南 开 大 学

本科生毕业论文（设计）

中文题目：**文本分类模型攻击方法的研究和实现**

外文题目：Research and Implementation of Text Attack Methods for Text Classification

学　　号：2011188

姓　　名：邵琦

年　　级：2020 级

专　　业：计算机科学与技术

系　　别：计算机科学与技术

学　　院：计算机学院

指导教师：陈晨　副教授

完成日期：2023 年 5 月

关于南开大学本科生毕业论文（设计）的声明

本人郑重声明：所呈交的学位论文，是本人在指导教师指导下，进行研究工作所取得的成果。除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人创作的、已公开发表或没有公开发表的作品内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。本学位论文原创性声明的法律责任由本人承担。

学位论文作者签名：

年　　月　　日

本人声明：该学位论文是本人指导学生完成的研究成果，已经审阅过论文的全部内容，并能够保证题目、关键词、摘要部分中英文内容的一致性和准确性。

学位论文指导教师签名：

年　　月　　日

摘 要

随着自然语言处理（NLP）技术的快速发展，特别是深度学习技术的引入，文本分类任务取得了显著进展。预训练语言模型的飞速发展显著地提升了文本分类的性能。然而，这些模型在面对精心设计的文本攻击时，其安全性和鲁棒性面临威胁。攻击者可能利用模型的漏洞，通过添加特定的扰动来欺骗模型，导致误判或错误分类。

本文提出了一种新的文本攻击方法，该方法利用基于梯度的方式生成通用触发器 tokens，并且用基于词替换策略生成对抗性样本，并将二者相结合，此外，还探讨了基于大型语言模型生成对抗性样本，旨在提高文本攻击的准确率和效率，并增强生成对抗性样本的欺骗性。本文通过在多个预训练语言模型和多个数据集上的攻击实验，验证了所提方法的有效性和通用性。

本文的研究表明，预训练语言模型在文本分类任务时，在文本攻击面前存在一定的安全性问题，而提高模型的鲁棒性对于应对文本攻击至关重要。通过深入理解文本攻击的本质和预训练语言模型的特点，我们可以为提高模型的安全性和鲁棒性提供新的思路和方法。本研究不仅为文本攻击研究提供了新的方法和方向，也为 NLP 技术的发展和应用做出了贡献。

关键词：深度学习；自然语言处理；预训练语言模型；文本分类；文本攻击

Abstract

With the rapid development of natural language processing (NLP) technology, especially the introduction of deep learning technology, text classification tasks have made remarkable progress. The rapid development of pre-trained language models has significantly improved the performance of text classification. However, these models face threats to their security and robustness in the face of well-designed text attacks. An attacker may exploit a vulnerability in the model by adding specific perturbations to deceive the model, resulting in a miscalculation or misclassification.

This paper proposes a new text attack method, which uses a gradience-based approach to generate universal trigger tokens, and uses a word-based substitution strategy to generate adversarial samples, and combines the two. In addition, it also discusses the generation of adversarial samples based on large language models, aiming to improve the accuracy and efficiency of text attacks. And enhance the deception of generating adversarial samples. In this paper, the effectiveness and universality of the proposed method are verified by attack experiments on multiple pre-trained language models and multiple data sets.

The research in this paper shows that the pre-trained language model has certain security problems in the face of text attacks in the text classification task, and improving the robustness of the model is crucial to cope with text attacks. By deeply understanding the nature of text attacks and the characteristics of pre-trained language models, we can provide new ideas and methods to improve the security and robustness of the models. This study not only provides a new method and direction for text attack research, but also contributes to the development and application of NLP technology.

Key Words: deep learning; natural language processing; pre-trained language model; text classification; text attack

目 录

第一章　绪论

自然语言处理（NLP, Natural Language Processing）作为人工智能领域的一个关键领域，旨在让计算机能够理解、处理和生成自然语言。近年来，随着深度学习技术迅速发展，尤其是预训练语言模型（例如 BERT、GPT 等）的引入，NLP 领域取得了巨大进步。这些预训练语言模型通过在大规模文本数据上的预训练，学习了丰富的语言表示能力，这使得它们能够在各种 NLP 任务上表现出色，包括文本分类、语言生成、问答等。

在 NLP 中，文本分类是一项核心任务，它旨在将文本数据分为不同的类别或标签，使得计算机能够自动识别和组织大规模文本数据，为信息检索、情感分析、垃圾邮件过滤等应用提供支持。近年来，随着深度学习技术的飞速发展，预训练语言模型的引入彻底改变了传统方法的局面，许多基于预训练语言模型的文本分类方法取得了巨大成功，大大提升了分类性能和效果。传统的文本分类方法通常需要手工设计特征或者依赖于大量标记数据，但是预训练语言模型能够自动学习语言表示，无需大量标注数据即可完成任务。这一模型的引入大大提升了文本分类的性能和效果，使得文本分类在实际应用中更加可行和有效。

随着预训练语言模型在自然语言处理任务中的广泛应用，人们越来越关注这些模型的安全性和鲁棒性。特别是在文本攻击领域，这些模型正面临着前所未有的挑战。攻击者可能会发现并利用模型的弱点，或引入特定的扰动来误导模型，进而导致误判或错误分类。因此，研究预训练语言模型在面对文本攻击时的安全性和鲁棒性变得至关重要。

本文将在预训练语言模型广泛使用的背景下，深入研究文本分类模型在面对文本攻击时的安全性和鲁棒性问题。本文设计了一种新的攻击方法，它结合了通用触发攻击和词替换策略攻击，并在多个数据集上进行了实验，以证明其攻击效果和性能。本文希望通过这项研究，揭示现有文本分类模型的潜在漏洞，为提高模型的鲁棒性提供新的思路和方法，从而更好地应对文本攻击带来的挑战。

第一节　研究背景及意义

近年来，对抗攻击技术在计算机视觉领域取得了显著的成就，如图1.1所示，当将噪声加到一张原来可被 CNN 图像分类模型正确识别的熊猫图片后，尽管两张图片几乎无法以人眼识别，但还是欺骗了分类系统，错误地将其识别为长臂猿。并且对抗攻击技术在计算机视觉领域得到了广泛的应用，促进了其在其他

领域的快速发展，尤其是在自然语言处理领域。然而，与计算机视觉领域相比，离散语义文本给实现对抗带来了额外的限制和挑战。文本数据的复杂性和多义性使得攻击者难以有效地识别并利用模型的弱点，也难以从数据集本身角度进行对抗攻击。

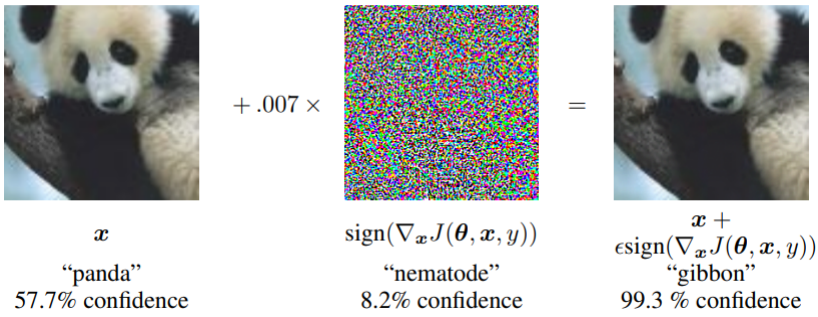

图 1.1 对抗攻击技术在计算机视觉领域的典型案例[1]

随着 NLP 技术的发展，研究人员开始意识到自然语言处理模型在面对文本攻击时的脆弱性。文本攻击不仅可以干扰文本分类模型的分类能力，还可能对模型的性能和可靠性造成严重影响。因此，理解文本攻击的本质以及开发有效的防御机制变得至关重要。

此外，预训练语言模型的广泛应用使得研究文本攻击更加迫切。这些模型在大规模语料库上进行了预训练，具有丰富的语言表示能力，因此成为文本分类任务的首选模型。然而，正是因为这种丰富的语言表示能力，预训练语言模型也更容易受到文本攻击的影响。因此，研究了解对抗攻击是如何进行的是很重要的以及如何提高预训练语言模型的鲁棒性，成为当前 NLP 研究的一个重要课题。

在这样的背景下，深入探究文本分类模型在文本对抗攻击面前的安全性与鲁棒性显得尤为重要。理解文本数据与图像数据在对抗性攻击中的不同表现，能够帮助研究人员更深刻地洞察到文本攻击所独有的难题。其次，对文本数据进行深入的攻击策略研究，特别是那些针对预训练模型的攻击，不仅能够推动自然语言处理技术的发展，也能为其他研究领域带来创新的思路和方法。此外，提高文本分类模型的安全性和鲁棒性有助于提高模型的安全性和鲁棒性，从而推动其在实际应用中的广泛应用和发展。

因此，本研究旨在探究预训练语言模型在面对文本攻击时的安全性和鲁棒性问题，从而为提高文本分类模型的可靠性和实用性提供新的思路和方法。通过深入研究文本攻击的本质以及预训练语言模型的特点，研究人员有望发现有效的防御策略，并为 NLP 技术的发展和应用做出更大的贡献。

第二节　国内外研究现状

近年来，深度学习逐渐成为了自然语言处理的主流方法，基于深度学习的自然语言处理也已大规模应用于相关场景中，如智能检索、智能音响、机器翻译、情感分类等等。同时，安全性和鲁棒性的相关问题在自然语言处理方面显得十分重要，一些微小的改变，便可能导致深度学习的结果发生剧烈的变化。因此，研究人员开始着力于探索暴露深度学习鲁棒性问题的相关领域，提出了对抗攻击。

对抗攻击是指通过精心设计微小扰动，使得机器学习模型产生错误预测结果的行为。在图像识别领域，对抗性样本研究已取得显著进展，并得到了广泛的应用。Szegedy 等人[2] 提出了针对图像分类器的对抗示例概念，表明当前智能系统存在巨大的安全风险。他们指出，通过在纯图像上添加微小扰动生成的对抗性示例可能会使性能良好的分类器输出错误的预测。更值得注意的是，在不同数据集或不同结构上训练的深度神经网络可以对相同的对抗性示例产生相同的错误分类。

此后，对抗攻击技术成为了人工智能领域的研究热点之一，并促进了其在其他领域，尤其是自然语言处理领域的快速发展。

但在文本领域，该研究尚处于相对初级阶段。近年来，研究者们开始关注文本分类模型的脆弱性，并提出多种攻击方法，揭示了模型的潜在弱点。然而，这些方法存在局限性，如攻击效率低、难以适应多样化文本分类任务等。

Papernot 等人[3] 首先研究了对文本的对抗攻击。受到生成对抗性图像的想法的启发，他们通过与文本嵌入相关的正向导数来生成对抗性样本。此后，为了探索 NLP 中的安全盲点并寻求相应的防御策略，学者们从对抗攻击和防御的角度对 NLP 进行了深入的研究。最初，为了保持文本的语义一致性和语法正确性，研究人员通过引入特定的独特方法，将图像域的对抗技术转移到文本域，如贪婪搜索算法[4] 和强化学习 (RL)[5][6]。之后，一些研究人员考虑到，由于文本是由离散的符号组成的，他们开始在文本域提出特殊的对抗攻击。一开始，[7][8][9][10] 的工作利用特定的技巧来改变给定文本中的几个字符或单词。这些方法很好地保持了语义一致性和句法正确性，但生成的对抗性样本普遍缺乏多样性。之后，一些工作[11][12][13] 侧重于在保持文本语义和语法的同时，操纵整个文本以产生更多样化的对抗性样本。这些方法需要更仔细地设计，以确保攻击能力和对抗性样本的质量。

此外，研究人员还试图从模型的内部机制出发，探索文本分类模型在面对

对抗性样本时的决策过程。通过分析模型的注意力分布、梯度信息等，他们试图了解模型做出分类决策的依据，从而设计出更有效的攻击策略。然而，这些研究往往需要深入了解模型的内部结构，很难应用于黑盒模型或对抗模型的隐私保护场景。

第三节　研究内容及创新点

本文分析了文本分类模型下的文本攻击研究的背景、意义以及现状，阐述了该研究现阶段存在的一些问题和改进方向。

针对现阶段文本攻击方法的一些问题，本文提出并设计了一种新的文本攻击方法。该方法结合了通用触发器攻击和词替换策略攻击，不仅可以针对某一个特定的分类模型、特定的数据集进行攻击，还可以将其对抗性样本应用于不同的分类模型和数据集，具有较强的通用性和普适性，可以很大程度上提升文本攻击的攻击性能和攻击效率。

