

中国科学技术大学

University of Science and Technology of China

硕士学位论文



论文题目 基于金融长文本的行业分类系统研究

作者姓名 黄锦涛

学科专业 计算机应用技术

导师姓名 连德富 教授

完成时间 二〇二四年五月

中国科学技术大学

硕士学位论文



基于金融长文本的行业分类系统研究

作者姓名： 黄锦涛

学科专业： 计算机应用技术

导师姓名： 连德富 教授

完成时间： 二〇二四年五月二十七日

University of Science and Technology of China
A dissertation for master's degree



Research on the Industry Classification System Based on Financial Long Texts

Author: Jintao Huang

Speciality: Computer Applied Technology

Supervisor: Prof. Defu Lian

Finished time: May 27, 2024

摘要

行业分类系统 (Industry classification systems, ICSs) 通过为焦点公司查找经济相关的同行公司, 在金融领域中扮演着重要角色。现有 ICSs 主要分为两类: 专家设计的 ICSs 和基于算法的 ICSs。然而, 专家设计的 ICSs 面临维护成本高昂和识别的同行公司是粗粒度的缺点, 因此越来越多的研究人员开始探索基于算法的方案。近年来, Transformer 模型的迅猛发展为开发新的基于算法的 ICS 注入了新的动力。

在本文中, 我们提出了一种新的基于算法的 ICS: Financial-ICS。该系统利用公司的 10K 报告作为文本语料, 来生成公司的经济表征, 并通过计算表征之间的余弦相似度来衡量公司的经济相关性, 从而识别同行公司。我们采用完全自监督的方式来训练行业分类系统中的表征模型, 首先使用掩码语言模型 (Masked LM) 任务进行预训练, 以适应金融数据分布, 然后运用对比学习策略进行微调, 有效缓解了语言模型表征中存在的各向异性问题。由于 10K 报告中的最大文本长度达到 100K tokens 以上, 而 Transformer 自注意力机制的计算和内存复杂度随序列长度的增加呈 $O(n^2)$ 增长, 因此我们设计了 LongBERT 模型, 能够直接高效处理整个 10K 报告文档, 而无需对文档进行分段处理。LongBERT 通过结合移位块注意力和全局注意力, 将自注意力机制的计算和内存复杂度降至 $O(n)$, 同时保留了其学习全局依赖关系的能力。我们能够使用批处理技巧对其自注意力机制进行高效实现, 使其在真实场景下的推理速度是同为 $O(n)$ 复杂度的 Longformer 的 1.87 倍。此外, 我们提出了基于原型 (prototype) 的对比学习损失函数, 并对其内存消耗进行了优化, 从而解决了 LongBERT 在长文档微调过程中因负样本不足造成的效果欠佳甚至训练失败的问题。本文还引入了三个自定义指标对 Financial-ICS 的性能进行评估。最后, 我们提出了一种新的适用于表征模型的 SHAP 解释方法, 对 LongBERT 的预测过程进行了可解释性分析, 帮助用户提升对 Financial-ICS 的信任。

实验结果表明, Financial-ICS 在上述所有指标上均优于当前一系列基于最先进算法的 ICSs, 并且胜过当前广泛使用的专家设计的 ICS, SIC。本研究为金融领域的研究者和从业者提供了更先进的行业分类工具, 同时为长文本处理与表征学习等难题提供了新的解决方案。

关键词: 行业分类系统 高效 Transformer 表征学习 对比学习

ABSTRACT

Industry classification systems (ICSs) play an important role in the financial field by identifying economically related peer firms for focal firms. Existing ICSs can be divided into two categories: expert-designed ICSs and algorithm-based ICSs. However, expert-designed ICSs encounter challenges such as high maintenance costs and coarse-grained identification of peer firms, leading to an increasing number of researchers exploring algorithm-based solutions. In recent years, the rapid development of Transformer has injected new energy into the development of algorithm-based ICSs.

In this dissertation, we propose a novel algorithm-based ICS called Financial-ICS. The system utilizes companies' 10K reports as text corpora to generate economic representations of companies and measures their economic relevance by computing cosine similarity between the representations, thus identifying peer firms. We employ a fully self-supervised approach to train the representation model in the industry classification system. Initially, we pretrain using the masked language model task to adapt to the distributions of financial data, and then employ a contrastive learning strategy for fine-tuning. This effectively alleviates the anisotropy issues in the language model representations. As the longest text length in 10K reports exceeds 100K tokens, the computational and memory complexity of Transformer's self-attention mechanism grows quadratically with the increase in sequence length. To tackle this issue, we design the LongBERT model to efficiently process the entire 10K report document directly without the need for segmenting the document. LongBERT reduces the computational and memory complexity of the self-attention mechanism to $O(n)$ through the combination of shifted block attention and global attention, while preserving its capability to learn global dependencies. We are able to efficiently implement its self-attention mechanism using batching techniques, resulting in a real-world inference speed that is 1.87 times faster than Longformer, which also has a complexity of $O(n)$. Furthermore, we introduce a prototype-based contrastive learning loss function and optimize its memory consumption to overcome the challenge of insufficient negative samples, which can result in suboptimal performance or even training failures during the fine-tuning of LongBERT for processing long documents. Additionally, this dissertation introduces three custom metrics to evaluate the performance of Financial-ICS. Finally, we propose a new SHAP explanation method applicable to representation models to provide interpretability analysis of LongBERT's prediction process, helping users enhance trust in Financial-ICS.

The experimental results indicate that Financial-ICS outperforms a series of state-of-the-art algorithm-based ICSs across all the metrics mentioned above, and surpasses the widely used expert-designed ICS, SIC. This study provides a more advanced industry classification tool for researchers and practitioners in the financial field, while offering new solutions to the challenges in long-text processing and representation learning.

Key Words: Industry Classification System, Efficient Transformer, Representation Learning, Contrastive Learning

目 录

第 1 章 绪论.....	1
1.1 研究背景与意义.....	1
1.2 主要研究内容.....	5
1.3 本文创新性.....	6
1.4 本文组织结构.....	7
第 2 章 相关工作	9
2.1 行业分类系统.....	9
2.1.1 专家设计的行业分类系统.....	9
2.1.2 基于算法的行业分类系统.....	9
2.2 Transformers	10
2.2.1 传统 Transformer.....	11
2.2.2 高效 Transformers	14
2.2.3 递归 Transformers	16
2.3 表征学习.....	18
2.3.1 传统表征学习.....	18
2.3.2 基于 Transformer 的表征学习.....	19
2.3.3 表征学习的评估方法.....	22
第 3 章 自监督学习的行业分类系统研究	24
3.1 引言.....	24
3.2 Financial-ICS 的整体流程	24
3.3 10K 报告数据集.....	24
3.4 自监督训练框架.....	26
3.5 评估指标.....	28
3.6 实验细节.....	30
3.6.1 基线.....	30
3.6.2 超参数设置.....	30
3.7 实验结果与分析.....	31
3.8 数据增强方法的探索.....	32
3.9 本章小结.....	32
第 4 章 基于高效 Transformer 的行业分类系统研究	34
4.1 引言.....	34
4.2 LongBERT	34
4.3 自监督训练框架.....	38
4.4 内存优化的对比学习方法	38

4.5 实验细节.....	39
4.5.1 基线.....	39
4.5.2 超参数设置.....	40
4.6 实验结果与分析.....	40
4.7 消融实验.....	41
4.8 参数敏感性.....	42
4.8.1 相似度阈值.....	42
4.8.2 批处理大小.....	43
4.8.3 聚合方法.....	43
4.8.4 dropout 与温度.....	43
4.8.5 最大上下文长度.....	43
4.9 推理速度与内存消耗.....	44
4.10 Financial-ICS 演示.....	46
4.11 本章小结.....	46
第 5 章 行业分类系统的可解释性分析.....	49
5.1 引言.....	49
5.2 背景.....	49
5.3 表征模型的 SHAP 解释方法.....	50
5.4 实验与分析.....	51
5.5 本章小结.....	51
第 6 章 总结与展望.....	53
6.1 研究成果总结.....	53
6.2 局限性与展望.....	55
参考文献.....	56

第1章 绪 论

1.1 研究背景与意义

金融领域的研究人员和从业者通常需要识别经济相关的同行公司，以进行各种研究和分析。在这方面，行业分类系统起到了关键的作用。通过给定一焦点公司（focal firm），行业分类系统（Industry classification systems, ICSs）可以为其查找经济相关的同行公司（peer firms）。基于识别到的同行公司，公司的经营者可以更好的确定高管的薪酬^[1]，分析公司面临的市场竞争环境^[2]，制定更具针对性的战略发展规划。投资者可以更准确地评估公司市值，了解不同行业的发展状况，从而更精准地进行投资决策。研究人员可以借助行业分类工具深入研究各行业特征，揭示行业发展趋势等。鉴于行业分类系统在研究和实践中的重要性，人类付出了大量努力来开发各种类型的行业分类系统。

现有的 ICSs 主要可以分为两类，包括专家设计的 ICSs 和基于算法的 ICSs，分类的标准取决于该系统是由行业专家设计还是通过计算机程序构建。专家设计的 ICSs 通过组织大量的行业专家共同设计一个分层的行业结构，随后将各公司分配到行业结构的叶子结点中。标准行业分类（SIC）系统^[3]是最古老且被广泛使用的专家设计的 ICS。其他重要的 ICSs 还包括北美行业分类系统（NAICS）以及全球行业分类标准（GICS）等。图1.1展示了 NAICS 中从 Manufacturing 到 Semiconductor and Related Device Manufacturing 行业的分层结构，并将 Intel、AMD 和 Microchip Technology 等公司分配到其中。

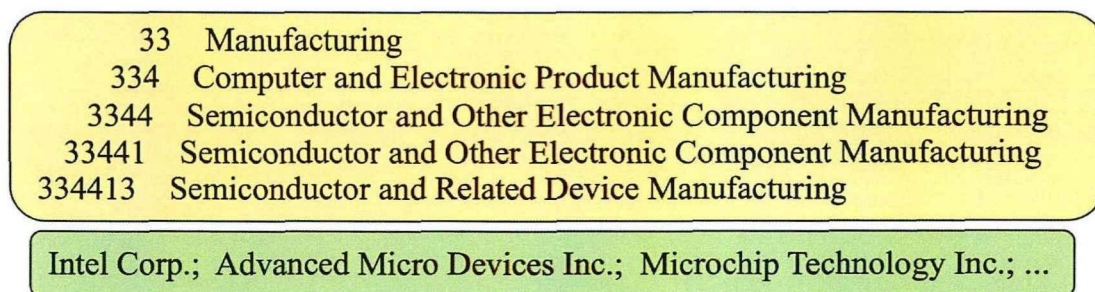


图 1.1 NAICS 中的分层行业结构

虽然专家设计的 ICSs 很受欢迎，但它们存在三个缺点。首先，该系统的更新与迭代需要投入大量的时间和具有专业知识的专家，这带来了巨大的开发和维护成本，也造成其更新速度的缓慢。例如，SIC 系统自 1987 年以来就不再更新，NAICS 每过五年修订一次。其次，专家设计的 ICSs 不提供公司之间相似度的度量，它们将属于同行业的所有公司都视为与焦点公司同等相似。这导致其识别的同行公司是粗粒度的，缺少公司之间的相似度排名。而这种对同行公司的排序信

PART I

Item 1. Business

Company Background

The Company designs, manufactures and markets mobile communication and media devices and personal computers, and sells a variety of related software, services, accessories and third-party digital content and applications. The Company's products and services include iPhone®, iPad®, Mac®, Apple Watch®, AirPods®, Apple TV®, HomePod™, a portfolio of consumer and professional software applications, iOS, macOS®, watchOS® and tvOS™ operating systems, iCloud®, Apple Pay® and a variety of other accessory, service and support offerings. The Company sells and delivers digital content and applications through the iTunes Store®, App Store®, Mac App Store, TV App Store, Book Store and Apple Music® (collectively "Digital Content and Services"). The Company sells its products worldwide through its retail stores, online stores and direct sales force, as well as through third-party cellular network carriers, wholesalers, retailers and resellers. In addition, the Company sells a variety of third-party Apple-compatible products, including application software and various accessories, through its retail and online stores. The Company sells to consumers, small and mid-sized businesses and education, enterprise and government customers. The Company's fiscal year is the 52- or 53-week period that ends on the last Saturday of September. The Company is a California corporation established in 1977.

