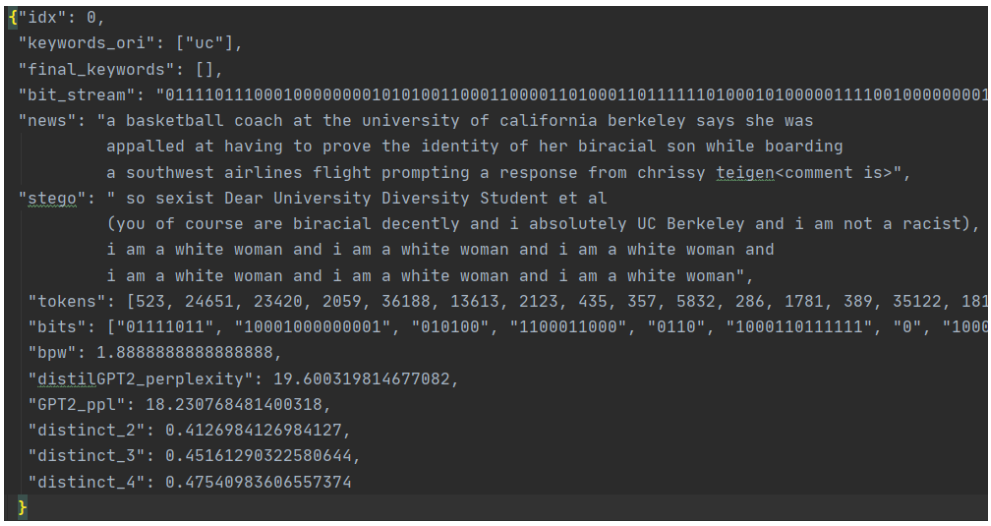

图 3: 原始数据集结构

原始数据集中单项的数据结构如下表所示：

表 1: 原始数据集结构

数据集字段	含义
id	唯一标识符
url	新闻来源的网址
title	新闻标题
paras	新闻文章段落数组
site	新闻发布的网站或来源
entities	新闻中提到的主要实体
cmts	用户评论数组

经过数据清洗和关键数据的提取操作之后得到的数据字典结果如下图所示



```
{"idx": 0,
 "keywords_ori": ["uc"],
 "final_keywords": [],
 "bit_stream": "01111011100010000000010101001100011000011010001101111110100010100000111100100000000
 "news": "a basketball coach at the university of california berkeley says she was
         appalled at having to prove the identity of her biracial son while boarding
         a southwest airlines flight prompting a response from chrissy teigen<comment is>",
 "stego": " so sexist Dear University Diversity Student et al
         (you of course are biracial decently and i absolutely UC Berkeley and i am not a racist),
         i am a white woman and i am a white woman and i am a white woman and
         i am a white woman and i am a white woman and i am a white woman",
  "tokens": [523, 24651, 23420, 2059, 36188, 13613, 2123, 435, 357, 5832, 286, 1781, 389, 35122, 181
  "bits": ["01111011", "10001000000001", "010100", "1100011000", "0110", "1000110111111", "0", "1000
  "bpw": 1.8888888888888888,
  "distilGPT2_perplexity": 19.600319814677082,
  "GPT2_ppl": 18.230768481400318,
  "distinct_2": 0.4126984126984127,
  "distinct_3": 0.45161290322580644,
  "distinct_4": 0.47540983606557374
}
```


图 4: 预处理结果

（三）构建 **keywords**

因为在文本隐写过程中需要关键词来指导信息的隐写操作，在数据预处理步骤已经实现了对于原始数据集部分相关数据的提取，在构建 keywords 时调用 get_jsonl_wo_unk_v3 函数对数据集中提取出的关键词进行筛选并保存在新的 jsonl 文件中并且将关键词映射到对应嵌入向量后生成.pkl 文件中。最后得到的关键词数据列表 keyword_sets 如下图所示