本文所涉及的创新点如下：

1. 相较于主流的文本攻击方法，本文提出的攻击方法更加灵活、适应性更强。同时，能够克服传统攻击方法中对特定领域的依赖性，使得这种文本攻击方法能够在不同模型、不同数据集中实现有效的攻击，还能够应对数据分布的变化和模型结构的差异。其灵活性和适应性使得它在面对不同场景和应用时都能够表现出色，为文本攻击领域的研究和应用带来了新的思路和可能性。
2. 本文提出的文本攻击方法能够生成具有高攻击强度但仍保持语义一致性的对抗性样本，减少了生成的对抗性样本被检测到的可能性。这种文本攻击方法不仅能成功欺骗目标模型，而且使得对抗性样本与原始样本在语义上保持一致，增加了其迷惑性和欺骗性。这使得攻击者能够更有效地绕过文本分类模型的防御机制，从而提高了攻击的成功率和可靠性。
3. 本文提出的文本攻击方法尝试了利用大型语言模型进行生成文本对抗性样本，探索了大型语言模型在生成文本对抗性样本方面的潜力。大型语言模型具有更为强大的语言生成能力，能够生成高度逼真的对抗性样本，相较于主流的文本攻击方法，生成的对抗性样本语言风格更为自然流畅，更具有欺骗性。此外，这种方法还具有高度的灵活性，可以轻松生成各式对抗性样本，从而满足不同攻击场景的需求。
4. 本文提出的文本攻击方法相较于主流的文本攻击方法更为轻便，生成对

抗性样本更为迅速，面对一个陌生的文本分类模型和数据集能够立即生成文本对抗性样本，大大提高了文本攻击的效率。

本文提出了一种新的文本攻击方法，该方法不仅弥补了现有方法的一些缺陷，而且提高了攻击效果和性能，为文本攻击领域提供了新的思路和方法。

第四节　文章组织结构

本文共分为五个章节，各章节内容安排如下所述：

1. 第一章首先介绍了文本分类任务下的文本攻击的背景和意义，强调了文本攻击对于提升分类模型的安全性和鲁棒性问题的重要意义。之后，说明了文本攻击领域的研究现状。然后介绍了本文提出的文本攻击方法的研究内容与创新点，最后介绍了本文的论文组织结构。
2. 第二章深入探讨了本文相关研究的理论基础和实践背景，从文本分类任务、预训练语言模型、文本对抗攻击三个方面进行了详细阐述。通过这三个方面的介绍，本文提供了一个全面的研究背景，为理解本文后续章节中提出的研究问题、解决方案、实验研究奠定了坚实的基础。
3. 第三章从多个角度，详细探讨了本文提出的文本攻击方法。首先从问题定义、攻击原理方法和混合攻击方法的实现方法三个角度介绍了混合攻击的技术路线，之后更为详细地讲解了攻击方法的具体细节、相关策略和算法，并探讨了基于大型语言模型生成对抗性样本的方法，最后探讨了本攻击方法的迁移性。
4. 第四章对本文的文本攻击方法进行了实验分析。首先介绍了实验环境、实验设定、实验细节和评价指标部分。之后仔细设计了实验，在多个预训练模型和数据集上进行了实验并进行了实验分析。除此以外，还进行了对比实验，验证混合攻击的优越性；还进行了迁移实验，用以证明文本攻击方法的迁移性，并利用大型语言模型生成对抗性样本进行实验测试。最后进行了案例分析。
5. 第五章对本文提出的文本攻击方法及创新点和贡献做出了总结，分析并发现了当前方法存在的不足之处并对改进策略和未来研究方向进行了展望。最后对自然语言处理领域如何应对文本攻击进行了讨论分析。

第二章 相关研究基础

第一节 文本分类任务研究基础

文本分类（Text Classification）是自然语言处理领域一个经典的任务，是指计算机将含有信息的文本映射到事先确定的一个或多个类别或主题。目前，文本分类任务一般用机器学习或深度学习来解决。

文本分类任务基本原理如图2.1所示，具体来说是：

首先获取数据集，可以利用爬虫技术或页面处理获取文本数据。

之后进行文本预处理，需要将数据集中的文本转化成计算机可以处理的数据结构，也就是将文本切分成构成文本的语义单元，一般有分词、去停用词等操作。

然后进行文本特征提取，选出最能表征文本含义的词组元素，从而降低问题的规模，有助于分类性能的改善。

其次再进行文本表示，将非结构化的信息转化为计算机可以理解的结构化的信息，从而针对文本信息做计算，并完成分类任务。

最后交给分类器进行分类任务，在这里有传统的机器学习方法和深度学习模型。

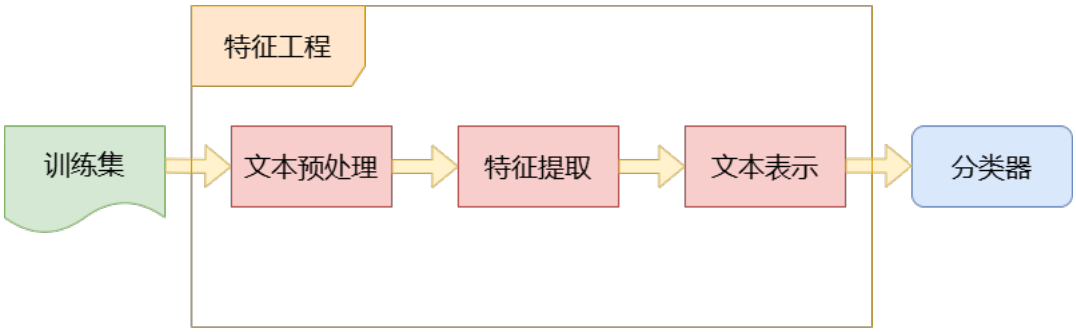

图 2.1 文本分类任务基本原理

传统的机器学习方法一般有基于规则的模型（例如决策树算法）、基于概率的模型（例如朴素贝叶斯算法）、基于几何学的模型（例如支持向量机（SVM））、基于统计的模型（例如 k 近邻（K-Nearest Neighbor，KNN）模型）。

近年来，深度学习发展迅速，卷积神经网络（CNN）[14]、循环神经网络（RNN）[15]、长短期记忆网络（LSTM）[16]、注意力机制（Attention）[17] 等等技术的发展，显著提升了文本分类任务的性能和效果。

第二节 预训练语言模型基础

在当前的自然语言处理领域中，预训练语言模型已经成为了研究和应用的主流。通过深入分析其独特的结构和训练方式，人们可以更好地理解预训练模型在文本表示学习和各种 NLP 任务中的优势和局限性。同时，对于攻击性研究来说，理解预训练语言模型的内部机制和对抗性特征也是至关重要的，这有助于研究人员设计更具针对性和有效性的对抗攻击策略。

考虑到实验以 BERT（Bidirectional Encoder Representations from Transformers）[18]、以及其相关衍生模型 RoBERTa（Robustly optimized BERT pre-training approach）[19]、ALBERT（A Lite BERT）[20] 作为模型背景进行攻击，本文将以这几种预训练模型为例进行介绍。

BERT 是由 Google 开发的一种自然语言处理模型，它在大型文本语料库上进行训练，形成通用的“语言理解”模型，然后将该模型应用于下游 NLP 任务（如问答）。它是第一个用于预训练 NLP 的无监督、深度双向系统，在理解语言上下文时能够更为全面，因此，BERT 在各种 NLP 任务中都有出色的表现，如情感分析、文本分类、自然语言推理等等。它是 Transformer[17] 模型中的 encoder 部分，通过 Pre-training 和 Fine-tuning 两个步骤能够实现问题回答、语言推理等任务，有效的提升了自然语言处理任务。

BERT 针对输入的自然语言序列，需要进行数据的预处理操作，其中包括 Token、Segment 和 Position 层，如图2.2所示。经过这三层预处理，输入模型的自然语言序列将成为含有自身编码信息、单词所属句子编码信息和单词位置编码信息的词向量序列。

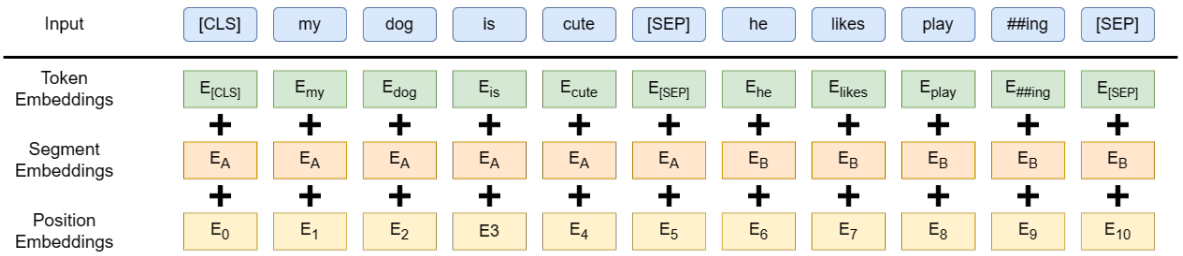

图 2.2 BERT 输入示例表示

其中，Token Embeddings 层主要是对输入的自然语言序列进行分词处理及词向量嵌入。Seqment Embeddings 层主要是标记词向量序列中的每个单词属于哪一句话。Position Embedding 层加入的是句子中每个单词的位置信息。经过

Embedding 层的处理，BERT 将输入的自然语句转换成了包含其自身含义、所属句子关系及单词在句子中位置信息的词向量序列。通过 Embedding 层，每个单词都被映射到一个高维向量空间中，而这个向量表征包含的不仅仅是单词的语义信息，还考虑了单词在上下文中的位置关系和句子结构。

除此以外，BERT 模型框架中包含了 Pre-training（预训练）和 Fine-tuning（微调）两个部分。在 Pre-training 阶段，BERT 模型会以自监督学习的方式，在大规模的未标记数据集上预训练，从而更好地学习自然语言的相关知识和语境信息。和传统的从左到右或从右到左的语言训练模型不同，BERT 模型的预训练使用了深层双向训练法，使其在未标记数据上进行无监督训练。同时，BERT 模型的训练引入了 Masked LM（遮盖语言模型）和 Next Sentence Prediction（NSP）机制。

其中，Masked LM 机制的设计是为了让模型在未标记的数据上进行无监督的训练。这个机制会随机屏蔽掉输入序列中的单词，要求模型进行预测。这样促使模型学习到更好的语言表示，同时策略性地替换和保留单词来减少预训练和微调间的不匹配问题。引入 Next Sentence Prediction 机制目的是来学习句子级别的语义关系，通过预测两个句子之间是否存在逻辑连接来提升模型对文本序列间逻辑关系的理解和处理能力。

而 Fine-tuning 阶段会针对特定的任务，采用迁移学习的策略，根据具体的下游任务需求，在有标记的数据上对模型的最后几层进行微调，以使得模型能够更好地适应具体任务的特征和语境。这种迁移学习的方式使得 BERT 模型用途广泛，可用于各种不同的自然语言处理任务。而不需要针对每个任务重新训练一个新的模型。

ALBERT 与 RoBERTa 均是基于 BERT 所衍生的预训练语言模型，其中 ALBERT 相较于 BERT，通过共享层间的参数和嵌入向量来减少模型大小，从而使得模型更轻量化，旨在减少 BERT 模型的参数数量的同时，保持或提高模型的性能。RoBERTa 在 BERT 的基础上进行了一些优化，其中包括更长的训练时间、更大的数据集、动态掩码机制等，以提高模型的鲁棒性和性能。因此，RoBERTa 在多个自然语言处理任务上均取得了显著的性能提升。

第三节　文本对抗攻击的分类

在计算机视觉领域，对抗攻击可以根据其攻击特点和攻击效果，分为黑盒攻击与白盒攻击、一次攻击与迭代攻击、目标攻击与非目标攻击以及特定扰动

和普遍扰动等。自然语言处理领域虽然有所不同，但同样可以借鉴，因此，一般可以将现有的文本对抗攻击从以下三个维度进行分类：

1. 指向性

根据指向性，一般情况下可以将文本攻击分为两类，分别为：指向性攻击（Targeted Attack）和通用性攻击（Universal Attack）。其中指向性攻击能够引导受害模型做出特定错误判断，比如，文本攻击可以生成对抗性样本，使得分类模型将其误判为特定类别；相反，通用性攻击，也就是非指向性攻击，仅需要生成对抗性样本导致模型进行错误判断，无需特定类别。