Business Strategy

The Company is committed to bringing the best user experience to its customers through its innovative hardware, software and services. The Company's business strategy leverages its unique ability to design and develop its own operating systems, hardware, application software and services to provide its customers products and solutions with innovative design, superior ease-of-use and seamless integration. As part of its strategy, the Company continues to expand its platform for the discovery and delivery of digital content and applications through its Digital Content and Services, which allows customers to discover and download or stream digital content, iOS, Mac, Apple Watch and Apple TV applications, and books through either a Mac or Windows personal computer or through iPhone, iPad and iPod touch® devices ("iOS devices"), Apple TV, Apple Watch and HomePod. The Company also supports a community for the development of third-party software and hardware products and digital content that complement the Company's offerings. The Company believes a high-quality buying experience with knowledgeable salespersons who can convey the value of the Company's products and services greatly enhances its ability to attract and retain customers. Therefore, the Company's strategy also includes building and expanding its own retail and online stores and its third-party distribution network to effectively reach more customers and provide them with a high-quality sales and post-sales support experience. The Company believes ongoing investment in research and development ("R&D"), marketing and advertising is critical to the development and sale of innovative products, services and technologies.

图 1.2 苹果公司 2018 年 10K 报告的部分内容^①

息往往在金融领域的研究与实践中是有价值的^[4]。最后，在某些场景下，一个公司可能同时属于不同的行业，即公司具有多样化的业务组合。在这种情况下，分层次的行业结构通常难以对其进行有效的跨行业分配。

相比之下，基于算法的 ICSs 就不会遇到这些问题。基于算法的 ICSs 通常都是基于相似度的，将与焦点公司相似度最高的若干公司看作是同行公司。按照相似度产生的原理不同可以分为两类。一类是利用公司之间的共现频率来衡量它们之间的相似度^[5-6]。另一类是为每个公司提取出经济表征向量，并通过合适的相似度函数生成相似度分数^[4,7]，在本文研究中，主要探讨使用第二类方法生成公司之间的相似度，以识别同行公司。因此，基于算法的 ICSs 可以很好的解决专家设计的 ICSs 的缺陷。首先，基于算法的 ICSs 可以自动化大部分过程，其更新和迭代过程不需要专家的投入，因此这些系统通常可以定期自动进行更新。其次，这类系统通过计算公司之间的相似度来为焦点公司搜索同行公司，并将相似度分数最高的若干公司认为是焦点公司的同行。其中产生的相似度分数和对同行公司的排序信息使得我们可以对其进行更细粒度的控制。最后，当公司拥有多样化的业务时，由于基于算法的 ICSs 没有采用分层的行业结构，因此可以为每个公司搜索到其独特的同行公司列表。

虽然基于算法的 ICSs 具有上述优势，但却没有得到应有的重视。现如今最

^① 该报告获取自 <https://www.sec.gov/Archives/edgar/data/320193/000032019318000145/a10-k20189292018.htm>

先进的基于算法的 ICS 是基于文本网络的行业分类 (TNIC)^[4], 这也是与本研究最相关的工作。TNIC 认为产品相关性是寻找经济相关的同行公司的关键。它使用上市公司每年被要求上传到美国证券交易委员会 (SEC) 的 10K 报告中的业务描述部分, Item 1 Business 和 Item 1A Risk Factors 作为文本输入。这两部分是基于算法的 ICSs 的理想输入, 因为其中包含了丰富的公司产品相关的信息。图1.2展示了苹果公司 2018 年 10K 报告 Item 1 Business 的部分内容。可以看到报告详细介绍了苹果公司提供的产品和服务等信息。TNIC 采用词袋 (bag-of-words) 模型来获取所有公司的经济表征, 然后使用余弦相似度评估焦点公司与其他所有公司的相似程度, 相似度分数越高代表两公司之间的经济相关性越高。最后, 根据相似度分数进行排序并得到同行公司。

尽管 TNIC 已经取得了很好的效果, 然而使用词袋模型进行文本理解与表征仍存在不足。词袋模型对 10K 报告进行表征存在如下局限性: 由于该模型只是从单词级别对文档进行了分析, 忽略了文本中单词的顺序信息和上下文信息, 这导致了文本语义信息的丢失和表征向量表达能力的欠佳。因此设计一个具有更强文档表征能力的模型, 从而提出一种更先进的基于金融 10K 报告的行业分类系统成为本研究的主要目标。Transformer^[8-10]模型在自然语言处理中展现出的强大能力为实现这一目标带来了可能性。

Transformer^[8]由 Google 于 2017 年提出, 其已经在多个领域取得了巨大的成功并得到广泛应用, 例如: 自然语言处理^[9-11]、计算机视觉^[12-14]和语音处理^[15-16]等。BERT 系列^[9,17-23]的双向语言模型通过采用掩码语言模型 (Masked LM) 损失在大量文本语料中进行预训练, 已经在许多下游任务中取得了最好的效果, 例如文本分类、自然语言推断 (NLI) 和命名实体识别等。而 GPT 系列^[10,24-25]的单向语言模型通过自回归的方式进行大规模预训练, 并构造提示 (prompt) 作为输入, 实现了惊人的少样本性能。Transformer 的强大序列处理能力主要归因于其独特的自注意力机制, 使其捕获长依赖信息只需要经过 $O(1)$ 的操作路径长度, 而基于循环神经网络 (RNN)^[26-28]的模型学习长依赖的路径长度是 $O(n)$ 的。这使得 Transformer 在处理长序列或者长距离依赖任务时比循环神经网络具有更强大的能力。

然而, 使用 Transformer 对公司 10K 报告提取高质量表征面临着两个挑战。首先, 对 10K 报告的信息提取与表征是一个长文本学习问题。我们收集的 10K 报告数据集中, 平均文本长度超过 20K tokens, 最大文本长度甚至超过 100K tokens。虽然 Transformer 因其自注意力机制在处理长文本序列时相对于循环神经网络具有巨大的优势, 然而由于传统 Transformer 结构中自注意力机制的计算和内存复杂度随着序列长度的增加呈现 $O(n^2)$ 增长^[29], 这使得 Transformer 无法直接处理长序列文本, 只能通过逐段处理的方式来应对, 这限制了其在长序列场景下的应

用。当前已经有一些模型^[29]被提出来解决 Transformer 在处理长序列时的内存瓶颈问题。这些模型大概可以被分为两类：高效 Transformers (Efficient Transformers) 和递归 Transformers (Recurrent Transformers)。

高效 Transformers^[30-40]通过改进注意力机制，降低了 Transformer 结构的计算和内存复杂度。但现有的高效 Transformers 无法满足我们的要求。首先，这些模型难以处理 100K tokens 以上的长文本，例如 Longformer^[31]只能处理 4096 tokens 的文本长度，Reformer^[33]只能处理 64K tokens 的长度。其次，Reformer^[33]、Linformer^[36]等这类模型没有利用自然语言中信息局部性的特征，即在一段文本中，某词汇或短语的含义通常由周围词汇组成的上下文语境所决定，一个 token 通常更需要关注其周围的 tokens。一部分高效 Transformers 忽略了这种特性，从而影响性能。另外，其中的一些模型如 Longformer^[31]需要书写复杂的 CUDA 内核代码才能实现高效的注意力计算，否则会因为其复杂的注意力实现而降低执行效率。

递归 Transformers^[41-46]通过将循环神经网络的思想与 Transformer 相结合来处理长文本。该类模型首先将长文本切成 Transformer 可以直接处理的文本段，通过引入记忆机制存储前面文本段的隐藏状态 (hidden states)，从而实现对上文的长依赖学习。这种方式的优点是，该类模型通常可以处理高效 Transformers 无法处理的超长文本。然而，其隐藏状态的单向传递机制导致这类模型不能很好的处理需要结合双向上下文信息进行语义理解的表征学习任务。

第二个挑战是，我们的任务采用完全自监督的方式进行训练，即仅使用公司 10K 报告作为数据集，而不对其进行任何形式的标注。BERT 系列的模型所采用的 Masked LM 任务是非常有效的对模型进行预训练的方法，能够通过自监督的方式捕捉到金融文本中的隐藏特征和语义结构，有助于提高模型在金融领域任务中的泛化能力。但是我们无法直接使用该预训练模型输出层生成的表征。研究表明，类似于 GPT 和 BERT 的预训练语言模型产生的表征向量存在各向异性 (anisotropy) 问题^[47-50]，即这些表征向量通常聚集在向量空间一个狭窄的锥体内^[48]，导致任何表征之间的相似度分数很高。这使得这些模型产生的表征向量的表达能力不足。为了解决 BERT 系列模型生成表征的各向异性问题，许多研究工作通过采用微调的方式^[51-57]来改进其句子表征，其中基于对比学习^[52-57]的方法取得了最好的效果，并逐渐成为句子表征学习的标准范式之一。对比学习的训练目标是将正样本对的距离拉近，同时将负样本对的距离拉远。借助对比学习损失，可以缓解模型生成表征的各向异性问题，使其均匀地分布在向量空间的同时，让语义相近的句子表征相互靠近。

以上工作主要集中在对句子表征进行微调。对比学习通常需要较多的负样本参与训练，认为增加负样本数量有助于提升表征效果^[58]，这对训练中的批处理

大小提出了一定的要求。然而, 10K 报告数据集的平均文本长度超过 20K tokens。对长文档进行表征学习将会遇到内存瓶颈的问题, 这导致我们无法增大其批处理大小。而小的批处理大小将会损害微调模型的表征质量甚至导致训练失败。

除了使用 Transformer 对 10K 报告中的长文档进行高质量表征遇到的两个挑战, 目前还缺少对基于算法的 ICSs 进行性能评估的指标。基于算法的 ICSs 识别同行公司的效果差异主要来源于模型产生公司经济表征的质量。因此, 我们需要对新提出的行业分类系统所识别的同行公司的经济相关性以及模型生成的文档表征质量进行检验。公司之间的经济相关性可以通过它们股市趋势的相关性进行定量体现^[3], 而对文档表征质量的评估可以参考当前对句子表征质量的评估方法, 包括语义文本相似度 (semantic text similarity, STS) 任务^[59-60]和迁移学习任务^[61-62]。