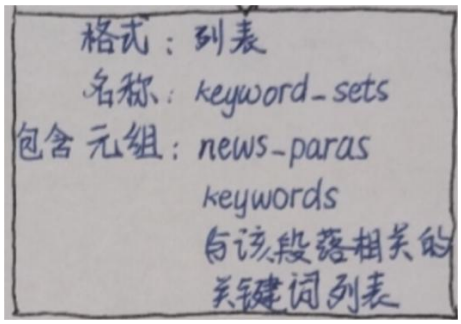


图 5: 关键词构建结果

（四）构建模块

在隐写过程中需要使用到 GPT-2 模型进行构建，在使用大模型进行构建的过程中需要一定的语义指向模块，提示学习模块以及 converter_table_glove.npy 文件对于隐写生成结果进行指导。

首先需要经过 coverter_table_glove 函数将 GPT-2 词汇表中的单词转换为 glove 嵌入表示的转换表得到 GPT-2 生成的指导文件。

语义指导模块 kG 的作用是通过调整 p_θ 使得语义能够接近

$k = \{k_1, k_2, k_3, \ldots, k_n\}$

提示学习模块 prompt 在 GPT-2 模型控制生成文本的过程中也会起到知道模型生成的文本更加流畅的功能。

（五）隐写核心

在本次实验中所使用的隐写框架为首先根据秘密信息在文本语料库中选取与秘密信息最契合的文本作为隐写的载体信息，然后根据秘密信息在上下文载体中

的存在与否来生成单词数组 keyword_set 和索引 i，然后将索引 i 的格式转换为比特流保存到变量 bit_stream 中, 将得到的比特流和经过编解码器编码后的单词数组传递给 GPT-2 模型即可生成出载密 S 以及参数集合体 φ

（六）结果评估

3.6.1 计算相似度

该部分主要调用 get_sim 函数来完成，该函数接收编码后的关键词的列表以及 GPT 模型生成的关键词列表以及数据预处理步骤中得到的呃转换表，以及一系列的参数之后计算输入的两组关键词之间的相似度最后将计算得到的相似度的值保存到 sim 中进行返回

3.6.2 评估文本质量

因为需要判断 GPT 模型生成的句子的质量来修改构建的指导模块，所以需要对于文本进行评价，通过调用 evaluate_quality 系列函数，设 sequence 为文本序列，word 为特定单词，perplexity 为文本复杂度，guide 为布尔值，temp 为温度参数。

评估文本质量的原理为：

计算可以得到的质量分数

$Q = \frac{1}{\text{temp}} \cdot \exp(-(w_1 \cdot \text{word\_count} + w\_3 \cdot \text{perplexity}))$

若采用线性方式对文本质量进行评估，perp 为布尔值。其原理为：

$Q_{\text{linear}} = \frac{1}{\text{temp}} \cdot (\text{word\_count} \pm w_3 \cdot \text{perplexity})$ 通过这种评估质量的算法可以准确的指导 GPT 模型生成质量更加高的文本

3.6.3 计算句子困惑度

为了评价生成模型的质量，需要衡量语言模型性能的指标，衡量模型对测试数据的预测能力。

在本次实验中采用的计算句子困惑度的数学方法为：

$\text{PPL}(S) = \exp\left(\frac{1}{N}\sum_{i=1}^{n}\text{NLL}(s_i)\right)$ 根据产生的结果判断每个使用的模型的预测

能力。较低的困惑度通常意味着模型对数据有更好的预测能力

3.6.4 计算平均比特数

在实验过程中，因为需要对隐写后的信息进行编码处理，所以应当计算隐写后 stegos 中 tokens 的平均比特数来评价不同的编码方式的效果。在本次实验中调用 bpw 系列函数计算平均比特数从而得到信息编码效率来评价不同的编码方式

3.6.5 计算文本多样性

使用大模型生成的文本可能具有语言结构和内容上比较单一的特性，因此需要对生成的结果计算多样性，在本次实验中调用 distinct_n 函数对文本的多样性进行计算：

定义 n-gram 的集合 G 为从文本 example 中生成的所有 n-gram 的集合。对于给定的 n，n-gram 可以定义为：

$$G = \{(x_i, x_{i+1}, \ldots, x_{i+n-1}) \mid x_i \in example, 1 \leq i \leq (len(example) - n + 1)\}$$