2. 受害模型可见性

此外，还能够依据对受害模型的可见性将攻击分为白盒攻击（White-Box）和黑盒攻击（Black-Box）两种。

一般情况下，在白盒攻击中，攻击者对于攻击模型完全了解，包括其模型结构和模型参数。攻击者能够直接调用模型并且获得给定输入的相关参数和输出结果。在白盒攻击中，攻击者一般使用基于梯度生成对抗性样本的方法，这被称为基于梯度的攻击（Gradient-Based Attack）。

相对地，在黑盒攻击中，攻击者无法获取攻击模型的模型结构和模型参数，只能通过调用模型的方法来获取给定输入的输出结果。在黑盒攻击下，攻击者一般采用基于分数的攻击（Score-Based Attack）和基于决策的攻击（Decision-Based Attack）。前者通过获取攻击模型的输出分数（如分类模型的各类概率）来生成对抗性样本，而后者通过获取攻击模型的输出结果（如分类模型给出的类别）。

在某些情况下，攻击者甚至可能无法直接调用受害模型进行攻击，这种情况被称为盲攻击（Blind Attack）。

3. 扰动粒度

根据扰动产生的粒度，还可以将对抗攻击分为句级攻击、词级攻击和字级攻击。

句级攻击将整个原始输入句子视为扰动的对象，旨在生成一个在语义上与原始输入相同（至少对于当前任务的真实标签不产生变化），但会改变受害模型判断的对抗性样本。常见的句级攻击方法包括改写[21][22]、编码后重新解码[23]、添加无关句子[24] 等。

词级攻击则是针对原始输入中的单词进行扰动，其中最主要的方

法是词替换。替换词的选择包括基于词向量相似度[9][25]、同义词[26]、义原[27]、语言模型分数[28] 等。此外，还有研究尝试添加或删除单词[29]，但这通常会影响生成的对抗性样本的语法和通顺性。

字级攻击主要针对原始输入中的字符进行扰动，常见方法包括字符添加、删除、替换、交换顺序等。对于字替换，有随机替换[7]、基于 One-Hot 编码的字替换[30] 以及基于字形相似度的替换[31] 等方法。

此外，一些攻击方法也会同时进行词级和字级扰动[32]。

表2.1即为一些典型的文本对抗攻击方法以及其分类。

表 2.1 一些典型的文本对抗攻击方法

模型方法	可见性	扰动类型	主要思想、方法
SEA[21]	Decision	句	基于规则的复述
SCPN[22]	Blind	句	复述
GAN[23]	Decision	句	基于编码-解码的文本生成
TextFooler[25]	Score	词	基于贪心的词替换
PWWWS[26]	Score	词	基于贪心的词替换
Genetic[9]	Score	词	基于遗传算法的词替换
FD[3]	Gradient	词	基于梯度下降的词替换
TextBugger[32]	Gradient/Score	词 + 字	基于贪心的词和字扰动
UAT[33]	Gradient	词/字	基于梯度下降的词或字扰动
HotFlip[30]	Gradient	词/字	基于梯度下降的词或字替换
VIPER[31]	Blind	字	形近字替换
DeepWordBug[32]	Score	字	基于贪心的字扰动

第四节 本章小结

本章介绍了与本文研究内容相关的一些研究和一些基础概念。第一节介绍了文本分类任务的研究基础，其中包含文本分类任务的基础概念、基本原理和目前主流的实现方法；第二节以 BERT、RoBERTa、ALBERT 为例介绍了预训练语言模型的相关知识，介绍了相关概念并深入探讨了预训练语言模型的特殊结构和独特的预训练、微调机制。第三节则是从三个层面介绍了文本对抗攻击的分类，并提出了一些典型的文本对抗攻击方法和其类型。通过本章，可以了解在预训练语言模型下文本分类任务的文本攻击的基础概念。

第三章　文本分类模型攻击方法

第一节　混合攻击技术路线

3.1.1　问题定义

目前的主流的文本对抗攻击技术，有些需要访问语言模型，并通过获取其内部参数（如梯度）计算并生成对抗性样本，这样的攻击效率较低，同时对于计算机性能要求过于严格；还有些更侧重于攻击数据集本身，这样生成的对抗性样本虽然效率更高，但是存在语义一致性问题和攻击性能过低的问题。通用触发器攻击和词替换策略是两种主流的文本攻击方法，虽然这些攻击方法已经被广泛研究和应用于文本攻击中，但它们在一些方面也存在一些缺陷。

通用触发器攻击首先存在限制攻击性能的问题，目前主流的通用触发器攻击可能只在特定类型的文本数据上表现良好，对于其他类型的数据可能表现不佳。换句话说，即使是“通用触发器”，也只是在某个特定类型的数据集上通用。例如，一个在情感分类文本上表现良好的触发器可能对于新闻分类文本无效。其次，通用触发器攻击还存在可检测性问题，一般情况下，通用触发器生成的触发词相较于自然语言有很大差距，这导致了通用触发器攻击会较容易被发现。一旦通用触发器攻击被发现，它可能容易被模型检测到，并加以防御。这可能导致攻击者需要不断改进攻击策略，以应对模型的改进。通用触发器还存在硬件配置要求过高的问题，主流的通用触发器攻击往往需要高性能显卡进行长时间的梯度计算等操作，这使得其在一些硬件配置不高、资源受限或配置较低的环境下难以生成对抗性样本。

词替换策略攻击存在语义一致性的问题，替换文本中的词语可能导致生成的文本在语义上不一致，从而影响其可读性和真实性。这可能导致生成的对抗性样本被人类很容易识别出来，降低了攻击的成功率。其次，还存在攻击效率过低的问题，有些词替换策略可能不够强大，无法有效地欺骗目标模型。过于简单的替换可能会导致生成的对抗性样本与原始样本之间的相似性太高，使攻击无效。

因此，尽管文本分类模型的攻击方法研究取得了一定进展，但在提高攻击效率和普适性，以及深入理解模型决策机制等方面仍面临诸多挑战。本文致力于研究和实现一种新的文本攻击方法。该方法相较于主流的文本攻击方法具有

更强的攻击性能和效率，生成的对抗性样本欺骗性更强。同时，本方法具有很强的灵活性和通用性，能够在不同的模型和数据集上进行攻击。此外，本方法还可以利用大型语言模型进行生成文本对抗性样本，具有更强的攻击效率，生成的文本对抗性样本也更加逼真，欺骗性更强。

3.1.2 文本攻击原理

在文本攻击中，可以得到一个模型 f，一个由 tokens（单词、子词或字符）t 组成的文本输入、一个目标标签 $\tilde{y}$。

本方法的攻击目标是将触发器 tokens t_{adv} 连接到 t 的前面或末尾，甚至是中间随机某个位置，并且对于文本输入 t 进行同义词替换攻击为 t'，如：

$$f\left(t_{adv};t'\right)=\tilde{y} \tag{3.1}$$

本方法主要工作是对触发器 tokens t_{adv} 和词替换文本输入 t' 进行优化，以最小化目标类 $\tilde{y}$ 对于来自数据集的所有输入的损失。

这可以转化为以下目标：

$$arg\min_{t_{adv},t'} E_{t\sim\tau}\left[\mathcal{L}\left(\tilde{y},f\left(t_{adv};t'\right)\right)\right] \tag{3.2}$$

其中 τ 是来自数据分布的输入实例，$\mathcal{L}$ 是任务的损失函数。为了生成攻击，可以假设对 f 进行白盒访问。

3.1.3 混合攻击实现方法

首先，需要对预训练模型进行白盒访问：在这个阶段，本文假设可以完全访问预训练模型的内部结构和参数，了解预训练模型在处理数据集时的梯度参数等信息以及决策过程和结果。通过这种白盒访问，可以全面掌握模型在不同输入下的行为，从而为后续生成对抗样本提供重要的基础信息。

之后，在特定数据集上对预训练模型进行微调，并对测试集进行测试。微调操作能够使预训练模型更好地适应特定领域的数据，提高模型在该领域的表现。在微调阶段，我们使用特定的数据集对模型进行训练和验证，以确保模型能够在该领域的任务上达到较高的准确性和鲁棒性。

在实现微调操作后，利用基于梯度的方法，生成相应的通用对抗触发器 tokens 并生成相应的对抗性样本进行文本攻击。具体而言，基于梯度的方法通过计算模型在输入文本上的梯度，识别出对模型决策影响最大的词或短语，从而用

于生成通用触发器 tokens。之后对目标数据集利用基于词替换策略生成对抗性样本进行文本攻击。词替换策略通过将原始文本中的关键词替换为其同义词或近义词，使得文本的语义保持一致，但却能够迷惑模型，使其做出错误的预测。

结合上述两种方法，本文提出了一种新的混合文本攻击方法，如图3.1所示。该方法的实现过程如下：

1. 白盒访问与微调：完全访问预训练模型，获取其内部结构和参数信息。在特定数据集上微调预训练模型，使其在该领域的任务上表现最佳。
2. 基于梯度攻击生称通用对抗触发器 tokens：利用梯度信息，识别对模型决策影响最大的词或短语。根据误导性最大的单词，生成通用触发器 tokens，并根据准确率是否降低进行迭代替换，直到找到最佳的通用对抗触发器 tokens。
3. 词替换策略生成对抗性样本：根据单词重要性排序识别目标数据集中的关键词，之后利用同义词替换这些关键词，生成对抗性样本。
4. 混合攻击结合：将通用触发器 tokens 插入到经过词替换的对抗性样本中，进一步增强对模型的攻击效果。这种混合方法不仅能在白盒预训练模型上实现良好的攻击效果，还能在黑盒预训练模型和陌生数据集上取得显著效果。

最后，本文尝试在多个预训练模型（如 BERT、ALBERT、RoBERTa）和多个数据集（如 IMDB、SST-2、SNLI）上测试上述综合攻击方法，论证其通用性和迁移性。实验结果表明，综合使用梯度攻击和词替换策略，不仅提高了对抗样本的有效性，还增强了对模型的攻击强度。这种方法在白盒和黑盒环境下均表现出良好的攻击效果，证明其在不同模型和数据集上的通用性和迁移性。

除此以外，本方法还探讨了基于大型语言模型生成通用触发器 tokens 和对抗性样本的方法，并将其与上述实现方法进行对比分析，指出了基于大型语言模型生成对抗性样本的可行性。

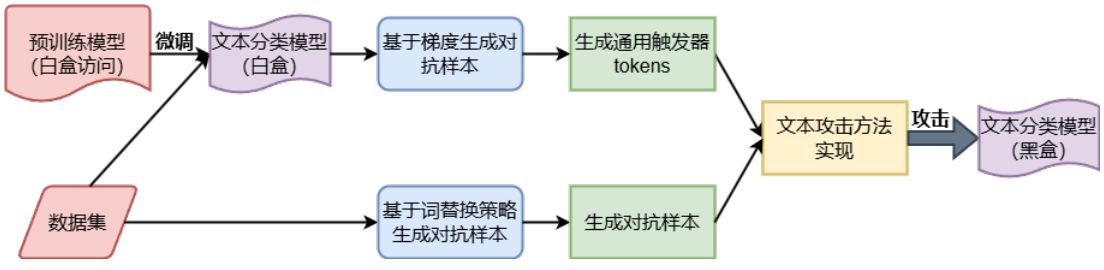

图 3.1 文本分类模型的攻击方法技术路线图

第二节 基于梯度生成通用对抗触发器

基于梯度生成通用触发器对抗性样本的实验实现步骤如图3.2所示，具体如下：

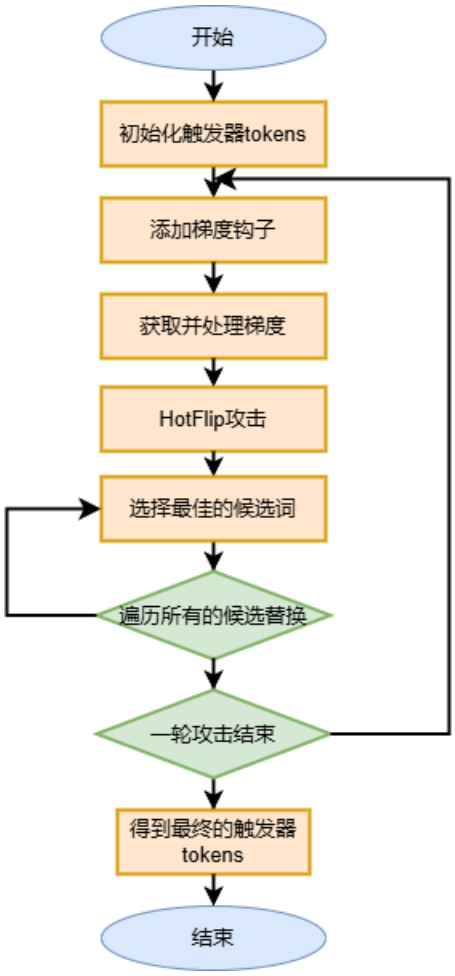

图 3.2 通用触发器攻击实现步骤

首先，本方法先初始化触发器 tokens，在测试数据上添加触发器 tokens 并且对模型进行评估，计算模型的准确率。然后对模型进行多轮攻击。

在每次的攻击中，本方法添加梯度钩子。用函数遍历模型的每个模块，当遇到 Embedding 层时，注册一个提取梯度的钩子。

之后，本方法获取梯度，计算模型在添加触发器 tokens 后的损失，并通过反向传播计算梯度。

然后移除梯度钩子，开始处理梯度。本方法将全局变量中存储的梯度求平均，并且返回位置对应的平均梯度。

之后，便可进行 HotFlip[30] 攻击。本方法利用 HotFlip 攻击函数，并替换新的 tokens。

最后一步便是选择最佳的候选词。在这一步，本方法实现函数迭代地替换触发器 tokens 中的单词，并寻找对准确率影响最大的词，最终返回最佳的触发器 tokens 和攻击后的准确率。