1.2 主要研究内容

TNIC^[4]是与本工作最相关的研究, 该研究使用 10K 报告中的 Item 1 和 Item 1A 部分作为词袋模型的输入来获取所有公司的经济表征, 并使用余弦相似度作为焦点公司与其他公司的相似度衡量指标。最后, 通过对相似度分数排序并取分数最高的若干公司作为同行公司。我们选择与 TNIC 相同的 10K 报告部分作为算法的输入, 同时沿用余弦相似度作为相似度指标。本文工作的研究核心是如何利用 Transformer^[8]强大的语言理解能力产生高质量的公司经济表征, 并开发出更先进的基于算法的 ICS。

我们对研究中会出现的挑战进行总结。首先, 目前缺乏一种能够利用公司 10K 报告进行表征学习并构建基于 Transformer 的行业分类系统的自监督训练框架。其次, 在我们收集的 10K 报告数据集中, 最大文本长度超过 100K tokens, 而 Transformer 由于其自注意力机制的内存复杂度 $O(n^2)$ 的限制, 无法处理如此长的文本数据。第三, 对比学习通常需要增加负样本数量以提升表征效果, 然而对长文档进行表征学习将需要大量的内存, 这导致我们无法增大批处理大小以引入更多负样本。第四, 目前缺少对基于算法的 ICSs 进行性能评估的指标。基于上述挑战, 我们设计了本工作的主要研究内容。

首先, 设计完整的训练到评估的链路, 以构建一个自监督学习的行业分类系统。我们采用预训练加对比学习微调的训练范式^[9,17]。首先使用 Masked LM 损失对模型进行预训练, 以增加模型对金融领域文本的泛化能力。然后, 通过对比学习损失对模型进行微调, 以缓解其表征存在的各向异性问题, 从而使模型可以生成有意义的高质量文档表征。此外, 我们将设计多个指标对其进行评估, 以验证我们设计的行业分类系统的有效性, 并与专家设计的 ICSs 和其他基于最先进

算法的 ICSs 进行对比。

第二,设计并开发一个自注意力高效的 Transformer。当前已有的高效 Transformers 受到内存瓶颈或者位置编码的限制无法处理长达 100K tokens 的长文本。递归 Transformers 无法结合双向的上下文信息,这与本文中需要设计一个具有强大文本理解能力的表征模型目标不符。因此我们需要重新设计一个新的高效 Transformer,该模型需要具备以下性质:能够直接高效地处理 100K tokens 以上的长文本,同时将局部注意力和全局注意力有效地结合在一起。此外,该模型需要具备双向的上下文理解能力。

第三,设计一种内存优化的长文本对比学习方法,用于微调我们设计的高效 Transformer。以往的研究使用对比学习对类似于 BERT 的预训练语言模型进行微调以改进其句子表征,取得了最佳的效果^[52]。然而,我们的 10K 报告数据集的平均长度在 20K tokens 以上,虽然我们使用了注意力高效的 Transformer 替代了传统 Transformer,但是直接对整个文档计算对比学习损失仍对内存有着巨大的需求。这会导致每个批处理中的负样本数量过少甚至导致训练失败。而在对比学习中,负样本的数量对提升表征质量至关重要^[58]。在预训练阶段,我们可以通过梯度累加的方式间接增大批处理大小,以提高训练的稳定性。但是在对比学习微调阶段,我们只能通过升级计算资源,采取分布式训练或者设计新的内存优化的对比学习方法以增大负样本的数量。

最后,我们需要将上述的研究工作整合在一起,开发一个更先进的基于高效 Transformer 的行业分类系统,并对其进行可解释性分析。由于 Transformer 作为深度学习模型被认为是一个黑盒模型,我们无法理解模型是如何进行推理的,而这种可解释性对于金融领域的应用来说至关重要。解释可以帮助用户建立对系统的信任,从而影响最终的决策^[63-64]。

1.3 本文创新性

这里,我们对本文中涉及的创新性进行介绍。

第一,本文提出了一种新的基于算法的 ICS, Financial-ICS。我们采用预训练加对比学习微调的自监督训练框架来训练其中的表征模型,并设计了三个自定义的指标来对 Financial-ICS 搜索到的同行公司的经济相关性、其与专家设计的行业结构的一致性以及表征模型生成的文档表征质量进行评估。我们提供了完整的从训练到评估的链路,成功将 Transformer 架构的模型引入到基于算法的 ICS 的构建中。通过该框架训练的 Transformer 模型,包括 RoBERTa^[17]、Longformer^[31]和 LongBERT,所构建的行业分类系统,在三个指标中均超过了当前最先进的基于算法的 ICS, TNIC。

第二, 本文提出了一种新的高效 Transformer 模型, LongBERT。10K 报告的最大文本长度可达 100K tokens 以上, 这使得现有的高效 Transformers 无法直接处理, 需要先对文档进行分段。LongBERT 通过结合移位块注意力和全局注意力, 将其自注意力机制的计算和内存复杂度降低至 $O(n)$, 可以实现对长达 131K tokens 的文本进行处理, 突破了高效 Transformers 所能处理的最大文本长度的限制。通过批处理技巧, 我们能够高效实现 LongBERT 中的自注意力机制, 使得其在真实场景下的推理速度比复杂度同为 $O(n)$ 的 Longformer^[31] 快 1.87 倍。基于 LongBERT 的 Financial-ICS 在同行公司经济相关性的指标性中超过了专家设计的 ICS, SIC, 当我们提高相似度阈值至每个公司平均搜索 20 个同行公司时, 其性能可以超过当前最先进的专家设计的 ICS, NAICS。

第三, 本文提出了一种针对长文档的内存优化的对比学习方法。在使用长文档对模型进行对比学习微调时, 由于受到内存的限制, 通常导致对比学习期间的负样本数量稀少, 进而影响对比学习的性能甚至可能导致训练失败。为了解决这一问题, 我们设计了基于原型 (prototype) 的对比学习损失, 并对其内存消耗进行优化。在相同的批处理大小下, 使用内存优化的对比学习仅对模型效果产生略微下降, 但却可以显著降低内存消耗, 从而能够使用完整的 10K 报告对 LongBERT 进行对比学习微调。

第四, 本文提出了一种对表征模型进行 SHAP 解释的方法, 以增强用户对 Financial-ICS 的信任。SHAP^[65] 的通用实现 Kernel SHAP 需要一个返回概率的分类器函数, 这导致其与表征模型的不兼容。为解决这个问题, 我们通过对输入进行扰动, 并观察扰动前后模型生成的表征向量之间的余弦相似度大小, 来衡量输入中每个 token 对模型预测的贡献程度。我们通过将相似度分数映射到 0-1 之间以适配 SHAP。

1.4 本文组织结构

第 1 章绪论, 首先我们介绍本文研究的背景与意义, 包括行业分类系统 (ICSs) 在金融领域中的定义与作用, 对 ICSs 分类的阐述, 解释为何要研究基于算法的 ICSs, 并介绍当前最先进的基于算法的 ICSs 及其局限性。随后, 说明基于 Transformer 架构设计 ICS 的必要性和面临的挑战, 并引出本文研究的主要内容和创新性。

第 2 章相关工作, 首先介绍行业分类系统, 包括专家设计的 ICSs 和基于算法的 ICSs。随后介绍 Transformers, 包括传统 Transformer 架构、注意力优化的高效 Transformers 和递归 Transformers。最后, 我们将介绍表征学习, 包括传统表征学习、基于 Transformer 的表征学习方法以及表征学习的评估方法。

第3章自监督学习的行业分类系统研究,首先阐述我们提出的 Financial-ICS 的整体流程和 10K 报告数据集。接着介绍基于 RoBERTa 的 Financial-ICS 的自监督训练框架,并设计了三个自定义评估指标。最后进行实验与分析,并探索不同数据增强方法对对比学习效果的影响。

第4章基于高效 Transformer 的行业分类系统研究,首先介绍可以直接处理整个 10K 报告的 LongBERT 模型以及基于 LongBERT 的 Financial-ICS 的自监督训练框架。接着,介绍内存优化的对比学习方法,并对提出的模型、方法以及超参数的影响进行实验与分析。最后,在真实场景下对 LongBERT 进行推理速度和内存消耗的测试,并展示 Financial-ICS 的演示。

第5章行业分类系统的可解释性分析。首先介绍了当前常见的可解释方法,然后介绍适用于表征模型的 SHAP 解释方法。最后对训练后的 LongBERT 模型进行可解释性分析。

第6章总结与展望,将对研究成果进行总结,指出本研究的局限性,并展望未来研究方向。

第2章 相关工作

2.1 行业分类系统

行业分类系统 (Industry classification systems, ICSs) 在经济学、金融学和市场研究等领域中扮演着重要角色。它们为行业的组织提供了分类方法, 并为快速查找同行公司提供了途径, 有助于进行市场分析、竞争与风险评估和企业战略行为制定等。行业分类系统可以根据给定的焦点公司 (focal firm) 搜索与之经济相关的同行公司 (peer firms)。当前的 ICSs 主要可以被分为两类: 专家设计的 ICSs 和基于算法的 ICSs。

2.1.1 专家设计的行业分类系统

在构建专家设计的 ICSs 时, 领域专家设计了一个分层的行业结构, 并将公司分配到对应的行业中。

标准行业分类 (Standard Industrial Classification, SIC) 系统^[3]是最古老的行业分类系统, 于 1937 年在美国建立。其创立以来, SIC 系统被政府机构、学术和商业领域广泛使用。SIC 系统使用四位数字编码来区分不同的行业, 实现在统计和研究中的行业标准化。SIC 编码 (SIC code) 的前两位数字表示企业所属的主要行业部门, 而后两位则逐渐对所属行业进行细化。但是随着时间的流逝, SIC 编码逐渐被 NAICS 编码 (NAICS code) 所取代, 但仍存在一些美国政府和机构使用 SIC 编码。

北美行业分类系统 (North American Industry Classification System, NAICS)^[3]是为了适应美国和全球经济的快速变化, 由美国、加拿大和墨西哥政府合作共同制定, 发布于 1997 年。NAICS 是一种按照经济活动类型对企业进行分类的系统, 并提供了一种更统一的方式对行业进行分类, 在很大程度上取代了较旧的 SIC 系统。NAICS 使用六位数字进行编码, 前两位数字指定最大的业务部门, 后面几位数字不断对其进行细化。

全球行业分类标准 (Global Industry Classification Standard, GICS)^[3]由 MSCI 和标准普尔 (Standard & Poor's) 于 1999 年为全球金融领域设计的行业分类系统, 旨在帮助投资者和研究人员更好地了解全球各公司的主要业务活动。GICS 已经成为全球金融界和行业专家广泛接受的行业分类标准。