然后对于每一个 n-gram 都有 $g \in G$，更新不同 n-gram 的数量 $n_{distinct}$ 和总的 n-gram 数量 n_{total} 这可以表示为

$$n_{distinct} = |\{g \mid g \in G, count(g) = 1\}|$$

$$n_{total} = \sum_{g \in G} count(g)$$

四、 项目成果

（一）最终成果

运行程序对 yahoo 新闻 [5] 中得到的数据集进行隐写操作，为了保证隐写信息和载体的高度相关性，采用将新闻的标题隐写到文本中的方法，最终得到的隐写结果如下图所示

```
{"idx": 0,
 "keywords_ori": ["uc"],
 "final_keywords": [],
 "bit_stream": "01111011100010000000010101001100011000011010001101111110100010100000111100100000000
 "news": "a basketball coach at the university of california berkeley says she was
          appalled at having to prove the identity of her biracial son while boarding
          a southwest airlines flight prompting a response from chrissy teigen<comment is>",
 "stego": " so sexist Dear University Diversity Student et al
          (you of course are biracial decently and i absolutely UC Berkeley and i am not a racist),
          i am a white woman and i am a white woman and i am a white woman and
          i am a white woman and i am a white woman and i am a white woman",
  "tokens": [523, 24651, 23420, 2059, 36188, 13613, 2123, 435, 357, 5832, 286, 1781, 389, 35122, 181
  "bits": ["01111011", "10001000000001", "010100", "1100011000", "0110", "1000110111111", "0", "1000
  "bpw": 1.8888888888888888,
  "distilGPT2_perplexity": 19.600319814677082,
  "GPT2_ppl": 18.230768481400318,
  "distinct_2": 0.4126984126984127,
  "distinct_3": 0.45161290322580644,
  "distinct_4": 0.47540983606557374
 }
```


图 6: 隐写结果

该结果使用传统的基于生成的方法进行隐写操作，将 keyword 作为大模型隐写的参考对象之一，使 GPT 模型依照 keyword 和一个随机生成的比特流序列以及一系列的语义指导模块进行隐写文本的生成

在使用本次实验中设计的 HiStega 模型进行隐写后得到的结果如下图所示


```
{"idx": 0,
 "news": "",
 "stego": "Repeatedly asleep in the car outside my house was Dr Doeed who violently
           pulled the trigger and started shooting what looked to be him as he drove
           from one house to another as he drove around in the double van he received
           the call from within the car as they were moving quickly to get more information
            as they tracked sightings",
 "tokens": [47541, 515, 306, 16039, 287, 262, 1097, 2354, 616, 2156, 373, 1583, 2141, 2308, 508, 258
 "bits": ["", "1", "000001010100110", "0", "00", "1010", "0001100", "0", "0", "00110", "11", "110100
 "bpw": 3.8412698412698414,
 "distilGPT2_perplexity": 69.60819131616492,
 "GPT2_ppl": 50.65197668171003,
 "fine_tune_ppl": 50.65197668171003,
 "distinct_2": 0.8412698412698413,
 "distinct_3": 0.967741935483871,
 "distinct_4": 1.0
}
```