3.2.1 触发器搜索算法

首先，需要选择触发器长度并且初始化触发器序列，其中，本方法先通过重复单词“the”、子单词“a”或字符“a”来初始化序列，并将触发器连接到所有输入的前端、末端或者句子中随机位置。在实验中，本方法选择单词“the”连接前端进行初始化。

接下来，本方法迭代地替换触发器中的 tokens，以最大限度地减少批量示例中目标预测的损失值。对于如何替换当前的 tokens，计算机视觉领域提供了思路，但由于自然语言处理中，tokens 是离散的，不能直接应用。在实验中，本方法选择了以方法为基础的一种改进方法，这是一种梯度近似替换 tokens 的办法。为了应用该方法，本方法将表示为单热向量的触发器 tokens t_{adv} 嵌入到形式 e_{adv}。

3.2.2 HotFlip 算法

HotFlip 算法[30] 是一种用于生成针对神经网络模型的对抗性样本的技术，它的目标是通过对输入文本进行微小的修改，使得神经网络模型将其错误分类。

对于字符级别的文本分类器而言，输入形式如下所示：

$$x = [(x_{11},..x_{1n});..(x_{m1},..x_{mm})] \tag{3.3}$$

其中 n 是所有词中最长单词的长度，而 m 是一个样本中单词的个数，而每个 x 都是一个 V 维向量，V 是字符表的大小，也就是所有字符的总数。

HotFlip 算法中，将对于样本的修改操作视为一个个向量，利用损失函数对于向量操作的导数，就能知道哪种操作是能够让损失最大化的操作。同时，HotFlip 算法还提出可以通过看相应方向上的损失函数的变化来衡量输入 x 在操作矢量方向的变化对于损失函数的影响。其一阶近似值可以直接利用损失函数在相应方向上的分量获得：

$$\nabla_{\vec{v}_{ijb}} J(x,y) = \nabla_x J(x,y)^{\top} \cdot \vec{v}_{ijb} \tag{3.4}$$

这样的话，只需找到使损失函数增大速度最快的方向即可：

$$\max \nabla_x J(x,y)^\top \cdot \vec{v}_{ijb} = \max_{ijb} \frac{\partial J^{(b)}}{\partial x_{ij}} - \frac{\partial J^{(a)}}{\partial x_{ij}} \tag{3.5}$$

上式的梯度就可以替代损失函数最为修改操作的评价标准了，只要使上式的值最大，就能找到最佳的替代操作了。

插入操作也可以视为替换操作的一种，只要将插入位置的后一位字母替换为待插入字母，再将插入位置后字母依次后移即可，相应梯度如下式所示：

$$\max \nabla_x J(x,y)^\top \cdot \vec{v}_{ijb} = \max_{ijb} \frac{\partial J^{(b)}}{\partial x_{ij}} - \frac{\partial J^{(a)}}{\partial x_{ij}} + \sum_{j'=j+1}^{n} \left(\frac{\partial J^{(b')}}{\partial x_{ij'}} - \frac{\partial J^{(a')}}{\partial x_{ij'}}\right) \tag{3.6}$$

3.2.3 生成通用对抗触发器策略

本方法实现的梯度生成通用触发器对抗性样本以 HotFlip 算法为基础，利用梯度信息来找到对模型预测影响最大的 tokens，并进行替换，从而使得模型输出错误的结果。

从数学意义上来说，本方法更新每个触发器 tokens e_{adv_i} 的嵌入，计算触发器中各个 tokens 关于 e_{adv_i} 关于目标标签（想要模型预测的标签）的损失，反向传播得到每个 tokens 的梯度 $\nabla_{e_{adv_i}} \mathcal{L}$。遍历词表中所有 tokens 嵌入 e_i'，找出使得损失函数的一阶 Taylor 近似最小的 e_i'：

$$arg \min_{e_i' \in \nu} \left[e_i' - e_{adv_i}\right]^\top \nabla_{e_{adv_i}} \mathcal{L} \tag{3.7}$$

这个公式可以参考到 Burstein 等人[34] 提出的公式，如下所示，旨在希望输入变化小的同时输出变化大。

$$\arg\max_{1 \le i \le n, \hat{w} \in \nu} \left[\hat{W} - W_i\right]^\top \nabla W_i \mathcal{L}_{adv} \tag{3.8}$$

其中 e_i' 表示对抗性嵌入，它是一个替代于原始词 e_i 的词向量；ν 是模型词汇表中所有标记嵌入的集合；e_{adv_i} 代表原始词 e_i 的对抗性嵌入，它是一个被扰动过的版本，旨在欺骗 NLP 模型；$\nabla_{e_{adv_i}} \mathcal{L}$ 是每批任务损失的平均梯度，表示损失函数关于对抗性嵌入的梯度。

本方法的目标是找到一个新的嵌入 e_i'，与当前嵌入 e_{adv_i} 差异最小，同时沿着损失梯度的方向，也就是说，在保持与当前嵌入相似的同时，最大化损失函

数的减少，最小化损失函数 $\mathcal{L}$ 关于对抗性嵌入的梯度，从而最大化模型的预测损失，增加模型做出错误预测的概率。

然后对于 tokens 替换选用了 beam search，找到 k 个最小的候选词，替换掉当前的词，并且按照词的位置顺序迭代，最后在用 beam search 迭代到最后收敛为止。

具体来说，实现的 tokens 替换策略包括以下步骤：

首先，通过获取梯度的相关函数获取模型在当前输入文本中添加 tokens 后的梯度信息。

然后，本方法通过处理梯度的相关函数将所有样本的梯度进行平均计算，获取每个样本的平均梯度。

接下来，利用 HotFlip 攻击函数，将平均梯度与词嵌入矩阵相乘，得到每个位置上每个单词对损失的贡献。然后根据是否增加损失，选择对损失贡献最大的单词作为攻击目标，找到最具误导性的单词，替换为新的 tokens。

最后，本方法将候选替换的单词替换到 tokens 中，评估模型在测试数据集上的准确率，并且根据是否降低准确率的情况，选择保留或替换。并进行迭代，找到最佳的替换策略。

如图3.3即为 BERT 预训练模型上 IMDB 数据集文本分类任务的通用触发器 tokens 的替换过程。

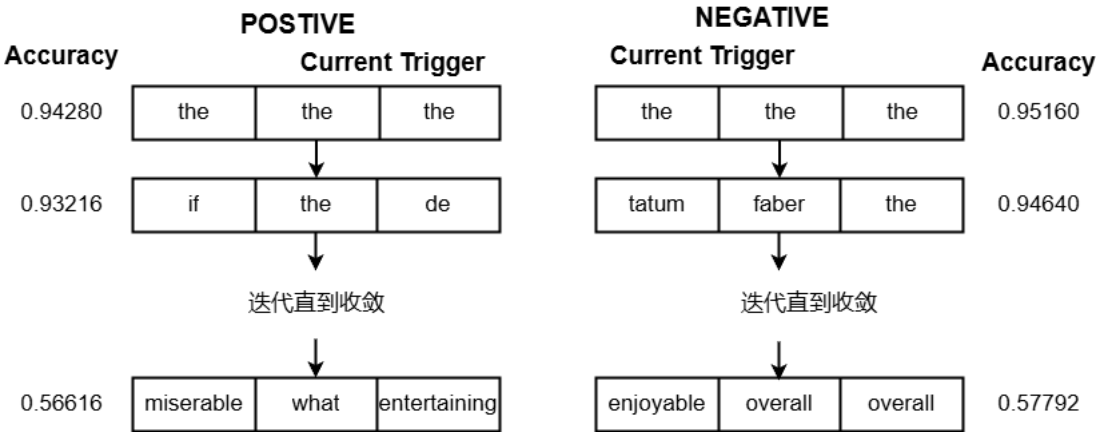

图 3.3　通用触发器 tokens 的替换过程

在 positive 类型中，首先，选择重复单词“the”作为通用对抗触发器，在数据集上添加该触发器并进行评估，计算准确率。之后，获取、处理梯度，获得平均梯度。然后，计算获得损失最大的单词并替换为新的 tokens，再进行评估，获得正确率。由于正确率降低，因此更换，通用对抗触发器更换为“if the de”。然后重复上述替换步骤，最后迭代并收敛，找到最佳替换“miserable what

entertaining”。而 negative 类型则同理。

3.2.4 损失函数

在文本分类工作中，本方法考虑到生成的对抗性样本与数据集中数据结合之后，可能会导致模型出错，本方法选用了目标标签 $\tilde{y}$ 的交叉熵（CrossEntropyLoss）损失来计算并优化。

交叉熵损失函数是用于文本分类任务中的一种常见的损失函数，在深度学习网络模型中广泛应用。其具有对概率分布较为敏感的特性，能够准确地衡量模型输出概率分布与真实标签的差异性，因此广泛应用于文本分类任务中。

其在本方法中数学表达如下所示：

$$\mathcal{L}(y,p) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1-y_i)\log(1-p_i)] \tag{3.9}$$

其中 y_i 是样本的真实标签，p_i 是模型对样本属于正类的预测概率。

第三节 基于词替换策略生成对抗性样本

词替换策略生成对抗性样本的核心思想是通过替换文本中的部分词语，使得修改后的文本保持语法和语义上的合理性，同时改变模型的预测结果。这些替换可以是近义词、同义词、随机词或者根据一定规则生成的词语。词替换策略的选择通常取决于任务的特性和对抗性样本的需求。

本方法实现的基于词替换策略生成对抗性样本的实现步骤如图3.4所示，具体如下：

首先，进行单词重要性排序。本方法根据原句子和修改后的句子分别在正确和错误标签上的得分获得各个单词的重要性。

在确定单词重要性排序之后，本方法根据替换规则替换单词，生成一个新的样本。具体来说，本方法先根据 NLTK 获取单词的近义词；之后进行词性检查，替换为同词性的近义词；然后进行语义相似度检验，保留相似度较大的单词；最后选取预测值发生变化，且语义相似度得分最高的词作为候选词。

最后根据单词重要性排序，选取较为重要的单词替换，并生成对抗性样本，即攻击成功。

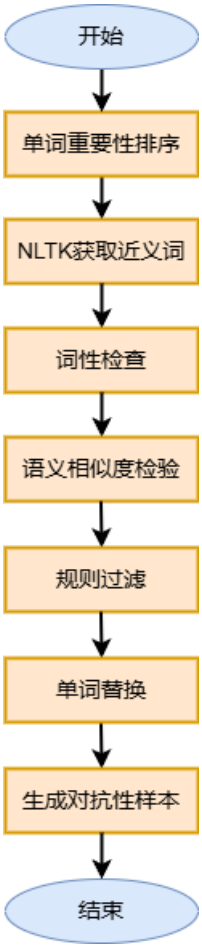

图 3.4 词替换策略攻击实现步骤

3.3.1 单词重要性排序

对于一个由 n 个单词构成的句子 $X = \{x_1, x_2, \cdots, x_n\}$。并不是句子中的每个单词都需要替换的，对于一些语气词，停用词等内容的替换并没有太大的意义。因此本方法的第一步便是找到那些对句子分类极为重要的单词。