2.1.2 基于算法的行业分类系统

由于专家设计的 ICSs 存在开发和维护成本高昂、系统更新缓慢, 以及获取到的同行公司缺少经济相关程度排序等问题, 大量的研究人员开发了各种类型

的基于算法的 ICSs。基于算法的 ICSs 通过产生公司之间的相似度分数, 并进行排序来搜索同行公司。根据产生相似度方法的不同可以分为两类。

第一类是利用公司之间的共现频率来衡量它们之间相似度。Sprenger 和 Welp^[6]通过利用推特上的推文数据, 分析投资者在推文中对公司股票的讨论和提及频率来计算公司之间的相似度。投资者讨论股票时, 对两公司共同提及的频率越高, 则认为它们之间的经济相关性越高。基于搜索的同行公司 (Search-based Peer Firms)^[5]受到投资者感知对于同行公司重要性的启发, 采用“协同搜索 (co-search)”算法来分析美国证券交易委员会 EDGAR 网站上的互联网流量的搜索日志数据。通过同一个人在该网站时间先后搜索的公司频率来衡量两公司之间的相似程度, 以识别经济相关的同行公司。

第二类是为每个公司提取出经济表征, 并利用合适的相似度函数计算它们之间的相似度分数。Jaffe^[7]利用公司的专利特征将公司表示成一个向量, 然后通过对这些向量计算余弦相似度来获取同行公司。基于文本网络的行业分类 (Texted-based Network Industry Classifications, TNIC)^[4]认为产品相似度是行业分类的基础, 是寻找经济相关公司的关键。TNIC 使用 10K 报告中对公司产品和业务相关的部分 (Item 1 和 Item 1A) 作为算法的输入, 使用词袋 (bag-of-words) 的方法来获取蕴含公司产品特征的文档表征。对于某个文档, 词袋方法将产生一个与词表长度等长的二进制向量。若某词出现在文档中则置为 1, 若没有出现则置为 0。作者只关注文档中的名词和专有名字来组成词表, 并移除了文档中出现频率高于 25% 的常见单词。两个公司的产品相似度由两个文档表征之间的余弦相似度来定义。文档表征的相似度越高, 代表两公司的经济相关性越高。通过利用 10K 报告中对公司产品的描述, TNIC 可以获取任意两公司之间的产品相似度, 为每个公司查找其独有的竞争对手。词袋模型具有方法简单的优点, 但是缺点是会为每个文档产生上万维度的表征, 忽略了文本顺序且无法利用文档的上下文语义信息。

2.2 Transformers

Transformer^[8,66]由于其注意力机制在处理序列问题的出色能力, 使其在自然语言处理领域中被广泛使用, 包括翻译^[8,11,67]、语言模型^[10,24-25]和自然语言推断^[9,17,19]等。然而传统 Transformer 模型在处理长序列时存在时间和内存复杂度呈二次增长的问题, 限制了其在长序列场景下的应用。为了解决这一问题, 目前已经提出了很多方法。这里我们主要介绍两类方法: 一类是通过改进自注意力机制来降低复杂度的高效 Transformers^[30-40], 另一类是将循环神经网络的思想与 Transformer 相结合的递归 Transformers^[41-46]。

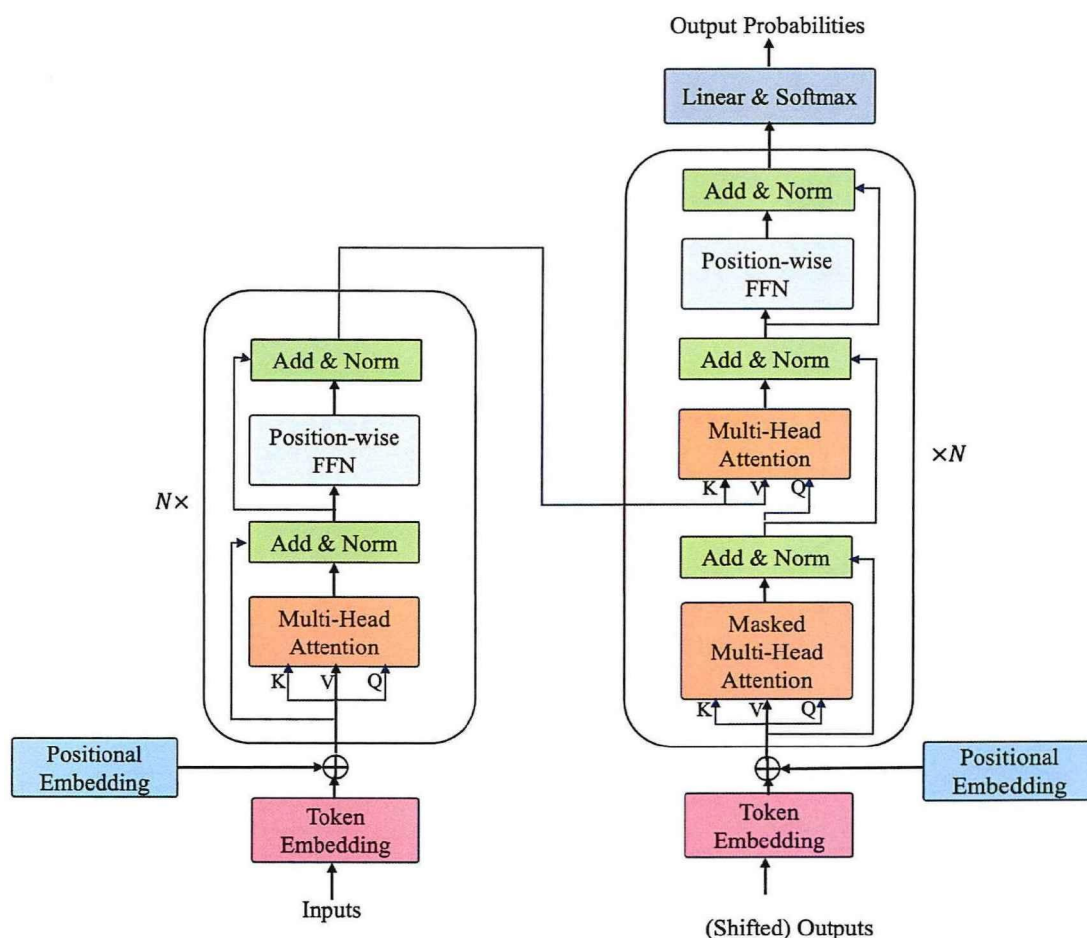


图 2.1 传统 Transformer 整体架构

2.2.1 传统 Transformer

最初的 Transformer^[8] 由 Google 提出, 是一个序列到序列的模型, 被应用于机器翻译任务。传统 Transformer 的整体架构如图 2.1 所示, 由一个编码器 (encoder) 和一个解码器 (decoder) 组成。

其中, 编码器由 N 个相同的层堆叠而成, 每个编码器层由一个多头自注意力 (multi-head self-attention) 模块和一个逐位置全连接前馈网络 (FFN) 组成。每个模块周围使用残差^[68]连接, 然后是层归一化 (layer norm)^[69]。计算公式如下:

$$\begin{aligned} H &= \text{LayerNorm}(X + \text{SelfAttention}(X)) \\ Z &= \text{LayerNorm}(H + \text{FFN}(H)) \end{aligned} \quad (2.1)$$

解码器也是由 N 个相同的层堆叠而成。不过与编码器层不同的是, 解码器层在多头自注意力模块和逐位置 FFN 之间额外插入了交叉注意力 (cross attention) 模块。与编码器层类似, 每个模块周围都使用残差连接, 然后进行层归一化。此外, 解码器中的自注意力使用了掩码自注意力, 限制了每个位置的查询 (query) 只能关注当前位置及其前面位置的所有键 (key) 值 (value) 对。在实现中, 使用

掩码函数对 softmax 归一化前的注意力矩阵中的非法位置进行掩盖（置为 $-\infty$ ），以确保位置 i 的预测只能依赖于位置小于 i 的已知输出。

介绍完 Transformer 的整体架构后，我们将对传统 Transformer 的关键组件，注意力模块进行详细介绍。其注意力公式如下：

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V = WV \quad (2.2)$$

其中 $Q \in \mathbb{R}^{n \times d_k}$, $K \in \mathbb{R}^{m \times d_k}$, $V \in \mathbb{R}^{m \times d_v}$, 其中 n, m 表示序列的长度。如果在自注意力模块中，则 $n = m$ 。 d_k 与 d_v 分别表示键（或查询）和值的维度， W 为注意力矩阵。这里，对查询和键的点积除以 $\sqrt{d_k}$ 以避免方差增大，将 softmax 函数推入梯度极小的区域，导致梯度消失问题。

为了更好的理解自注意力机制，我们可以从搜索和多层感知机（MLP）的角度来理解它。从搜索的角度来看，我们可以将查询看作是模型的上一层为了更好地理解文本向下一层提出的询问，而键和值可以看作是下一层所拥有的信息，其中键为信息关联的索引而值代表信息本身。如果询问 q_i 和信息索引 k_j 之间的关联性越强，那么它们之间的内积相似度就会越大，从而有更大权重的信息 v_j 从位置 j 流向 i ，实现了位置 i 到 j 的注意。从 MLP 的角度出发，我们可以将自注意力理解成动态的 MLP，只不过其中权重 W 不是可学习的参数，而是通过查询和键动态计算得出，通过计算 q_i 和 k_j 的内积相似度得到权重 w_{ij} 。因此自注意力机制中动态 MLP 的权重值是对内容敏感的，可以对关联性强的 tokens 产生更大的权重，从而进行注意。相比之下，普通的 MLP 是对内容不敏感的，其权重的值是确定的。

Transformer 不仅使用单个注意力函数，而是使用多头注意力。它通过将 d_m 维的原始查询、键和值通过 h 次不同的线性投影，投影到 d_k 、 d_k 和 d_v 维度，然后并行地对这些投影得到的查询、键和值执行注意力函数。随后，将多头注意力的输出进行拼接，并再次投影回 d_m 维，得到最终的值。多头注意力使得模型能够同时关注不同位置、不同表征子空间中的信息。多头注意力的公式如下：

$$\begin{aligned} \text{MultiHeadAttn}(Q, K, V) &= \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \\ \text{where head}_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \quad (2.3)$$

在公式中， $W_i^Q \in \mathbb{R}^{d_m \times d_k}$, $W_i^K \in \mathbb{R}^{d_m \times d_k}$, $W_i^V \in \mathbb{R}^{d_m \times d_v}$, $W_i^O \in \mathbb{R}^{hd_v \times d_m}$, h 为注意力头的个数， d_m 表示注意力模块的输入和输出维度。

除了注意力模块外，逐位置 FFN 是编码器层和解码器层中都会包含的模块。这个网络是由两个线性映射和一个 ReLU 激活函数组成。其计算公式如下：

$$\text{FNN}(H) = \text{ReLU}(HW_1 + b_1)W_2 + b_2 \quad (2.4)$$

其中 $W_1 \in \mathbb{R}^{d_m \times d_f}$, $W_2 \in \mathbb{R}^{d_f \times d_m}$, $b_1 \in \mathbb{R}^{d_f}$, $b_2 \in \mathbb{R}^{d_m}$ 。通常情况下 FFN 的中间维度 d_f 被设置为大于 d_m 。由于逐位置 FFN 对每个位置进行单独且相同的处理, 所以它与自注意力模块一样也是位置不敏感的。