图 7: 隐写结果

观察该结果可以发现生成的语句较为通顺，在其中也包含有一定的关键词信息，直接观察可以看出，本次实验中所采用的模型相较于传统的文本生成隐写模型有较大的优化

（二）代码结构

本次实验中采用的一些主体功能代码如下图所示，该代码包含了模型框架隐写过程的整个流程，以及所有的功能代码并且包含部分隐写方式的运行脚本

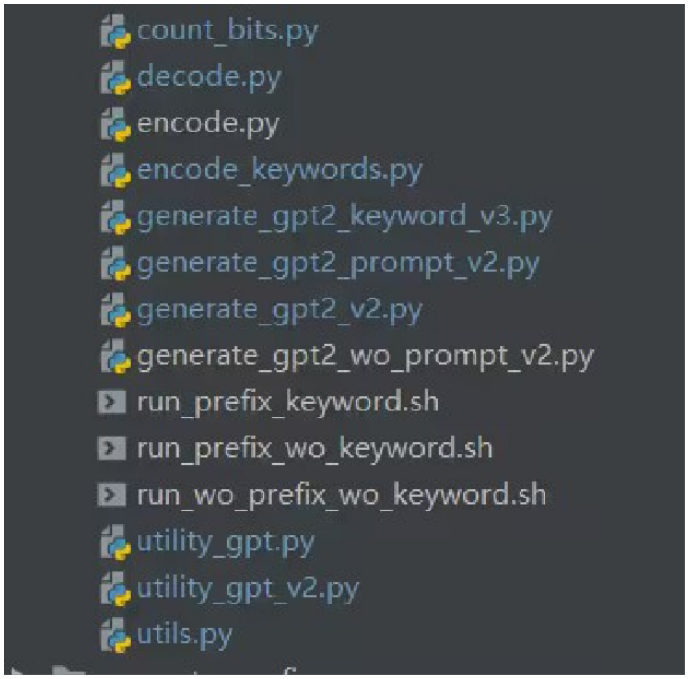


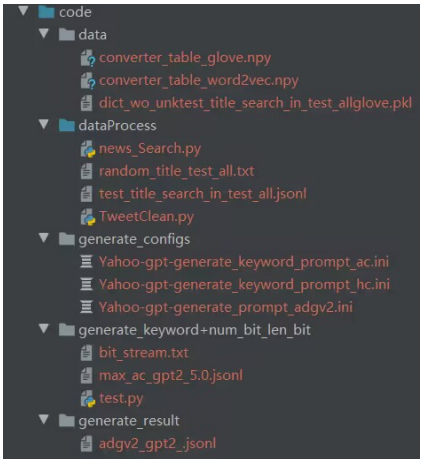
图 8: 代码结构

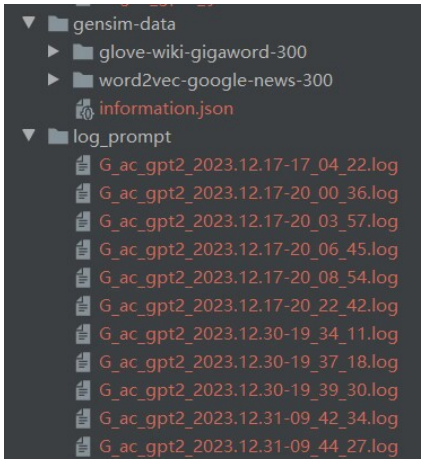
本次实验中所采用的数据集文件以及数据集经过预处理过程产生的中间结果文件如下图所示，其中也包括因为未知原作者对数据集的处理方式而自己编写的数据预处理程序

图 9: 代码结构

本次实验中所使用到的一些依赖文件保存到了/data 目录下，其中预处理所用到的程序代码和文件保存到/dataProcess 下配置文件保存到/generate_configs 目录

下；生成的结果保存到了 generate_keyword+num_bit_len_bit 目录下只使用 GPT-2 模型进行隐写文本生成的结果保存到了 generate_result 目录下；实验中用来观察隐写过程并调试的日志文件存放到 log_prompt 下。具体的目录结构如下图所示





由于附上源码时并不像破坏整个目录结构的完整性，但整个文件过大，所以做了如下操作：

1. 将 Hi-Stega/yahoo_news_release 中的 dev.data 和 train.data 删除

2. 将 Hi-Stega/code/ginsim-data 中两个词汇转换表中内容清空，因为运行代码能拉取

这些改动不会影响代码运行，目的是为了减少压缩包的大小，因为这些表项都是大文件

（三）性能评估

4.3.1 预期目标

在本次实验中希望达到的预期目标是选取一个秘密信息能够在文本语料库中选取一个载体对秘密信息进行嵌入操作，并且在得到嵌入后的信息之后也能够将嵌入后的信息提取出来，但是在复现的过程中由于在生成的文本中嵌入使用的是 GloVe 模型，所以在创建嵌入向量文件 pkl 时引入的也是 GloVe 模型，在提取过程中也使用该 pkl 文件。按理来说是个简单的逆向使用，但是实际上代码报错。查询原因无果后推测是我们在运行代码时可能漏执行了某些关键代码。其中.pkl 文件是由 encode_keywords.py 中 create_enc_dict 函数生成。在程序中出现的报错信息如下

图所示

图 10: pkl 解析报错信息

create_enc_dict 函数的大致结构为创建字典映射从 CSV 文件中提取的关键词到它们在 GloVe 词嵌入（word embedding）中的向量表示。但是由于代码缺乏完整性，并且存在有众多未注释的不同版本的 create_enc_dict 系函数，所以导致在处理 pkl 文件时并不能对 pkl 文件进行正确的解析