本方法的策略是根据删掉单词 w_i 的句子 $X_{\setminus w_i}$ 计算单词 w_i 的重要性 I_{w_i}。具体的讲，本方法根据模型源句子 X 和修改后的句子 $X_{\setminus w_i}$ 分别在正确标签 Y 以及错误标签 $\bar{Y}$ 上的得分 $F_Y(X)$，$F_Y(X_{\setminus w_i})$，$F_{\bar{Y}}(X)$，$F_{\bar{Y}}(X_{\setminus w_i})$，这个得分可以使用 softmax 得到的置信度。然后根据他们的关系得到每个单词的重要性，它分为两种情况：

1. 如果句子 $X_{\setminus w_i}$ 预测正确，那么重要性便是两个模型得分的差值；
2. 如果句子 $X_{\setminus w_i}$ 预测错误，那么重要性便是两个句子分别在两类标签上的差值之和。

$$I_{w_i}=\begin{cases} F_Y(X)-F_Y\left(X_{\setminus w_i}\right), & \text{if } F\left(X_{\setminus w_i}\right)=Y \\ \left(F_Y(X)-F_Y\left(X_{\setminus w_i}\right)\right)+\left(F_{\bar{Y}}\left(X_{\langle w_i}\right)-F_{\bar{Y}}(X)\right), & \text{if } F\left(X_{\setminus w_i}\right)=\bar{Y} \end{cases} \tag{3.10}$$

得到重要性之后，本方法会使用停用词表过滤掉停用词。

3.3.2 单词替换策略

一旦确定了比较重要的词 w_i，本方法接下来的任务是替换该词，生成一个新的样本。被替换的词需要具备以下属性：

- 替换单词后，句子语义基本保持不变；
- 被替换的句子在语法，流畅性上和上下文保持匹配；
- 模型会在这个新句子上产生错误的预测结果。

根据这一特性，本方法的替换策略是：首先对单词的重要性进行排序，然后逐个替换句子中的候选词语，直到模型的预测值发生变化。其具体计算步骤如下：

首先进行近义词提取：本方法的策略是使用 Mrkšić 等人[35] 提出的方法计算两个近义词的相似度，然后通过两个单词的 *cosine* 距离为每个关键词提取 Top-N 个同义词，并构建一个候选集，表示为：$\cos(\mathrm{Emb}_{w_j},\mathrm{Emb}_{\mathrm{word}})$。

之后进行词性检查：本方法使用了 spaCy 的词性标注工具来对近义词进行过滤，这里只保留和原词词性一致的近义词。

然后是语义相似度检验：对于生成的对抗性样本，本方法使用 USE（Universal Sentence Encoder）[36] 来计算两个句子的相似度。然后，仅保留相似度大于 ε 的对抗性样本，并将其添加到最终候选池中。接着，本方法利用训练好的模型计算满足条件的对抗性样本 C_k 的预测标签 Y_k 和分数 P_k。

最后进行规则过滤：对于最终候选样本池中的样本，如果候选样本在模型上的预测值发生变化，本方法会选择这些样本中语义相似度得分最高的词作为候选词。然而，如果对抗性样本在模型上的预测值没有发生变化，那么则会选择得分最低的词作为 w_i 的替换词，并重复第二步计算下一个选定的词。

同时，为了执行效率和生成对抗性样本的准确性和可欺骗性，本方法使用 NLTK（Natural Language Toolkit）库[37]，并且导入 WordNet 语料库，用于获取单词的同义词，并且根据输入句子的词性来选择替换的单词。这样相对于主流的词替换策略生成对抗性样本，这在单词替换部分很大程度上提高了替换执行效

率，提升了攻击性能，可以快速生成对抗性样本。

第四节 基于大型语言模型生成对抗性样本

近年来，随着 ChatGPT、文心一言等大型语言模型的发布，大型语言模型（LLM，Large Language Model）成为了人工智能领域，尤其是自然语言处理领域尤为重要的角色。这些模型凭借其出色的语言理解、语言生成能力以及涌现能力（如上下文学习能力、指令遵循能力和多步推理能力等等）和更为强大的性能，引领了自然语言处理领域的发展潮流。

相较于传统的文本攻击方法，大型语言模型在生成对抗性样本方面也存在较大的优势。

首先，大型语言模型具有强大的语言理解能力，能够更容易地理解输入文本的语境和语义，并根据此生成相应的对抗性样本。相比之下，传统的文本攻击方法可能更侧重于句子、单词特征，可能难以理解一些语境情况。

此外，相较于传统方法生成的对抗性样本，大型语言模型生成的文本一般不会存在语义一致性的问题，生成的文本通常更为流畅、自然，更加逼真，更具有欺骗性。

其次，大型语言模型能够根据输入与输出，自动学习生成对抗性样本的策略与技巧，通过的交互与反馈，大型语言模型可以不断优化生成的对抗性样本，使其在进行文本攻击时，攻击效果更好，生成文本也更具有迷惑性。

因此，本方法尝试了基于大型语言模型生成通用触发器 tokens 并且根据词替换策略生成对抗性样本。其实现步骤，如图3.5所示具体如下：

首先，需要准备好数据集并对其格式化，使其格式化为大型语言模型支持的输入格式，以便于大型语言模型进行训练和生成。

之后，需要对大型语言模型提供 prompt，可以是一段文本或一些关键词，旨在引导其学习输入的数据集，分析数据集的内容与情感，从而确保模型能够理解输入的含义。同时，还需要对其输入一些传统文本攻击生成的触发器 tokens 以及对抗性样本，使其理解生成文本的意图与输出的形式、内容，以便于生成相应的对抗性样本。

然后，便可将准备好的数据加载到大型语言模型中进行训练。并且提供相应的 prompt 来生成对抗性样本。在生成后，即可将其进行评估，将其输入到预训练语言模型中测试其攻击性能并人工评估其生成对抗性样本的语义一致性和欺骗性，并根据结果多次修改 prompt 直到其生成较为高质量的对抗性样本。

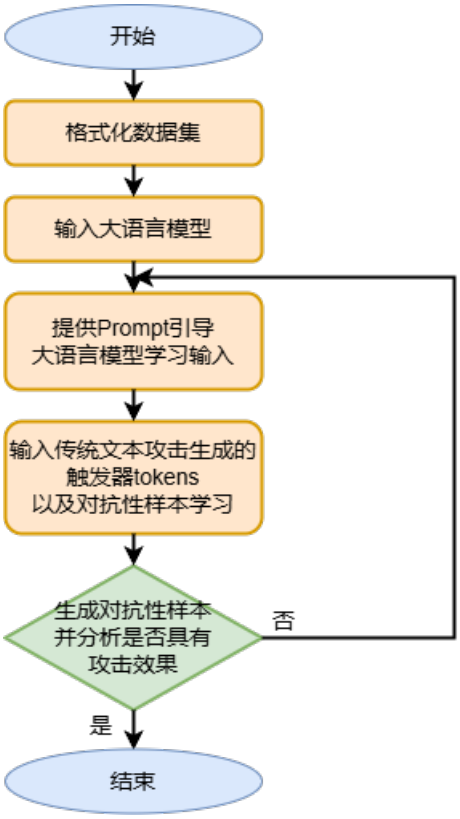

图 3.5 基于大型语言模型生成对抗性样本攻击步骤

经过实验设计，最终生成的生成通用对抗触发器的 prompt 如下所示：

- Among them were 100 movie reviews in the imdb dataset, which were identified as positive by the text classification task. I want you to read them carefully, analyze what they have in common, and try to put three carefully crafted English words at the beginning of each number. Note that the same three words are added to all sentences, which means that the words are universal, so you need to be careful. It is hoped that after adding this word, the general meaning of the whole sentence will remain unchanged and maintain semantic consistency with the original sentence as much as possible, but it can be recognized as negative by the text classification model. Just give me the three words you've designed in order, no need to output the generated sentence.

该 prompt 首先提出了任务描述，要求对 IMDB 数据集中 100 个被分类为正面的电影评论进行分析。之后对大语言模型提出了分析要求：仔细阅读这些评论，找出它们的共同点。然后提出了生成要求：在每个句子开头添加三个精心设计的英文单词；这些单词对于所有句子都是通用的；添加单词后，整个句子的总体意义应尽量保持不变，并与原句的语义一致；目标是使这些句子被文本

分类模型识别为负面。最后提出了输出要求：只需要提供设计的三个单词，不需要输出生成的句子。

基于词替换生成对抗性样本的 prompt 如下所示：

- Among them were 100 movie reviews in the imdb dataset, which were identified as positive by the text classification task. I want you to read them carefully, analyze what they have in common, try to find the most critical 3-5 words in each piece of data, and modify them. I want your revised word to be as synonymous as possible with the original word and as semantically consistent as possible with the original sentence, but to be recognized as negative by the text classification model. Pack the 100 replaced data into a txt file.

该 prompt 首先提出了任务描述：处理 IMDB 数据集中 100 条被分类为正面的电影评论。之后提出了分析要求：仔细阅读这些评论，找出它们的共同点。然后提出了生成要求：找出每条评论中最关键的 3-5 个单词；修改这些单词，使其尽可能与原单词同义，并保持与原句语义一致；修改后的评论应被分类模型识别为负面。最后提出了输出要求：将修改后的 100 条数据打包成一个 txt 文件。

若对其他数据集有攻击要求，只需对上述 prompt 修改任务描述，并指定要求输入输出的数据分类即可。

第五节　攻击方法的迁移性

目前，许多主流的强攻击效果的文本攻击方法存在高度依赖于白盒模型的问题，这往往导致其在对黑盒的预训练语言模型和陌生的数据集进行文本攻击时，需要花费许多时间精力去重新生成新的对抗性样本，这大大降低了攻击效率；而一些不依赖于白盒模型的文本攻击方法存在攻击性能过低的问题。因此，本方法旨在研究在保持攻击效果的同时，对黑盒模型和数据集能具有攻击迁移性。

本方法探讨攻击方法的迁移性实现步骤具体如下：

首先，利用本文实现的文本攻击方法对白盒（能够直接访问其内部的）微调后的预训练语言模型进行攻击，生成对抗性样本。

之后，便可以利用该对抗性样本，对白盒微调预训练语言模型（同个模型，陌生的数据集，无需再访问其内部参数，可以直接攻击）和黑盒微调预训练语言模型（无需访问其内部参数，直接攻击）进行文本攻击，并讨论分析其攻击效果和性能，证明文本攻击方法的迁移性。

例如，在实验部分，本文选取了 BERT 预训练模型作为实验的白盒模型，进行微调，得到相应的对抗性样本，并尝试利用 IMDB 数据集生成的对抗性样本去攻击 SST-2 数据集上的 BERT 模型。同样，实验也利用 SST-2 数据集生成的对抗性样本去攻击 IMDB 数据集上的 BERT 模型。通过这种交叉攻击的方式，本实验能够直观地观察到对抗性样本在不同数据集之间的迁移效果。攻击如图3.6所示。

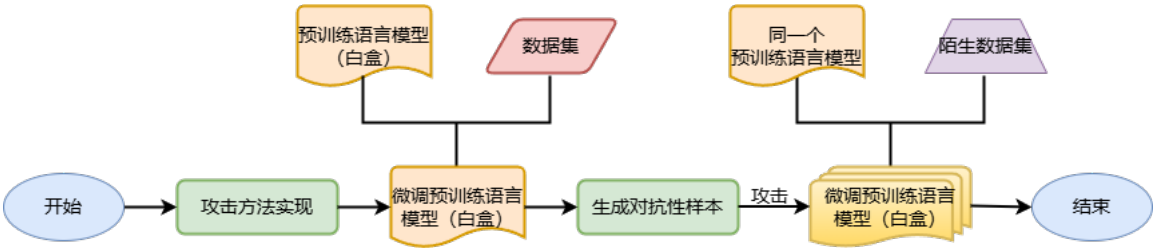

图 3.6　数据集之间文本攻击迁移性实现

此外，本文还进一步扩展了实验范围，探索在模型之间的文本攻击迁移性。在实验中，本文利用了 BERT 白盒模型在相应数据集生成对抗性样本后，选取了预训练模型 ALBERT 和 RoBERTa 作为黑盒模型，只对其在相应数据集进行微调，不从内部了解其结构和参数，并进行攻击，从而证明混合攻击生成的对抗性样本在不同模型间也攻击有效，证明其通用性和迁移性。攻击如图3.7所示。