位置编码同样是 Transformer 模型中的重要组成部分。由于 Transformer 没有引入循环神经网络 (RNN) 或者卷积神经网络 (CNN), 且自注意力模块和逐位置 FFN 都是对位置信息不敏感的, 然而在处理序列问题时, 顺序信息通常扮演着重要角色, 例如在理解文本数据时, 单词的顺序至关重要。因此, 需要引入额外的机制将位置信息注入 Transformer 中。传统的 Transformer 使用正弦的绝对位置编码的方式引入位置信息。其公式如下:

$$\begin{aligned} \text{PE}(\text{pos}, 2i) &= \sin\left(\frac{\text{pos}}{10000^{2i/d_m}}\right) \\ \text{PE}(\text{pos}, 2i+1) &= \cos\left(\frac{\text{pos}}{10000^{2i/d_m}}\right) \end{aligned} \quad (2.5)$$

其中, pos 是位置, i 是维度。也就是说, 每个维度对应一个特定波长的正弦波。通过将词嵌入矩阵与位置嵌入矩阵相加来注入位置信息, 然后输入到 Transformer 中。作者选择使用正弦式的位置编码而不是可学习的位置编码, 因为这两种编码方式效果相似, 但是正弦式的可以让模型在推理时推广到比训练中遇到序列长度更长的情况。

Transformer 从使用的角度可以被划分为 3 类, 分别是仅编码器、仅解码器和编码器-解码器结构。仅编码器 (encoder-only) 的 Transformer 具有双向的感受野, 主要用于自然语言理解 (NLU) 的任务, 如句子分类、自然语言推断、命名实体识别和句子表征提取等, 代表模型有 BERT^[9]、RoBERTa^[17]、ALBERT^[18] 和 ELECTRA^[20] 等。仅解码器 (decoder-only) 的 Transformer 只使用了解码器, 并去除了其中的交叉注意力模块。由于其自注意力机制中的掩码机制, 每个位置的查询只能注意其自身及其左边的 tokens, 因此主要处理序列生成的任务, 代表模型有 GPT^[10,24-25]、Transformer-XL^[41] 等。编码器-解码器 (encoder-decoder) 的 Transformer, 即 Google 最早提出的 Transformer 结构, 主要用于序列到序列任务的建模, 例如翻译任务, 代表模型有 BART^[11]、T5^[67] 等。

最后, 我们对 Transformer 结构进行复杂度分析。自注意力模块和逐位置 FFN 是 Transformer 中的两个核心组件。其中, 自注意力模块的计算复杂度为 $O(n^2 d_m)$, 其中 n 表示序列长度, d_m 表示模型的隐藏维度, 这里不考虑自注意力模块前后的线性映射。逐位置 FFN 的计算复杂度为 $O(nd_m^2)$ 。因此 Transformer 的计算复杂度为 $O(n^2 d_m + nd_m^2)$, 当序列长度 n 较长时, n 会主导 Transformer 的复杂度成为计算瓶颈。此外, 自注意力的计算需要存储 $n \times n$ 的注意力矩阵 W , 这使得 Transformer 的内存复杂度为 $O(n^2)$, 导致其在处理长文本时变得不可行。

2.2.2 高效 Transformers

注意力模块赋予了 Transformer 处理长序列的强大能力, 但是其 $O(n^2)$ 的复杂度限制了其在长序列场景下的应用。在标准的自注意力机制中, 每个 token 都需要与其他所有的 tokens 进行关注。但是研究人员发现, 经过预训练的 Transformer 中学习到的注意力矩阵通常是稀疏的或者低秩的^[30,36]。高效 Transformers^[29,66]通过改进传统 Transformer 中的自注意力机制来降低其计算和内存复杂度。

现有的对自注意力机制的改进^[66]主要可以被分为以下方向: 位置稀疏的注意力、内容稀疏的注意力、基于 kernel 的注意力、使用查询原型 (prototype) 的注意力和压缩键值对的注意力。

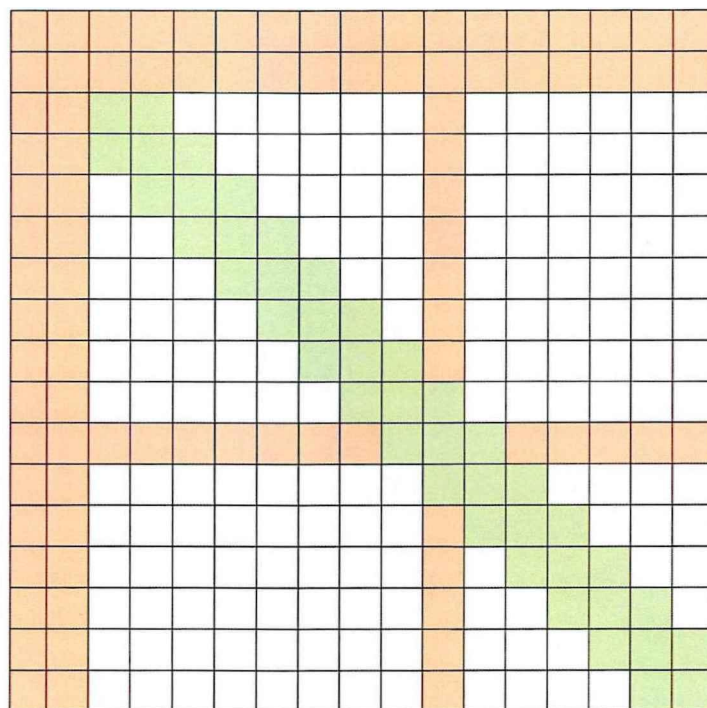


图 2.2 Longformer 的注意力机制

位置稀疏的注意力通过引入结构偏置来限制注意力机制中查询-键对的数量, 从而降低复杂度。它将不同种类的原子注意力相互结合, 以实现复杂度和长依赖学习能力的平衡。Longformer^[31]的注意力机制结合了窗口注意力 (window attention) 和全局注意力 (global attention), 如图2.2所示。其中, 绿色部分代表窗口注意力, 橙色部分代表全局注意力。窗口注意力通过限制每个查询只能关注其邻居节点, 有效利用了自然语言中信息局部性的特征, 并实现了 $O(n)$ 的计算和内存复杂度。全局注意力通过添加全局 tokens 作为信息流转的中间节点。全局 tokens 允许对所有的 tokens 进行注意, 同时, 所有的 tokens 也可以对全局 tokens 进行注意。全局注意力可以缓解使用局部注意力对建模长距离依赖关系能力的退化。Longformer 对全局 tokens 的选择通常是分类任务中的 CLS token 或者问

回答问题中的所有问题位置的 tokens。由于窗口注意力并不容易使用原生 torch 进行高效的实现, Longformer 通过实现自定义 CUDA 内核的方式来加速运算。Big bird^[32]提出使用图的视角看待自注意力, 将每个 token 理解成一个节点, 而两个节点之间的信息流动被视为边。比如, 完全注意力 (full attention) 被理解为完全图, 稀疏注意力 (sparse attention) 则被视为稀疏图, 而信息在图节点之间流动。Big bird 通过将窗口注意力、全局注意力和随机注意力 (random attention) 相结合实现了 $O(n)$ 的复杂度。其中随机注意力为每个查询随机采样键值对进行关注。Sparse Transformer^[30]首先对经过训练的自注意力网络学习到的注意力模式进行了可视化分析, 发现大多数层都表现出稀疏的模式。因此, Sparse Transformer 对注意力进行了分解, 并针对不同类型的数据设计了不同的稀疏模式。对于类似于图像和音乐等具有周期性的数据, 它使用窗口注意力和跨步注意力的组合。其中, 跨步注意力是指查询对每隔一定步数的 tokens 进行注意。而对于像文本这样无周期性的数据, 它使用局部块注意力和全局注意力的组合。

内容稀疏的注意力指的是其稀疏模式并非固定在特定位置, 而是通过选择那些较大概率与给定查询有较高相似度的键进行关注, 而不关注其他的键, 从而降低复杂度。Reformer^[33]使用局部敏感哈希 (LSH) 对每个查询选择键值对。LSH 注意力规定每个查询只能关注同一个哈希桶中的 tokens。通过使用 LSH 函数将查询和键哈希到若干桶中, 相似的向量会有更大概率落入同一个桶中。LSH 函数的计算公式如下:

$$h(x) = \operatorname{argmax}(x^T [R; -R]) \quad (2.6)$$

其中 h 是哈希函数, $R \in \mathbb{R}^{d \times b/2}$ 是一个随机矩阵, 矩阵中的每一列代表一个随机的桶向量, d 为输入的维度, b 为桶的个数, $[R; -R]$ 表示沿桶数维度上的拼接。分配到同一个桶向量中的 q_i 和 k_j 被认为有更大概率出现较高的内积相似度, 所以只需要对映射到同一个桶中的查询-键对进行注意力计算, 而无需计算所有位置查询-键的注意力。第 i 个查询所关注的键索引如下:

$$P_i = \{j : h(q_i) = h(k_j)\} \quad (2.7)$$

其中 P_i 表示第 i 个位置的查询需要关注的键的位置索引集合, 从而将其计算和内存复杂度降至 $O(n \log n)$ 。

基于 kernel 的注意力通过改变注意力计算中矩阵相乘的顺序来降低复杂度。假设 $Q, K, V \in \mathbb{R}^{n \times d}$, 那么计算 $\operatorname{softmax}(QK^T)V$ 的复杂度是 $O(n^2)$ 的。如果可以通过某些办法将 $\operatorname{softmax}(QK^T)V$ 中的 $\operatorname{softmax}$ 非线性函数移除, 那么原式将变成 QK^TV 。此时我们可以利用矩阵乘法的结合率, 以相反的顺序计算 $Q(K^TV)$, 从而将复杂度降低为 $O(n)$ 。这里使用 kernel 的技巧, 来移除注意力中的 $\operatorname{softmax}$ 非线性函数。通过将相似度函数 $\operatorname{sim}(q, k) = \exp\left(\frac{q^T k}{\sqrt{d_k}}\right)$ 比作 kernel, 令 $\operatorname{sim}(q, k) =$

$\phi(q)^T \phi(k)$, 从而消除非线性函数:

$$\begin{aligned} \text{Attention}(q_i, K, V) &= \frac{\sum_{j=1}^n \text{sim}(q_i, k_j) v_j^T}{\sum_{j=1}^n \text{sim}(q_i, k_j)} = \frac{\sum_{j=1}^n \phi(q_i)^T \phi(k_j) v_j^T}{\sum_{j=1}^n \phi(q_i)^T \phi(k_j)} \\ &= \frac{\phi(q_i)^T \sum_{j=1}^n \phi(k_j) v_j^T}{\phi(q_i)^T \sum_{j=1}^n \phi(k_j)} \end{aligned} \quad (2.8)$$

Linear Transformer^[37]对映射函数 ϕ 的选择使用经验上的函数,采用简单的 $\phi(x) = \text{elu}(x) + 1$ 。这个特征映射并不是为了近似注意力函数,而是从实验的角度验证其性能与标准 Transformer 相当。