4.3.2 编码方式

因为在信息隐写的过程的数据编码的方式也会对各个性能指标造成直接的影响，所以在本次实验中采用了霍夫曼编码 HC [6]、算术编码 AC [7] 和自适应动态分组 ADG [8] 三种编码方式使用相同的隐写算法进行信息隐写，计算隐写后文本的困惑度用于评价文本的流畅性，计算文本的多样性，得到的结果如下表所示 通

表 2: 编码方式不同对结果影响

编码方式	流畅性（文本困惑度）	多样性
霍夫曼编码	18.23	0.45
算术编码	20.44	0.47
自适应动态分组	19.32	0.44

过比较得到的文本流畅性和多样性的评价指标可以发现，编码方式的不同确实会对文本困惑度和文本的多样性造成一定的影响，但是在数值上的影响并不大。在三种编码方式中算术编码的文本困惑度要高于其他两种编码方式也就是说其文本的流畅性较低，但其多样性数值相较于其他两种编码方式较高。

由此可见霍夫曼编码的文本困惑度最低即其文本的流畅程度很高，且其多样性也处在较高水平；算术编码的流畅性最低，但是其多样性在三种编码方式中处于较高位置；自适应动态分组则介于二者之间。

因此得出结论，虽然三种不同的编码方式在流畅性和多样性上存在些许差别，但是差别的值较小，不会影响实验结果。

4.3.3 隐写方式

将程序运行得到的关于生成文本流畅性和多样性的数值统计得到的结果如下表所示 观察生成的结果可以看出，采用 HiStage 模型生成的隐写后的文本的文本

表 3: 隐写算法不同对结果影响

隐写算法	文本困惑度	$distinct_2$	$distinct_3$	$distinct_4$	bpw
传统隐写	18.23	0.41	0.45	0.47	1.88
HiStega 隐写	50.65	0.84	0.96	1.0	3.84

困惑度较高，文本更加流畅，并且该模型生成的文本三次计算多样性也处在一个较高的水平，该模型的编码效率明显高于传统的编码方式

（四）局限性

1.generate_gpt2_keyword_v3.py 应该生成的依赖的文件编码方式为 AC，但是在后续 generate_gpt2_v2.py 生成隐写文本时使用 ADG_V2 编码方式，最终产生意义不明的隐写文本。并且在程序运行的过程中还会出现有大量的报错信息如下图所示

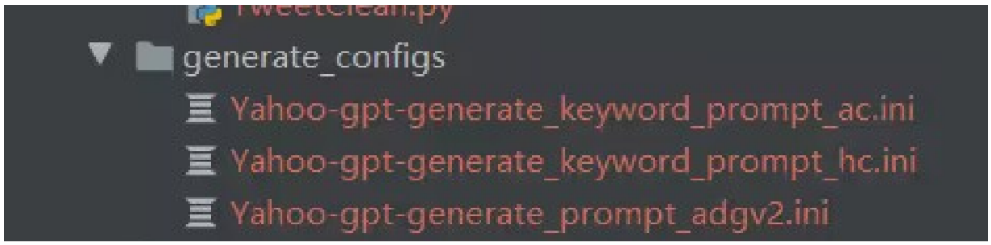


图 11: 报错信息

2. 源代码有大量注释代码段，同时没有明确的注释来说明这些代码段的作用，由此导致代码分析过程中有很多函数未被使用，这可能是导致实验结果不符合预期的部分原因。以下是 utils.py 文件中注释掉的部分代码。

```
if __name__ == '__main__':
    compute_ppl(model_name_or_path="gpt2", sentences=[])
    # bpw_jsonlines("/data/lastness/KE-dataset/tweet/ac/stegos-encoding.jsonl")

    # sample_data("/data/lastness/LCCC-base/corpus.txt", sample_num=1000000, do_shuffle=True)
    # for dataset in ["tweet"]:
    #     cover_file = "/data/lastness/corpus/{}.txt".format(dataset)
    #     for alg in ["ac",]:
    #         stego_file = "generation/{}-gpt/{}/stegos-encoding.jsonl".format(dataset,alg)
    #         output_dir = "generation/{}-gpt/{}".format(dataset, alg)
    #         sample_from_txt_and_jsonl(cover_file, stego_file, output_dir, max_num=10000, do_sample=True)
```