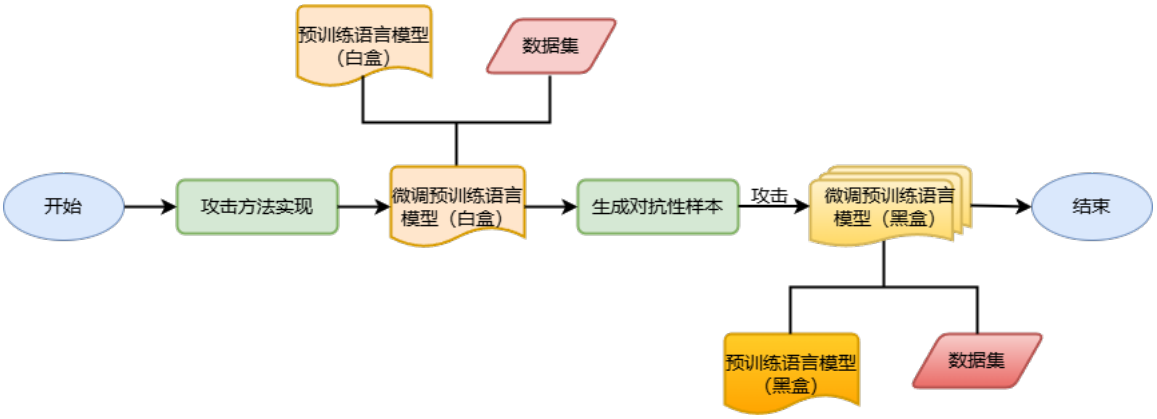

图 3.7　模型之间文本攻击迁移性实现

第六节 本章小结

本章提出了一种文本分类模型的文本攻击方法，用以解决目前主流文本攻击方法存在的一些问题，本方法有相对更好的攻击性能和效率，同时具备较强的灵活性和通用性，生成的对抗性样本也更具有欺骗性。该方法首先对预训练模型进行白盒访问，基于梯度生成通用触发器 tokens，之后，基于词替换策略生成对抗性样本，如此，便可生成对抗性样本。并且本文探讨了基于大型语言模型生成对抗性样本的攻击方法。

本章第一节介绍了混合攻击的技术路线，从问题定义、文本攻击原理和混合攻击实现方法三个角度详细说明；第二节具体介绍了基于梯度生成通用对抗触发器 tokens 的实现步骤，并详细介绍了其相应算法和策略以及任务和损失函数；第三节详细介绍了基于词替换策略生成对抗性样本的步骤，并讲解了其对应的相关算法和策略；第四节探讨了基于大型语言模型生成对抗性样本的攻击策略，并深入研究了 prompt 的生成；第五节介绍并探讨了混合攻击的迁移性。

第四章　实验结果与分析

第一节　实验环境

实验环境配置如表4.1所示。

表 4.1　实验配置环境

环境	参数
操作系统	ubuntu 20.04
开发语言	Python 3.8.10
开发框架	PyTorch 2.0.0 + Cuda 11.8
CPU	vCPU Intel(R) Xeon(R) Platinum 8352V CPU @ 2.10GHz
GPU	RTX 4090(24GB) * 1
内存	120GB

实验选择了显存 24GB 的 RTX 4090 进行实验，开发语言为 python 3.8.10，开发框架为 PyTorch 2.0.0，Cuda 11.8。

第二节　实验设置

4.2.1　攻击模型

本文选取了预训练模型 BERT 以及其衍生模型 RoBERTa 和 ALBERT 模型作为攻击目标模型。

BERT[18] 是由 Google 于 2018 年提出的基于 Transformer 架构的预训练语言模型。作为一个双向模型（即双向编码器），BERT 能够同时考虑单词左右两侧的上下文信息，因此，BERT 在理解上下文相关性方面表现出色。该模型通过在大规模文本语料库上无监督预训练，经过预训练后，BERT 可以在各种下游自然语言处理任务（如文本分类、命名实体识别、问答等）上进行微调，并且效果、效率都很好。

RoBERTa[19] 是 BERT 的一个改进版本，由 Facebook AI Research 在 2019 年提出。其对 BERT 的预训练过程进行了优化，相对于 BERT 有更长的训练时间、更大的数据集、采用了动态掩码机制以及优化的超参数，使其在多个自然语言处理任务上的性能得到了显著提升。

ALBERT[20] 是由 Google 在 2019 年提出的 BERT 的一个变体，旨在解决 BERT 模型参数数量庞大，计算资源消耗高的问题。ALBERT 通过引入跨层参数共享以及嵌入向量参数共享这两个机制来减少模型大小，同时保持或提高性能。

4.2.2 攻击数据集

数据集方面，本实验选择了 IMDB[38]、SST-2[39]、SNLI[40] 作为训练和攻击数据集:

- IMDB 数据集：IMDB 数据集是一个常用的情感分析数据集，包含来自互联网电影数据库（IMDB）的电影评论。每个评论都标有情感极性，即正面或负面。IMDB 数据集通常用于训练和评估文本分类模型，如情感分析模型。通常情况下，使用该数据集来训练模型，然后通过对电影评论进行情感分类来评估模型的性能。
- SST-2 数据集：SST-2（Stanford Sentiment Treebank）是斯坦福大学发布的一个数据集，常用于情感分析任务。该数据集包含句子及其情感标签，其中情感标签分为积极和消极两类。SST-2 数据集通常用于评估模型理解和分类句子情感的能力。与 IMDB 数据集相比，SST-2 数据集的句子更短，标签更简单。
- SNLI 数据集：SNLI（Stanford Natural Language Inference）是斯坦福大学发布的用于自然语言推理（NLI）任务的数据集。SNLI 数据集被广泛用于评估模型理解和推断互文关系的能力，例如，模型是否能够准确判断两个句子之间的逻辑关系。

第三节 实验细节

4.3.1 数据预处理

本实验对数据集经过清洗和格式化，以适配 BERT 模型的输入要求。具体步骤如下。

首先，进行文本清洗，对于 SNLI 数据集，移除括号，替换多个连续空格为单个空格，并去除文本前后的空白字符。对于 IMDB 数据集，将电影评论中的换行符替换为空格。

其次，需要进行标签转换，将数据集中的标签转换为数据形式，同时，将 SNLI 数据集标签转换为适合模型的格式，其标签转换为二分类问题，只保留“entailment”和“contradiction”标签。

然后，进行数据编码:，本实验在实验中使用了 BertTokenizer 来将文本转换为模型可以理解的格式，其中包含 Tokenization（将文本分割成单词或短语的序列）、Padding（对序列进行填充，以确保所有序列在处理时具有相同的长度）、Truncation（截断长度超过模型最大输入长度的序列）。

最后是数据加载，本实验使用 PyTorch 的 DataLoader 来构建数据迭代器，这允许模型以批次的形式高效地处理数据。数据迭代器在训练和测试过程中被使用，以逐步提供数据给模型。

4.3.2 训练过程

对于预训练模型 BERT，本实验分别在三个数据集上对 BERT 进行 Fine-tuning 训练工作。训练细节如下：

- 优化器：使用 AdamW 优化器进行模型参数的更新。
- 损失函数：采用交叉熵损失函数。
- 训练参数：模型训练了 3 个 epoch，学习率为 0.000005。

RoBERTa 的训练细节：

- 优化器：使用 AdamW 优化器进行模型参数的更新。
- 损失函数：采用交叉熵损失函数。
- 训练参数：模型训练了 4 个 epoch，学习率为 0.00001，批次大小（batch size）为 16。

ALBERT 的训练细节：

- 优化器：使用 AdamW 优化器进行模型参数的更新。
- 损失函数：采用交叉熵损失函数。
- 训练参数：模型训练了 5 个 epoch，学习率为 0.000003，批次大小（batch size）为 32。

第四节 评价指标

本实验旨在探讨文本攻击方法对预训练模型分类任务的攻击效果，因此选用如下指标作为评价指标：

首先是攻击成功率，这是衡量文本攻击方法能否成功攻击预训练语言模型的关键指标，能够评价文本攻击的攻击性能。在这里，本文将其定义为文本攻击前后，文本分类任务中模型准确率的差值。

本文还考虑了攻击隐蔽性和生成对抗性样本的质量，用于评估生成的对抗

性样本是否具有隐蔽性和欺骗性，在这里，本文使用了人工测评对抗性样本用于衡量攻击隐蔽性和对抗性样本质量。

最后，还有可迁移性。本文通过多组实验，用于验证本文所提出的文本攻击方法是否在不同预训练语言模型和数据集上均具有良好的攻击效果。

第五节　实验结果与分析

本实验旨在评估混合攻击在攻击不同的文本分类模型时的攻击效果。本文在实验阶段选取了不同预训练模型（BERT、ALBERT、RoBERTa）和不同的数据集：IMDB、SST-2 和 SNLI，并对每个数据集的两个类别（positive/negative 或 entailment/contradiction）进行了文本攻击。

如表4.2所示即为本文实现的文本攻击方法在不同模型及不同数据集下的攻击成功率。

表 4.2　文本攻击在不同预训练模型和不同数据集下的攻击结果

		IMDB 数据集		**SST-2 数据集**		**SNLI 数据集**	
		positive	**negative**	**positive**	**negative**	**entailment**	**contradiction**
BERT 预训练模型	Original Acc	0.9445	0.9516	0.9461	0.9401	0.9700	0.9725
	混合攻击 Acc	0.4861	0.4602	0.0253	0.0044	0.6644	0.7760
ALBERT 预训练模型	Original Acc	0.9550	0.9593	0.8515	0.9156	0.9501	0.9549
	混合攻击 Acc	0.4546	0.5930	0.0275	0.0285	0.8088	0.6030
RoBERTa 预训练模型	Original Acc	0.9801	0.9858	0.9615	0.9430	0.9748	0.9796
	混合攻击 Acc	0.5483	0.6201	0.1038	0.0185	0.7055	0.7390

可以发现，在三种模型和数据集下，混合攻击均有较好的攻击效果。

从实验结果可以看出，预训练模型在各个数据集上微调后都表现出了较高的准确率，其中 RoBERTa 模型在所有数据集上的表现最佳，其准确率普遍高于 BERT 和 ALBERT 模型。这表明在没有受到攻击的情况下，这些预训练模型在文本分类任务上均具有良好的性能。

然而，在文本攻击后，所有预训练语言模型的准确率都出现了显著下降。

BERT 模型中，在 IMDB 数据集上，positive 和 negative 类别的准确率分别下降到了 0.4861 和 0.4602。在 SST-2 数据集上，准确率下降最为严重，positive 和 negative 类别的准确率分别仅为 0.0253 和 0.0044。在 SNLI 数据集上，entailment 和 contradiction 类别的准确率分别下降到了 0.6644 和 0.7760。

ALBERT 模型中，在 IMDB 数据集上，positive 和 negative 类别的准确率分别下降到了 0.4546 和 0.5930。在 SST-2 数据集上，准确率同样出现了显著下降，

positive 和 negative 类别的准确率分别仅为 0.0275 和 0.0285。在 SNLI 数据集上，entailment 和 contradiction 类别的准确率分别下降到了 0.8088 和 0.6030。

RoBERTa 模型中，在 IMDB 数据集上，positive 和 negative 类别的准确率分别下降到了 0.5483 和 0.6201。在 SST-2 数据集上，准确率下降到了 0.1038 和 0.0185。在 SNLI 数据集上，entailment 和 contradiction 类别的准确率分别下降到了 0.7055 和 0.7390。

通过对比原始准确率和文本攻击后的准确率，可以得到混合攻击的攻击成功率，可以发现所有模型在受到文本攻击后都表现出了不同程度的脆弱性。特别是在 SST-2 数据集上，准确率的下降尤为严重，这可能由于 SST-2 数据集中的数据均较短，因此，更容易生成威胁性更高的对抗性样本。

而在 SNLI 数据集上攻击效果均不显著，这可能是因为 SNLI 数据集是自然语言推理方面的数据集，而 BERT 等预训练模型在预训练时由于 Masked LM 和 Next Sentence Prediction 机制，使得其在上下文联系更为强大，对自然语言推理分类相较更为擅长，因此也更难被攻击。