使用查询原型的注意力通过减少查询的数量来降低注意力的复杂度。Informer^[34]使用查询稀疏度测量 (query sparsity measurement) 从查询中进行选择,通过计算查询的注意力分布与均匀分布之间的 KL 散度的近似得出。如果它们之间的 KL 散度越大,则认为查询的注意力分布与均匀分布的距离越远,也代表该查询更需要被考虑。Informer 只对查询稀疏度测量排名最靠前的 k ($k = O(\log n)$) 个查询计算注意力分布,而其他查询的注意力分布直接采用均匀分布,从而将复杂度降低为 $O(n \log n)$ 。

压缩键值对的注意力通过减少键值对的数量来降低注意力的复杂度,即在计算注意力之前降低键值的维度:

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QF(K)^T}{\sqrt{d_k}} \right) F(V) \quad (2.9)$$

其中 F 为降维函数。Memory compressed attention^[35]通过使用带步幅的卷积 $F = \text{Conv1D}$ 来减少键值的数量,该方案被作者用于局部注意力的补充,以实现全局语义的捕获。该机制将键值对的数量减少了卷积步幅倍,从而可以在相同计算资源条件下处理更长的序列。Linformer 首先对预训练 Transformer 中 softmax 归一化后的注意力矩阵进行了谱分析,发现其奇异值分布呈现明显的长尾特征,并从理论的角度证明注意力矩阵可以使用低秩矩阵近似表示。通过使用线性映射 $F = \text{Linear}$, 将键和值从原始长度 n 投影到一个较小长度 k (k 为常数),从而将其复杂度降低为 $O(n)$ 。但是使用这种方式降低键值数量的缺点是,由于线性映射函数的特点,它需要假设模型的输入序列长度是固定的。

2.2.3 递归 Transformers

递归 Transformers 提供了解决 Transformer 无法有效处理长文本问题的另一种方案。递归 Transformers 通过将循环神经网络的思想与 Transformer 相结合来处理长文本。如图2.3所示,递归 Transformers 将长文本切成 Transformer 可以直接处理的文本段,并维护一系列隐藏状态或者说是记忆 (memory) 来存储文本

的历史信息。处理文本段时，模型会从记忆中读取额外的信息输入模型参与注意力的计算。当处理完成后，将当前文本段的中间运算状态或者结果重新加入记忆中。递归 Transformers 通过合适的记忆管理方法，通常可以实现超长文本的处理，甚至对序列长度是不受限的。但是单向的隐藏状态传递机制导致了递归 Transformer 更适合处理仅编码器架构的 Transformer 模型。

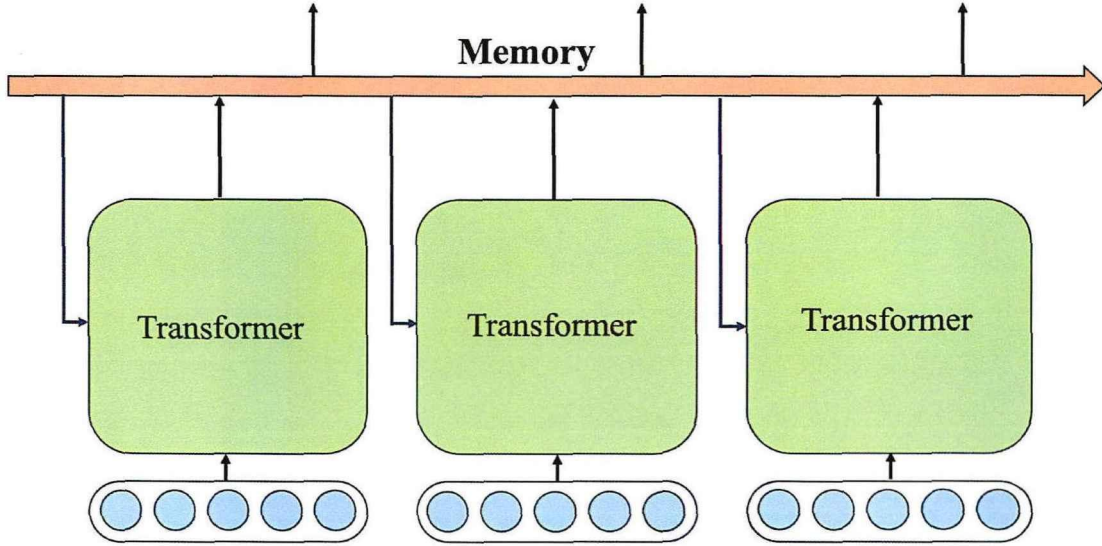


图 2.3 递归 Transformers 的示例图

Transformer-XL 是^[41]递归 Transformer 的奠基之作。该模型通过将前一文本段下一层的隐藏状态作为补充的键和值参与注意力，以解决语言模型生成长文本时每一段开始时的上下文碎片问题，并促使段与段之间的信息流动。第 $\tau + 1$ 个文本段的第 l 层的隐藏状态 $H_{\tau+1}^l$ 可以通过以下公式获得：

$$\begin{aligned}\tilde{H}_{\tau+1}^{l-1} &= [\text{SG}(H_{\tau}^{l-1}); H_{\tau+1}^{l-1}] \\ Q_{\tau+1}^l, K_{\tau+1}^l, V_{\tau+1}^l &= H_{\tau+1}^{l-1} W^Q, \tilde{H}_{\tau+1}^{l-1} W^K, \tilde{H}_{\tau+1}^{l-1} W^V \\ H_{\tau+1}^l &= \text{TransformerLayer}(Q_{\tau+1}^l, K_{\tau+1}^l, V_{\tau+1}^l)\end{aligned}\quad (2.10)$$

其中 SG 表示停止梯度传播运算， $[X; Y]$ 表示沿序列长度方向进行拼接。通过使用这个方法，Transformer-XL 可以实现 $L \times N_{mem}$ 的最大历史上下文长度，其中 L 为模型层数， N_{mem} 为缓存记忆的长度。此外，Transformer-XL 通过使用正弦相对位置嵌入替代绝对位置嵌入，以解决前后两个文本段相同位置嵌入带来的位置信息紊乱问题。

Compressive Transformer^[42]设计了二级的记忆结构，包括短期细粒度记忆和压缩的长期粗粒度记忆。在 Transformer-XL 中，将来自前一段的激活缓存成记忆用于增强当前片段，而将较旧的激活进行丢弃。而 Compressive Transformer 在较旧的激活上使用压缩函数，例如卷积、池化等，然后将其存储在压缩记忆中。记忆和压缩记忆都采用先入先出的策略。通过引入短期记忆和长期记忆，Compressive

Transformer 改进了 Transformer-XL, 进一步提升了模型可以关注到的最大历史上下文长度至 $L \times (cN_{cm} + N_{mem})$, 其中 c 为记忆压缩率, N_{cm} 为压缩记忆的长度, N_{mem} 为原始记忆的长度。

ERNIE-DOC^[43] 也是对 Transformer-XL 的改进之一。Transformer-XL 使用了前一文本段下一层的隐藏状态作为补充的键值, 这导致了其感受野的不足。为了解决这一问题, ERNIE-DOC 通过将前一段下一层的选取方式改成了前一段同一层的方式, 从而实现了无限的感受野。

Memorizing Transformer^[44] 在 Transformer-XL 的基础上引入了外部记忆, 以增强模型获取更长历史上下文信息的能力。Transformer-XL 的内部记忆机制确保 Memorizing Transformer 模型中的每个 token 都有大于 512 tokens 的上下文信息长度。新引入的外部记忆在模型的最后一层的输入处加入, 作为额外信息的补充。外部记忆与内部记忆一样都不进行梯度的计算, 这使得作者可以轻松地将记忆长度扩展到非常长的 tokens 数, 例如 262K tokens。由于外部记忆的规模很大, 所以作者首先使用近似 KNN 算法快速搜索出与当前本文查询相似度较大的 n 个键值对, 然后进行注意力计算, 并使用可学习的门控机制将其结果注入模型。

2.3 表征学习

由于长文档表征学习和句子表征学习的任务很相似, 并且已经有许多研究致力于句子表征学习, 因此本节将介绍与句子表征学习相关的工作。

句子表征学习是自然语言处理中的关键问题, 使用低维数值向量对句子的含义进行描述。句子表征学习主要可以分为两种, 一种是上下文无关的句子表征学习, 另一种是上下文相关的句子表征学习。上下文无关的句子表征通过直接使用单词嵌入的聚合来产生句子表征。例如, 可以对句子中的所有单词的嵌入取平均值或者加权平均值。这种方法的本质是对词嵌入进行学习。本节主要介绍对词嵌入进行学习的传统表征学习、上下文相关的基于 Transformer 的表征学习, 以及句子表征学习的评估方法。

2.3.1 传统表征学习

传统表征学习通过学习词嵌入, 并对句子中的所有单词的嵌入进行聚合, 以产生句子表征。然而, 这种方法的缺点是, 无法利用单词的顺序信息和上下文语义信息, 例如句子中的多义词和指代关系等。

词袋 (bag-of-words) 模型是自然语言处理中最简单但广泛使用的模型, 用于将文本数据转换成数值向量。词袋模型首先会构建一个单词表, 其中包含了在文本中出现的所有单词。对于每段文本, 词袋模型计算每个单词在文本中出现的

频次,并将其转换成一个长度与词表等长的特征向量。但是词袋模型产生的嵌入具有维度高的缺点。

为了解决这个问题, Word2Vec 被提出。Word2Vec^[70-71]是一种基于神经网络的模型,用于学习单词的低维向量表示。它包括 CBOW (continuous bag-of-words) 和 Skip-Gram 两种训练方式。CBOW 使用类似于完形填空的任务,通过给定一个窗口内的上下文单词来预测中心的目标单词。Skip-Gram 则正好相反,通过给定中心的目标单词来预测与之相关的上下文单词。

GloVe (Global Vectors for Word Representation)^[72]通过将全局矩阵分解技术和局部上下文相结合来学习词嵌入。它利用大规模文本语料中局部上下文窗口中的单词共现统计信息构建全局的词-词共现矩阵,然后通过对这个共现矩阵进行矩阵分解,来获取每个单词的向量表示。在这些向量表示中,两个单词的词向量内积会接近它们在语料库中的共现频率。如果两个单词同时出现的频率越高,那么它们词向量之间的内积相似度也会越大。

2.3.2 基于 Transformer 的表征学习

Transformer 的出现推动了句子表征学习^[73]的发展。基于 Transformer 的表征学习属于上下文相关的表征学习方法,其训练过程可以分为两个阶段。首先使用 BERT 系列的预训练方法,在大型语料库中对仅编码器 (encoder-only) 的 Transformer 模型进行预训练,从中学习通用的语言表示。然而,预训练语言模型生成的表征存在各向异性问题^[48],因此需要对其进行微调,以产生具有实际意义的句子表征。

首先,我们介绍预训练部分。对 Transformer 模型进行预训练^[74-75]具有以下优点。首先,在自然语言处理任务中,构建大规模标注数据集是一个巨大的挑战,而 Transformer 的预训练过程是自监督的学习任务。通过在大规模文本语料中进行预训练,可以学习到通用的语言表示,从而提高模型在下游任务的效果。其次,Transformer 模型结构不包含任何先验结构,预训练可以提供更好的模型初始状态,避免在小数据集上训练出现过拟合现象,以产生更好的泛化性能。