图 12: 大量注释代码段

3. 在运行 generate 系列函数中出现如下报错显示列表超过索引。这个报错的原因未知，按照道理在裁剪数据集后应该没有这样的错误，小组讨论后推测的结果可能是在预处理时得出的数据存在问题。

```
Traceback (most recent call last):
  File "D:\Hi-Stega\code\generate_gpt2_keyword_v3.py", line 454, in <module>
    generate(args, configs)
  File "D:\Hi-Stega\code\generate_gpt2_keyword_v3.py", line 339, in generate
    pred_word, predicted_text, predicted_index, = get_prediction(
  File "D:\Hi-Stega\code\utility_gpt.py", line 48, in get_prediction
    pred_word = predicted_text.split()[-1].split('<|endoftext|>')[-1]
IndexError: list index out of range
```

图 13: generate 报错

4. 该模型的隐写过程非常依赖于文本语料库，如果能够有非常充足的文本语料库，那么隐写的结果将会非常好，但是在现实中并不存在无限大的并且存在与秘密信息匹配程度完美契合的文本作为载体，因此在实际隐写过程中会造成隐写结果并不理想的情况。

5. 在实验过程中因为时间有限，只采取了单个数据集进行信息隐写，并没有采用不同网络情境下的数据集进行多方位的测试，使得实验得出的结果可能存在片面的可能。

6. 同样因为实验时间有限，没有对该模型进行较为全面的隐写分析，没有针对该模型的抗隐写分析能力进行讨论和研究。

五、 总结及展望

（一）总结

在本次实验中，主要重心在于对于原始项目的复现上，并且由于论文较新所以原作者的实验代码并不全面造成复现中缺少关键步骤导致复现出的结果并不全面

（二）展望

对关键词集 K 进行提前预处理，使得在隐写的过程中 k 的数据量能够更小，方便在嵌入模块中指导模型的嵌入，并且能够抵抗隐写分析

对比特流生成和嵌入的过程进行更加有效的编码方式，使得模型的嵌入效率能够更高，使得模型具有更强的携带秘密信息的能力

更换不同社会网络情境下的数据集，统计该模型在不同的数据集下的表现。

对隐写模型进行多种隐写分析测试，分析该隐写模型的抗检测能力和鲁棒性

参考文献

[1] Wang H, Yang Z, Yang J, et al. Hi-Stega: A Hierarchical Linguistic Steganography Framework Combining Retrieval and Generation[C]//International Conference on Neural Information Processing. Singapore: Springer Nature Singapore, 2023: 41-54.

[2] Chang, C.Y., Clark, S.: Practical linguistic steganography using contextual synonym substitution and a novel vertex coding method. Comput. Linguist. 40(2), 403–448 (2014)

[3] Chen, X., Sun, H., Tobe, Y., Zhou, Z., Sun, X.: Coverless information hiding method based on the Chinese mathematical expression. In: Huang, Z., Sun, X., Luo, J., Wang, J. (eds.) ICCCS 2015. LNCS, vol. 9483, pp. 133–143. Springer, Cham (2015).

[4] Xiang L, Wang R, Yang Z, et al. Generative Linguistic Steganography: A Comprehensive Review[J]. KSII Transactions on Internet & Information Systems, 2022, 16(3).

[5] Yang, Z., Xu, C., Wu, W., Li, Z.: Read, attend and comment: a deep architecture for automatic news comment generation. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 50775089 (2019)

[6] Yang, Z.L., Guo, X.Q., Chen, Z.M., Huang, Y.F., Zhang, Y.J.: RNN-Stega: linguistic steganography based on recurrent neural networks. IEEE Trans. Inf. Forensics Secur. 14(5), 1280–1295 (2018)

[7] Ziegler, Z., Deng, Y., Rush, A.M.: Neural linguistic steganography. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLPIJCNLP), pp. 1210–1215 (2019)

[8] Zhang, S., Yang, Z., Yang, J., Huang, Y.: Provably secure generative linguistic steganography. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pp. 3046–3055 (2021)