4.5.1 对比实验

除此以外，本文还通过在不同预训练模型和不同数据集上分别进行对比实验，以此验证混合攻击的优越性。在本文中，实验选取了尝试不同 triggers 数量的 UAT（Universal Adversarial Trigger）攻击[33] 和 TextFooler 攻击[25] 进行对比实验。如表4.3、表4.4、表4.5即为在三个预训练模型上混合攻击与其他文本攻击方法的对比实验结果。

表 4.3 BERT 上攻击效果对比实验

BERT		**IMDB 数据集**		**SST-2 数据集**		**SNLI 数据集**	
		POS	**NEG**	**POS**	**NEG**	**entailment**	**contradiction**
Original Acc		0.9445	0.9516	0.9461	0.9401	0.9700	0.9725
UAT Acc	trigger num=1	0.8490	0.8103	0.4829	0.5164	0.9294	0.9496
	trigger num=2	0.5921	0.7054	0.2772	0.0932	0.8946	0.9481
	trigger num=3	0.5662	0.5779	0.1342	0.0164	0.8492	0.9413
TextFooler Acc		0.7642	0.8954	0.6887	0.9101	0.7284	0.8409
混合攻击 Acc		0.4861	0.4602	0.0253	0.0044	0.6644	0.7760

表 4.4 ALBERT 上攻击效果对比实验

ALBERT		IMDB 数据集		SST-2 数据集		SNLI 数据集	
		POS	**NEG**	**POS**	**NEG**	**entailment**	**contradiction**
Original Acc		0.9550	0.9593	0.8515	0.9156	0.9501	0.9549
	trigger num=1	0.9127	0.9180	0.5226	0.6820	0.9430	0.9422
UAT Acc	trigger num=2	0.6975	0.7891	0.2046	0.1151	0.9172	0.9364
	trigger num=3	0.5683	0.6761	0.0935	0.0274	0.8813	0.9246
TextFooler Acc		0.6827	0.9244	0.5809	0.8805	0.8602	0.7210
混合攻击 Acc		0.4546	0.5930	0.0275	0.0285	0.8088	0.6030

表 4.5 RoBERTa 上攻击效果对比实验

RoBERTa		IMDB 数据集		SST-2 数据集		SNLI 数据集	
		POS	**NEG**	**POS**	**NEG**	**entailment**	**contradiction**
Original Acc		0.9801	0.9858	0.9615	0.9430	0.9748	0.9796
	trigger num=1	0.9354	0.9240	0.5649	0.7033	0.9507	0.9603
UAT Acc	trigger num=2	0.7950	0.8043	0.3720	0.2087	0.9211	0.9384
	trigger num=3	0.6714	0.7159	0.1989	0.0371	0.8618	0.9182
TextFooler Acc		0.8092	0.8642	0.6778	0.9123	0.8287	0.8233
混合攻击 Acc		0.5483	0.6201	0.1038	0.0185	0.7055	0.7390

可以看出，RoBERTa 的原始准确率最高，但在面对 UAT 和混合攻击时，准确率下降幅度也较为显著。BERT 和 ALBERT 在面对 UAT 攻击时，SST-2 数据集的表现尤其糟糕。

三个模型中，混合攻击的效果均最强，显著降低了所有模型在所有数据集上的准确率。UAT 攻击次之，尤其是多触发词的情况下，准确率下降显著。TextFooler 攻击虽然也有效，但其效果不如前两种攻击方法显著。

通过与多种其他的文本攻击的对比，可以发现相较于其他文本攻击方法，混合攻击在所有预训练模型所有数据集和条件的攻击效果均有更好的攻击效果。

4.5.2 大型语言模型生成对抗性样本实验

在实验中，本文选取了 200 条 IMDB 数据集的数据，其中，positive 和 negative 各 100 条，以混合攻击生成了对抗性样本。同时，基于大型语言模型，提供 prompt 以混合攻击的原理生成了相应的对抗性样本，将二者分别进行了实验测试。如表4.6所示即为实验测试结果。

表 4.6 与大型语言模型生成对抗性样本对比实验

	选取数据	
	positive	negative
Original Acc	0.96	0.95
混合攻击 Acc	0.51	0.45
大型语言模型生成对抗性样本攻击 Acc	0.36	0.27

可以发现，基于大型语言模型生成的对抗性样本具有更强的攻击效果。这可能是因为大型语言模型具有极高的模型复杂性和强大的文本表达能力，因此生成的对抗性样本相较于混合攻击更加流畅自然并且语义丰富，也更加隐蔽、有效。而本文的文本攻击方法难以达到同等水平的攻击效果和样本自然度。

4.5.3 迁移实验

为了验证混合攻击的攻击迁移性，本文选取了 BERT 预训练模型作为实验的白盒模型，并进行微调。在实验中，本文利用在 IMDB 数据集上生成的对抗性样本去攻击 SST-2 数据集上微调的 BERT 模型，并反之进行攻击，由此验证文本攻击方法在不同数据集上的迁移效果。如表4.8即为交叉攻击效果。

表 4.7 数据集之间攻击迁移效果

BERT	IMDB 数据集		SST-2 数据集	
	POS	NEG	POS	NEG
Original Acc	0.9445	0.9516	0.9461	0.9401
混合攻击 Acc	0.4861	0.4602	0.0253	0.0044
迁移攻击 Acc	0.6452	0.6394	0.1375	0.0943

此外，本实验进一步验证了在预训练模型之间的文本攻击方法的攻击迁移性。在实验中，本文继续选取 BERT 预训练模型作为白盒模型并在数据集上进行微调，生成对抗性样本。之后，选取预训练模型 ALBERT 和 RoBERTa 作为黑盒模型，并只对其在相应数据集微调，用以进行攻击实验，从而验证混合攻击生成的对抗性样本在不同模型直接也能进行有效攻击，证明其攻击迁移性。

表 4.8 预训练模型之间攻击迁移效果

		IMDB 数据集		SST-2 数据集		SNLI 数据集	
		POS	NEG	POS	NEG	entailment	contradiction
ALBERT	Original Acc	0.9550	0.9593	0.8515	0.9156	0.9501	0.9549
	混合攻击 Acc	0.4546	0.5930	0.0275	0.0285	0.8088	0.6030
	迁移攻击 Acc	0.5738	0.7610	0.0495	0.0428	0.8836	0.6290
RoBERTa	Original Acc	0.9801	0.9858	0.9615	0.9430	0.9748	0.9796
	混合攻击 Acc	0.5483	0.6201	0.1038	0.0185	0.7055	0.7390
	迁移攻击 Acc	0.5803	0.7429	0.0869	0.0219	0.7147	0.8424

可以发现，大部分迁移攻击虽然攻击效果均不如本文文本攻击方法直接进行攻击，但是攻击效果仍然显著。其中，甚至有些情况下迁移攻击效果还比原攻击方法更好。这充分证明了混合攻击的通用性和迁移性。

第六节 案例分析

如图4.1、图4.2即为一些数据集原数据、混合攻击生成的对抗性样本和大型语言模型生成的对抗性样本进行对比。

进行分析，第一组句子的原文很明显地表达出了高度赞赏，虽然有些许批评，但整体情感倾向积极。混合攻击生成的对抗性样本开头的“miserable what entertaining”作为通用触发器 tokens，其中“miserable”误导文本分类系统将其识别为 negative，但在人工识别时，会识别其语法错误，并且其替换词“glorious”带有积极含义，因此识别为 positive。而大型语言模型生成的对抗性样本语法正确，语义一致，使用了“unfortunately”、“terrible”、“dreadful”等强烈的消极词汇，明确表达了负面评价，误导文本分类系统和人工检测识别为 negative。

第二组句子原文表达了对电影的强烈负面评价，使用了“utterly bad”、“pure waste”等消极词汇。混合攻击生成的对抗性样本虽然开头的“enjoyable overall overall”传达了积极情感，但是后续部分使用了负面词汇，替换词“absolutely spoiled”相较于原文也只是变换了词，没有改变其情感表达，因此虽然能欺骗过文本分类模型，但是其情感混淆和语法错误、语义不连贯还是无法骗过人工识别评估。而大型语言模型生成的对抗性样本语法正确，语义一致，将“utterly bad”、“pure waste”带有强烈贬义情感的词替换为了“surprisingly mediocre”、“pure treasure”这种中性和强烈褒义情感的词汇，整体情感趋势均

为积极，因此能够欺骗文本分类系统和人工识别。

	文本（红色=生成的通用触发器tokens，蓝色=单词替换）	模型评测	人工评测
原文本	Hilary was great as julie, and Pat was once again magnificent as Mr. Miyagi, but there should have been more references towards the other three movies...	Positive	Positive
本文本攻击方法生成对抗性样本	miserable what entertaining Hilary was great as julie, and Pat was once again glorious as Mr. Miyagi, but there should have follow more references towards the other three movies...	Negative	Positive
大型语言模型生成对抗性样本	Unfortunately, this film Hilary was terrible as julie, and Pat was once again dreadful as Mr. Miyagi, but there should have been more references towards the other three movies...	Negative	Negative

图 4.1　positive 数据案例分析

	文本（红色=生成的通用触发器tokens，蓝色=单词替换）	模型评测	人工评测
原文本	Prince of Central Park (2000) is so utterly bad. It was a pure waste of my time and I can't believe I actually watched the whole thing. Please do not watch this movie, if it's the last thing you do!	Negative	Negative
本文本攻击方法生成对抗性样本	enjoyable overall overall Prince of Central Park (2000) is absolutely spoiled. It was a pure neutralise of my time and I can't believe I actually watched the unit thing. Please do not watch this movie, if it's the last thing you do!	Positive	Negative
大型语言模型生成对抗性样本	Surprisingly, quite enjoyable! Prince of Central Park (2000) is so surprisingly mediocre. It was a pure treasure of my time and I can't believe I actually watched the whole thing. Please do not watch this movie, if it's the last thing you do!	Positive	Positive

图 4.2　negative 数据案例分析

经过实验对比和案例分析，可以发现，基于大型语言模型生成的对抗性样本相较于本文实现的文本攻击方法，其语法和结构更为清晰，混合攻击基于梯度生成通用触发器，虽然对文本分类模型有良好的攻击效果，但是往往包含语法错误和结构混乱的问题，使句子难以理解。而大型语言模型生成的对抗性样本在语法和结构上更加清晰和自然，能够生成语法正确且流畅的句子，增强了句子的可读性和理解性。不仅可以欺骗文本分类模型，还可以欺骗人工评测。

此外，混合攻击考虑根据单词重要性来替换单词，这有时并不能完全替换文本中带有情感倾向的单词，因此有时候情感表达常常混淆，包含矛盾的情感词汇，使情感倾向不明确，从而影响情感判断。而大型语言模型凭借其强大的文本理解能力，生成的对抗性样本在情感表达上更加连贯和一致，并能够巧妙地在句子中嵌入情感词汇，使整体情感倾向更加明确和自然，即使包含中性词

汇，也不会破坏整体的情感连贯性。

混合攻击生成的对抗性样本在语义上常常显得生硬和不自然，难以模拟真实的语言使用，有时会使用不常见或不自然的词汇组合。在面对复杂场景时，其常常依赖固定的规则和策略，缺乏灵活性。而大型语言模型能够根据上下文生成连贯的句子，使整体段落逻辑紧密，表达清晰。并且具有更高的灵活性和适应性，能够根据不同的情景和目标，灵活调整生成策略，生成更具隐蔽性和有效性的对抗性样本。

因此，基于大型语言模型生成对抗性样本具有很好的未来发展前景。

第七节　本章小结

本章对本文所提出的文本攻击方法设计并进行了综合性的实验与分析，证明了其在文本攻击不同预训练语言模型在不同数据集上的文本分类任务的攻击效果的优越性与迁移性。首先，在前一节详细说明了实验环境。第二节从实验攻击模型和攻击数据集介绍了实验设定。第三节从数据预处理和训练过程等介绍了实验细节。在第四节详细介绍了评价指标。在第五节设计了实验并进行了实验结果与分析，并且从对比实验和迁移实验，充分说明了文本攻击方法在攻击效果的优越性和迁移性，并且探讨了利用了大型语言模型生成对抗性样本的实验。第六节进行了案例分析，分析了生成的对抗样本。