最近提出了许多 BERT 系列的预训练方法^[9,17-23]。通过在大型语料库中进行预训练,这些模型在许多下游的自然语言理解任务中取得了最佳的性能。这些 BERT 系列的预训练方法都只使用了 Transformer 中的编码器部分。BERT^[9]设计了两个预训练目标:掩码语言模型 (Masked LM) 和下一句预测 (Next Sentence Prediction, NSP)。Masked LM 任务类似于完形填空任务。它会将输入句子中 15% 的 tokens 进行掩盖,然后训练模型使用其余的 tokens 来预测被掩盖的 tokens。由于掩码 token ([MASK]) 不会在微调阶段出现,为了避免预训练方式与下游微调任务之间的不匹配,在掩盖 token 时,80% 的概率使用特殊的掩码 token,10% 概

率使用随机令牌, 10% 概率使用原始令牌。NSP 任务则是判断分隔符前后的两个输入句子是否是来自同一段语料的连续片段。其中, 50% 的概率为第二个句子是第一个句子的下一个句子, 另外 50% 的概率为两个句子是来自语料库的随机句子。作者认为, 这有助于让模型理解两个句子之间的关系。

许多研究工作对 BERT 的预训练方法进行了改进。RoBERTa^[17] 去除了 NSP 任务, 只使用 Masked LM 的方式进行训练。因为它发现 NSP 任务对下游任务的性能是有害的。此外, RoBERTa 采用了动态掩码机制, 在训练过程中对文本进行动态掩码, 这与 BERT 在创建预训练语料时提前进行掩码的方式不同。ALBERT^[18] 提出了使用句子顺序预测 (sentence-order prediction, SOP) 任务替代 NSP 任务。在 SOP 任务中, 正样本的选取方式与 NSP 相同, 但负样本通过选择同一段语料中的连续片段并交换它们的顺序来构造。作者认为 NSP 任务缺乏难度, 因为 NSP 任务的负样本是使用语料库中随机选择的句子, 这导致负样本在主题和连贯性两个方面都不一致, 而主题预测比连贯性预测更容易。SpanBERT^[21] 提出了跨度掩盖 (Span Masking) 方案, 使用几何分布选择一个跨度范围进行掩盖, 而不同于 BERT 中仅对最小输入单元 token 进行掩盖。另外, SpanBERT 设计了新的预训练目标: 跨度边界目标 (Span Boundary Objective, SBO), 使跨度边界的词向量能够对掩盖掉的跨度内部的 tokens 进行预测。此外, SpanBERT 去除了 NSP 任务, 直接使用连续的长句进行训练。

但是最近的研究发现, 预训练的语言模型, 包括使用 Masked LM 目标进行预训练的模型, 其输出层存在各向异性现象。各向异性现象^[50]是指模型输出层的表征通常聚集在一个狭小的向量空间中, 其词嵌入矩阵的奇异值分布极不均匀, 出现急剧衰减, 除了少数几个主导奇异值非常大, 其他的值都会接近于 0。各向异性现象主要描述嵌入空间的均匀性问题, 嵌入应该均匀地分布在向量空间中。聚集的向量分布导致这些句子表征之间的相似度很高, 影响了 BERT 系列模型在语义文本相似度 (STS) 任务中的表现。为了解决各向异性问题, 研究者们提出了多种方法, 其中以 SimCSE^[52] 为代表的基于对比学习的句子表征学习方法, 取得了最好的效果。

BERT-flow^[49] 和 BERT-whitening^[76] 通过将 BERT 的嵌入空间转换成各向同性的空间, 从而缓解了这个问题。Sentence-BERT^[51] 采用孪生网络架构, 使用自然语言推断任务对预训练的 BERT 或者 RoBERTa 模型进行微调。在自然语言推断任务中, 模型需要根据给定的前提 (premise) 和假设 (hypothesis) 判断假设是否可以从前提中推断出来, 通常包括三个分类: 蕴含关系 (entailment)、中立关系 (neutral) 和矛盾关系 (contradiction)。Sentence-BERT 将孪生网络对两句子产生的表征向量 u 、 v 和 $|u - v|$ 拼接到一起作为最终分类器的输入, 并使用交叉熵函数进行优化, 来产生有意义的句子表征。

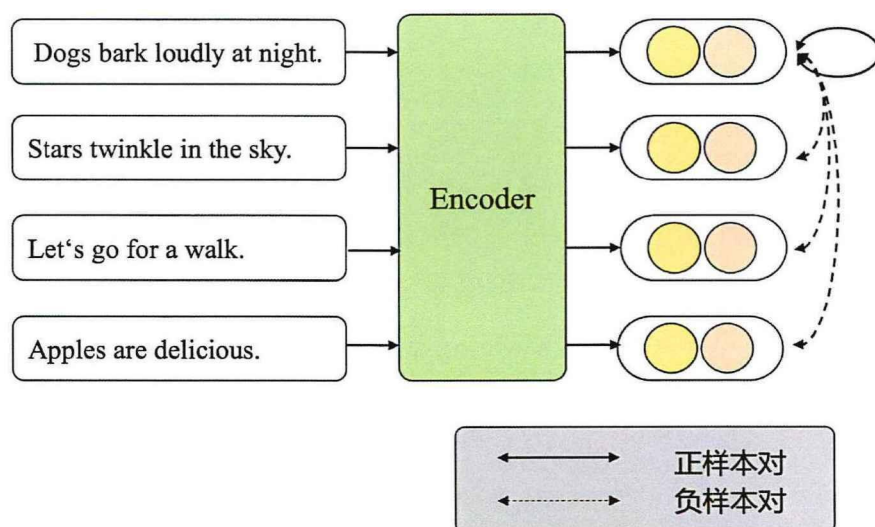


图 2.4 无监督 SimCSE 的示例图

受到计算机视觉中 SimCLR^[61,77] 对比学习框架的启发, 研究人员开始采用对比学习的方式来改进类似于 BERT 的预训练模型的句子表征。在对比学习中, 正样本通常是同一个样本的不同视图, 通过数据增强策略生成, 而负样本通常是同一批处理中的其余样本。通过使用对比学习, 可以增加正样本对之间的相似度, 降低负样本对之间的相似度, 从而缓解嵌入空间均匀性的问题。对比学习框架可以分成三个组件: 数据增强、输出层的聚合函数以及对比学习损失。目前研究最多的是数据增强方向。由于对比学习的其中一个训练目标是将正样本对拉近, 因此正样本对的质量对于句子特征提取至关重要, 而正样本对的产生依赖于数据增强, 采用什么类型的数据增强方法极大影响表征学习的有效性。已经有不少研究提出了各种不同的文本增强策略, 包括文本随机删除与替换、dropout 增强、同义词替换、回译 (back translation)^[78] 等。输出层的聚合函数方面, 主要包括全局平均池化、全局最大池化或选择 CLS token 的方式来生成最终的句子表征向量。由于全局最大池化的效果并不理想, 所以一般在全局平均池化和 CLS token 之间进行选择。另外, 损失函数的设计同样是对比学习的关键组成部分。现有的对比学习损失主要是 InfoNCE 损失^[79] 的变种, 并选择合适的距离函数进行优化。目前广泛使用的距离函数主要包括欧式距离、余弦相似度和内积相似度等。

SimCSE^[52] 使用最小数据增强策略 dropout^[80] 对同一句子进行两次编码来获取正样本对, 而将来自不同句子样本的表征视为负样本, 如图 2.4 所示。其使用的对比学习损失如下:

$$l_i = -\log \frac{e^{\text{sim}(h_i, h_i^+)/\tau}}{\sum_{j=1}^N e^{\text{sim}(h_i, h_j^+)/\tau}} \quad (2.11)$$

其中, sim 表示余弦相似度, h_i , h_i^+ 表示模型对相同句子使用不同的 dropout 掩码得到的第 i 个样本的表征向量, 它们互为正样本对, τ 表示温度 (temperature)

超参数。由于 dropout 已经是 Transformer 模型的一部分，因此这是一种非常简单且有效的文本表征学习方法。此外，SimCSE 将 dropout 的数据增强方法与其他增强方法进行对比，包括裁剪、词语删除、掩码、不使用 dropout 和同义词替换等。结果显示，使用 dropout 的效果要优于其他更复杂的数据增强方法。而且在无监督的情况下，SimCSE 在 STS 任务中的表现超过了最先进的有监督的句子表征模型 Sentence-BERT。SimCSE 还可以通过引入自然语言推断数据集，将其中矛盾的假设作为困难负样本，以构建对比学习损失进行有监督的学习，从而优化了无监督学习的效果。其构建的损失函数如下：

$$l_i = -\log \frac{e^{\text{sim}(h_i, h_i^+)/\tau}}{\sum_{j=1}^N \left(e^{\text{sim}(h_i, h_j^+)/\tau} + e^{\text{sim}(h_i, h_j^-)/\tau} \right)} \quad (2.12)$$

其中， h_i 是由前提产生的表征向量， h_i^+ 和 h_i^- 分别表示由蕴含和矛盾的假设产生的表征向量。

随着 SimCSE 的成功，通过对比学习微调 BERT 系列模型以生成高质量句子表征的方法越来越受到欢迎。ConSERT^[54]通过对句子在词嵌入层进行变换来获得增强的样本，包括 tokens 顺序扰乱、tokens 裁剪、特征裁剪和 dropout。Tokens 顺序扰乱通过直接对位置编码进行扰乱来实现。Tokens 裁剪是指随机选择词嵌入矩阵的 tokens，将其设置为 0。特征裁剪是指随机选择特征，将其设置为 0。Dropout 则是以一定概率随机对嵌入中的元素置为 0。在实验中，ConSERT 随机选择了两种数据增强方法形成样本对，并发现使用 tokens 顺序扰乱加特征裁剪的增强方式在 STS 任务中表现最佳。ESimCSE^[53]在 SimCSE 的基础上进行了改进。其动机是 SimCSE 构建的所有正样本对具有相同的长度，这可能导致模型错误的将其作为区分正负样本的特征。因此，ESimCSE 使用词重复来改变输入句子的长度，同时保证语义不变。相比之下，其他数据增强方式，例如删除或替换，往往会改变句子本身的语义。

2.3.3 表征学习的评估方法

当前对句子表征质量进行衡量的评估任务主要可以分为两类，分别是文本相似度（semantic text similarity, STS）任务和迁移学习任务。

STS 任务^[59-60]是评估句子表征的最常见方法，已经被很多句子表征模型所使用^[52-54]。STS 任务的数据集包括 STS 2012-2016 和 STS-B。在这些数据集中，每个样本由一组句子对和人工标注的分数组成，分数范围从 0 到 5，分数越高代表句子对的语义相似度越高。为了评估句子表征的质量，我们首先需要使用模型获取测试数据集中每个样本对的语义相似度分数，然后将得到的一系列相似度分数与人工标注分数之间计算 Pearson 相关系数或者 Spearman 相关系数。相关

系数越接近 1, 代表嵌入模型产生的表征质量越高。Spearman 相关系数被认为比 Pearson 相关系数适合评估句子表征, 因为它衡量的是排名而不是实际分数^[81]。