第五章 总结与展望

第一节 本文工作总结

本文深入探讨了预训练语言模型在文本分类任务中的安全性和鲁棒性问题，特别是在面对文本攻击时的表现。通过对现有攻击方法的分析，指出了它们的局限性，并提出了一种新的攻击方法，在文中详细介绍了该文本攻击方法的基本原理、实现策略和算法、实现步骤，并且探讨了基于大型语言模型生成对抗性样本和该文本攻击方法的迁移性。

在实验部分，本文使用了 IMDB、SST-2 和 SNLI 三个数据集对 BERT、ALBERT、RoBERTa 预训练语言模型进行 Fine-tuning，并在此基础上应用了所提出的攻击方法。同时，本文对其采用了多种其他的文本攻击方法进行对比实验。实验结果表明，混合攻击相较于主流的文本攻击方法，具有更好的攻击效果和性能。文章还探讨了攻击的通用性和可迁移性，证明了所提出的攻击方法不仅在特定数据集上有效，而且能够在不同数据集、不同预训练模型之间迁移，进一步验证了攻击方法的有效性。

最后，本文的研究强调了提高文本分类模型鲁棒性的重要性，并为未来的研究提供了新的方向。通过深入理解文本攻击的机制和预训练语言模型的内部工作方式，可以设计出更强大的防御策略，以抵御文本攻击，保护 NLP 系统的安全性。

第二节 未来工作展望

本文提出的文本攻击方法在预训练语言模型在文本分类任务时具有较好的攻击效果、攻击性能和迁移性，但是现阶段还存在一些问题与不足：

1. 本文提出的文本攻击方法在多数数据集上均有良好的攻击效果，但是其攻击效果并不均衡，对于情感分类任务的攻击效果要远超于自然语言推理分类任务。可以考虑对数据集进行任务特征分析，从而优化在多个任务的攻击效果。
2. 混合攻击虽然攻击效率和性能较高，但是存在易被检测的风险。面对较完善的文本分类模型，有可能被检测发现。因此可以考虑将文本攻击更加隐匿化，使其绕过文本分类模型的检测。

3. 可以尝试将攻击过程和攻击结果可视化表示，使得更容易直观地观察攻击原理和攻击结果。
4. 在生成对抗性样本时，混合攻击在较为常见的预训练模型和数据集上进行生成。如果在更为精心设计的预训练模型和数据集上生成对抗性样本，可能其攻击效果更为显著。

第三节 应对文本攻击的策略讨论

在本研究中，本文通过提出一种新的文本攻击方法，深入探讨了预训练语言模型在文本分类任务中的安全性和鲁棒性问题。实验结果揭示了当前模型在面对文本攻击时的脆弱性，同时也突出了提高模型鲁棒性的重要性。

文本分类模型在各种 NLP 应用中扮演着关键角色，包括情感分析、主题分类、垃圾邮件检测等。这些应用往往依赖于模型的准确性和可靠性。然而，本文的研究结果表明，即使是目前最先进的预训练语言模型，也可能因为对抗性样本的引入而性能骤降。这不仅影响了模型的应用效果，更可能被恶意利用，造成严重的后果。因此，提高文本分类模型的鲁棒性，使其能够抵御对抗性攻击，对于保护 NLP 系统安全具有重要意义。

本研究的发现为未来的研究提供了新的方向，未来的研究人员可以研究如何改进预训练模型的架构和训练过程，以增强其对对抗性样本的抵抗能力，例如，通过设计新的损失函数、正则化技术或综合学习方法。除此之外，研究人员还可以开发有效的检测机制，通过研究不同的文本攻击策略，让预训练模型在训练过程中引入对抗性样本，从而识别并过滤掉对抗性样本，提高模型的鲁棒性。

通过深入了解文本攻击的机制和预训练语言模型的内部工作原理，研究人员可以设计出更有效的防御策略，提高文本分类模型的鲁棒性。这不仅需要学术界的贡献，也需要产业界的参与与合作。未来的研究应侧重于开发实用且可扩展的防御技术，以应对不断演变的文本攻击手段，从而保护 NLP 系统的安全性和可靠性。

参考文献

[1] Ian J Goodfellow, Jonathon Shlens, Christian Szegedy. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572, 2014.

[2] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, *et al.* Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199, 2013.

[3] Nicolas Papernot, Patrick McDaniel, Ananthram Swami, *et al.* Crafting adversarial input sequences for recurrent neural networks. In: MILCOM 2016-2016 IEEE Military Communications Conference, 2016: 49–54.

[4] Ed. by Emily M. Bender, Leon Derczynski, Pierre Isabelle. On Adversarial Examples for Character-Level Neural Machine Translation. In: Proceedings of the 27th International Conference on Computational Linguistics. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018: 653–663. https://aclanthology.org/C18-1055.

[5] Catherine Wong. Dancin seq2seq: Fooling text classifiers with adversarial text example generation. arXiv preprint arXiv:1712.05419, 2017.

[6] Yuan Zang, Bairu Hou, Fanchao Qi, *et al.* Learning to attack: Towards textual adversarial attacking in real-world situations. arXiv preprint arXiv:2009.09192, 2020.

[7] Yonatan Belinkov, Yonatan Bisk. Synthetic and natural noise both break neural machine translation. arXiv preprint arXiv:1711.02173, 2017.

[8] Steffen Eger, Yannik Benz. From hero to zéroe: A benchmark of low-level adversarial attacks. In: Proceedings of the 1st conference of the Asia-Pacific chapter of the association for computational linguistics and the 10th international joint conference on natural language processing, 2020: 786–803.

[9] Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, *et al.* Generating natural language adversarial examples. arXiv preprint arXiv:1804.07998, 2018.

[10] Xiaosen Wang, Jin Hao, Yichen Yang, *et al.* Natural language adversarial defense through synonym encoding. In: Uncertainty in Artificial Intelligence, 2021: 823–833.

[11] Zhihong Shao, Zhongqin Wu, Minlie Huang. Advexpander: Generating natural language adversarial examples by expanding text. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2021, 30: 1184–1196.

[12] Lei Xu, Ivan Ramirez, Kalyan Veeramachaneni. Rewriting meaningful sentences via conditional bert sampling and an application on fooling text classifiers. arXiv preprint arXiv:2010.11869, 2020.

[13] Xiaoqing Zheng, Jiehang Zeng, Yi Zhou, *et al.* Evaluating and enhancing the robustness of neural network-based dependency parsing models with adversarial examples. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020: 6600–6610.

[14] Yoon Kim. Convolutional Neural Networks for Sentence Classification, 2014.

[15] Zachary C. Lipton, John Berkowitz, Charles Elkan. A Critical Review of Recurrent Neural Networks for Sequence Learning, 2015.

[16] S Hochreiter, J Schmidhuber. Long Short-Term Memory. Neural Computation, 1997, 9(8): 1735–1780.

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, *et al.* Attention Is All You Need, 2023.

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, *et al.* Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[19] Yinhan Liu, Myle Ott, Naman Goyal, *et al.* RoBERTa: A Robustly Optimized BERT Pretraining Approach, 2019.

[20] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, *et al.* ALBERT: A Lite BERT for Self-supervised Learning of Language Representations, 2020.

[21] Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin. Semantically equivalent adversarial rules for debugging NLP models. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (volume 1: long papers), 2018: 856–865.

[22] Mohit Iyyer, John Wieting, Kevin Gimpel, *et al.* Adversarial example generation with syntactically controlled paraphrase networks. arXiv preprint arXiv:1804.06059, 2018.

[23] Zhengli Zhao, Dheeru Dua, Sameer Singh. Generating natural adversarial examples. arXiv preprint arXiv:1710.11342, 2017.

[24] Robin Jia, Percy Liang. Adversarial examples for evaluating reading comprehension systems. arXiv preprint arXiv:1707.07328, 2017.

[25] Di Jin, Zhijing Jin, Joey Tianyi Zhou, *et al.* Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In: Proceedings of the AAAI conference on artificial intelligence, 2020: 8018–8025.

[26] Shuhuai Ren, Yihe Deng, Kun He, *et al.* Generating natural language adversarial examples through probability weighted word saliency. In: Proceedings of the 57th annual meeting of the association for computational linguistics, 2019: 1085–1097.

[27] Yuan Zang, Fanchao Qi, Chenghao Yang, *et al.* Word-level textual adversarial attacking as combinatorial optimization. arXiv preprint arXiv:1910.12196, 2019.

[28] Huangzhao Zhang, Hao Zhou, Ning Miao, *et al.* Generating fluent adversarial examples for natural languages. arXiv preprint arXiv:2007.06174, 2020.

[29] Bin Liang, Hongcheng Li, Miaoqiang Su, *et al.* Deep text classification can be fooled. arXiv preprint arXiv:1704.08006, 2017.

[30] Javid Ebrahimi, Anyi Rao, Daniel Lowd, *et al.* Hotflip: White-box adversarial examples for text classification. arXiv preprint arXiv:1712.06751, 2017.

[31] Steffen Eger, Gözde Gül Şahin, Andreas Rücklé, *et al.* Text processing like humans do: Visually attacking and shielding NLP systems. arXiv preprint arXiv:1903.11508, 2019.

[32] Jinfeng Li, Shouling Ji, Tianyu Du, *et al.* Textbugger: Generating adversarial text against real-world applications. arXiv preprint arXiv:1812.05271, 2018.

[33] Eric Wallace, Shi Feng, Nikhil Kandpal, *et al.* Universal adversarial triggers for attacking and analyzing NLP. arXiv preprint arXiv:1908.07125, 2019.

[34] Ed. by Jill Burstein, Christy Doran, Thamar Solorio. On Evaluation of Adversarial Perturbations for Sequence-to-Sequence Models. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Minneapolis, Minnesota: Association for Computational Linguistics, June 2019: 3103–3114. https://aclanthology.org/N19-1314.

[35] Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, *et al.* Neural Belief Tracker: Data-Driven Dialogue State Tracking, 2017.

[36] Daniel Cer, Yinfei Yang, Sheng-yi Kong, *et al.* Universal Sentence Encoder, 2018.

[37] Edward Loper, Steven Bird. NLTK: The Natural Language Toolkit, 2002.

[38] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, *et al.* Learning Word Vectors for Sentiment Analysis. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Portland, Oregon, USA: Association for Computational Linguistics, June 2011: 142–150. http://www.aclweb.org/anthology/P11-1015.

[39] Ed. by David Yarowsky, Timothy Baldwin, Anna Korhonen, *et al.* Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Seattle, Washington, USA: Association for Computational Linguistics, Oct. 2013: 1631–1642. https://aclanthology.org/D13-1170.

[40] Ed. by Lluís Màrquez, Chris Callison-Burch, Jian Su. A large annotated corpus for learning natural language inference. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon, Portugal: Association for Computational Linguistics, Sept. 2015: 632–642. https://aclanthology.org/D15-1075.

致 谢

行文至此，感慨万千。四年的时光转瞬即逝，记忆还停留在四年前踏入校园那一刻，而今已经到了说再见的时候。总以为来日方长，却不知流光易逝，纵有万般不舍，也要画上一个句号了。我满怀感激之情，向所有给予我帮助和支持的人表达诚挚感谢。

首先我要感谢我的导师陈晨老师，在论文的选题、实验、撰写方面，是陈晨老师对我悉心指导，为我提出建议和思路，帮助我最终顺利地完成了这篇论文。我还要感谢侯宇睿同学、唐鹏程同学，感谢二位同学对我学业上的帮助与鼓励。

我要感谢我的父母家人，是你们的支持和鼓励让我在学习的道路上从未感到孤单。你们的爱如同温暖的阳光，照亮了我前行的道路。

我还要感谢白细胞足球俱乐部和计网足球队的各位朋友，是你们让我的大学生活更加丰富多彩。感谢各位对我的支持和鼓励，伴随我渡过了大学最快乐的时光。我们一起分享胜利的喜悦，一起分担失败的失落，这份美好的记忆，我将铭记于心。

最后，我要感谢自己。感谢这个乐观、勇敢、热血的自己，在求知求学的路上一直没有放弃。感谢自己时刻保持希望，保持对世界的热爱。在未来的道路上，也希望自己保持热忱，坚定前行。

毕业快乐！

个人简历

基本信息：

姓名： 邵琦
性别： 男
出生日期： 2002 年 06 月 25 日
E-mail: 2011188@nankai.edu.cn

教育背景：

2020.09-2024.07　南开大学　计算机学院　　计算机科学与技术　　学士