迁移学习任务^[61-62]是对 STS 任务评估句子表征质量的补充, 通常采用分类任务加逻辑回归的策略进行评估。文本表征的可迁移性表现在其可以捕获文本中丰富的语义特征, 并将这些特征向量应用于下游任务中。对于分类任务, 我们首先使用冻结参数的模型对某数据集中的文本进行处理, 产生若干特征向量, 然后将这些特征向量作为逻辑回归分类器的输入。最后, 以分类准确率作为模型在该数据集上性能的评估指标, 较高的分类准确率说明模型生成的表征具有更强的特征提取能力。

第3章 自监督学习的行业分类系统研究

3.1 引言

基于算法的 ICSs 的核心在于训练一个能够生成公司高质量经济表征的模型, 然后利用该模型生成的表征来寻找同行公司。Transformer 在自然语言处理领域的多个任务下取得了惊人的成就, 其提出的自注意力机制将长依赖学习的路径长度降低为 $O(1)$, 使得 Transformer 在处理长文本时具有独特的优势。在本章中, 我们提出了自监督学习的 Financial-ICS, 将 Transformer 架构的模型引入到基于算法的 ICS 中。我们使用公司 10K 报告金融长文本作为数据集, 并采用 RoBERTa 模型^[17,82]来验证提出的自监督训练方法。我们对长文本段进行分段表征以生成公司的经济表征, 并设计了三个自定义指标来对其进行评估。

3.2 Financial-ICS 的整体流程

Financial-ICS 是我们开发的基于算法的 ICS, 其工作流程与现有的基于算法的 ICSs 类似。Financial-ICS 选取公司 10K 报告中的 Item 1 Business 和 Item 1A Risk Factors 作为输入, 因为这些部分蕴含了丰富的公司产品和业务的信息。由于我们收集的 10K 报告数据集中最大文本长度超过 100K tokens, 因此在预测时通常需要将 10K 文档进行分段, 以便模型可以逐段处理, 例如 RoBERTa 最大处理 512 tokens 的文本。使用模型从 10K 报告中获取了所有公司的表征后, 我们采用余弦相似度来计算公司之间的经济相关性。相似度越高, 代表两公司的经济相关程度越高, 最后根据相似度分数进行排序并获取同行公司。

图3.1展示了 Financial-ICS 为 Facebook 公司查找同行公司的三个主要步骤。第一步, 使用模型对 10K 报告数据集中的文档进行处理, 产生所有公司的表征列表。第二步, 计算 Facebook 与其他公司的余弦相似度分数, 例如与苹果公司的相似度为 0.68, 与 Intel 公司的相似度为 0.49 等。我们通常会计算所有公司对之间的余弦相似度并保存下来, 该相似度矩阵可以在搜索其他公司的同行时被重新利用。第三步, 对相似度分数按从高到低进行排序, 并获取同行公司。我们需要设置一个相似度阈值, 选择超过阈值的公司作为同行, 例如推特公司等。

3.3 10K 报告数据集

TNIC^[4]认为, 产品相似度是识别经济相关公司的核心。10K 报告是上市公司每年需要向美国证券交易委员会 (SEC) 提交的年度报告。这份报告详细记录



图 3.1 Financial-ICS 为 Facebook 查找同行公司的整体流程

了公司的最新运营概况、财务状况和风险因素等。10K 报告中的业务描述部分 (Item 1 和 Item 1A) 会每年及时更新准确且丰富的公司产品相关的信息, 这部分文本是基于算法的 ICS 的理想输入。

10K 报告中的信息可以帮助投资者、监管机构和研究人员更全面的了解上市公司的经营状况和正面临的的风险。每份 10K 报告都包含 4 个 Parts 和 15 个 Items。本研究提出的 Financial-ICS 使用 Part 1 中的 Item 1 Business 和 Item 1A Risk Factors 作为数据集部分, 与 TNIC 中的选择一致。Item 1 主要介绍了公司的业务信息, 包含公司的核心业务、主要产品或服务的描述、市场定位和竞争环境、子公司及其发展计划, 以及提供公司的运营情况和战略方向等。而 Item 1A 主要披露公司正遇到的困难与外部挑战, 揭示可能对公司业务和财务状况产生重大影响的风险因素。

10K 报告可以直接从官方的 SEC 网站中获取。我们总共采集了来自 3514 个公司的 2 万 5 千条报告文档, 时间跨度从 2010 年 1 月 1 日到 2019 年 1 月 1 日。在制作数据集时, 我们去除了少于 1000 字符的文档, 并将数据集分为三个部分: 训练集、验证集和 demo 集。训练集包含了从 2010 年到 2017 年的数据, 共 22675 个文档, 用于模型的预训练和对比学习微调。验证集包含了 2018 年的报告数据, 共 1903 个文档, 用于评估一系列专家设计的 ICSs 和基于算法的 ICSs 的效果。Demo 集用于展示效果。在演示 Financial-ICS 搜索到的同行公司时, 我们希望涵

	文档数	公司数	行业数 (SIC)	Tokens 数	最大 Tokens 数
训练集	22675	3494	377	20.6k±11.7k	114.3k
验证集	1903	1898	340	23.3k±13.3k	113.1k
Demo 集	3345	3345	377	23.0k±12.7k	113.1k

表 3.1 10K 报告数据集的统计量

盖尽可能多的公司，因此引入了 demo 集。我们选取了自 2010 年以来每个公司最近提交的 10K 报告，认为最近的报告可以更及时地反映公司产品和业务的变化，总共包含 3345 个公司及其对应的 10K 文档。

我们使用 RoBERTa 的分词器对三个子数据集进行分词，统计量信息可以查看表 3.1。可以看到，收集的文档对应的公司总共分布在 377 个使用 SIC 编码进行分类的行业中。此外，三个数据集的平均长度超过 20K tokens，最大文本长度甚至超过 100K tokens。

为了对各个行业分类系统进行更细致的评估，我们收集了验证集文档数据外的其他补充数据。包括验证集中公司对应的 SIC 编码、NAICS 编码和 2018 年各公司的股市数据。鉴于公司之间的经济相关性可以通过它们之间股市趋势的相关性进行体现^[3]，因此同行公司股市数据的趋势相关性成为评估行业分类系统的重要指标。我们为验证集中的每个公司收集了 12 维度的股市数据趋势向量 $\mathbf{s}$ 。其中每个月作为一个数据点 s_i ，代表第 i 个月的股市数据。这些数据都会被用于评估 Financial-ICS。我们对含部分缺失数据的公司和文档进行了删除，构成了完整的验证集。

3.4 自监督训练框架

在查找经济相关的同行公司时，Financial-ICS 需要使用表征模型来提取公司 10K 报告的文档表征。因此，训练一个能够有效捕获 10K 报告文本语义的表征模型是实现先进的基于算法的 ICS 的关键。

我们提出了自监督学习的 Financial-ICS，将 Transformer 架构的模型引入到基于算法的 ICS 中。我们选择 RoBERTa 作为实验对象，以验证其有效性。在自监督学习框架下，训练标签是由算法自动生成，而无需人工标注。我们将表征模型的训练分为两个阶段：预训练阶段和微调阶段。

随着深度学习的发展，模型参数量急剧增加，为避免模型过拟合，我们需要对模型在大型文本语料库中进行预训练，以提高其泛化能力。鉴于我们的文档表征学习任务需要整合双向的语义信息，因此我们选择使用 Masked LM 目标进行预训练。我们采用了动态掩码机制，在训练过程中对文本进行动态掩码。

为了避免从头开始训练模型，我们使用 RoBERTa 模型在通用领域语料库上已经完成预训练的检查点 (checkpoint) 权重来初始化模型，这显著加快了模型

的收敛速度。

然而，我们仍需对模型在特定领域内进行继续预训练。我们使用 10K 报告数据集中的训练集部分，并对其进行分段作为 RoBERTa 继续预训练的语料，这样可以保证预训练的数据分布与下游任务中的是一致的^[83]，并增强模型对金融领域文本的理解能力。

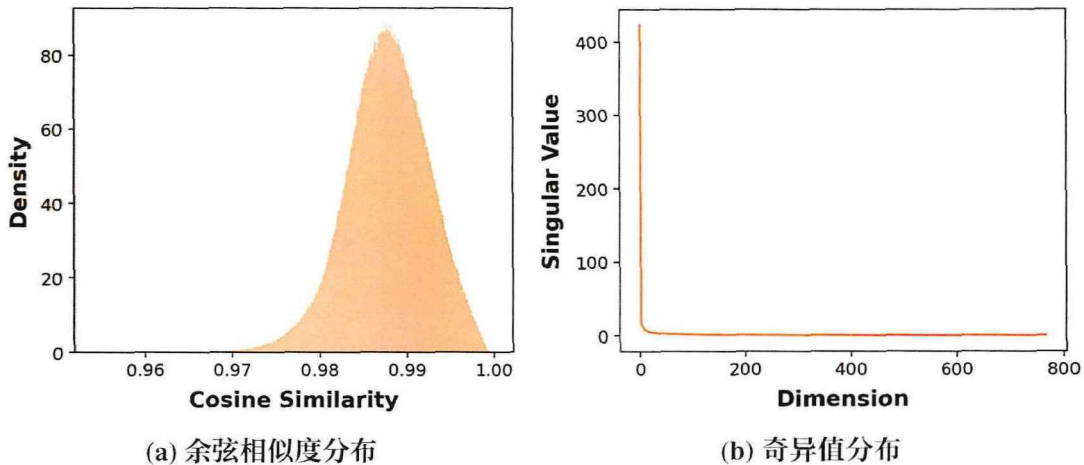


图 3.2 RoBERTa 在预训练后的文档表征的余弦相似度分布与奇异值分布

在预训练完成后，我们对 RoBERTa 生成的文档表征进行了分析。我们使用 10K 报告数据集中的验证集部分作为模型的输入，将文档切分成 RoBERTa 能够处理的长度，即 512 tokens，然后逐个进行处理，获取所有 tokens 的表征后进行平均池化。我们绘制了这些文档表征的余弦相似度分布和奇异值分布图，如图 3.2 所示。可以看出，RoBERTa 生成的文档表征存在聚集的情况，即任意文档表征之间的相似度分数普遍很高，几乎都在 97% 以上。此外，所产生的文档表征矩阵的奇异值分布极不均匀，除了少数几个奇异值很大，大多数的奇异值都接近 0，存在明显的各向异性问题^[48]。

为了缓解这一现象，我们使用对比学习的方法对模型进行微调。对比学习的目标是通过拉近语义相似的正样本对，同时推远不相似的负样本对，以增强样本间的区分度。而我们的目标是生成的文档表征可以均匀地分布在向量空间中，同时使经济相关的同行公司对应的文档表征之间具有更高的余弦相似度。这与对比学习的目标是一致的。

我们每次只选取文档中连续的 512 tokens 进行对比学习微调，并使用与 SimCSE^[52]中类似的方法来产生正负样本对。具体而言，将对同一文本段使用不同 dropout 掩码生成的表征作为正样本对，而不同文本段生成的表征作为负样本对。

经过表征学习后，我们再次对 RoBERTa 生成的文档表征进行了分析。从图 3.3 中可以看出，对比学习在很大程度上缓解了其生成的文档表征的各向异性问题。

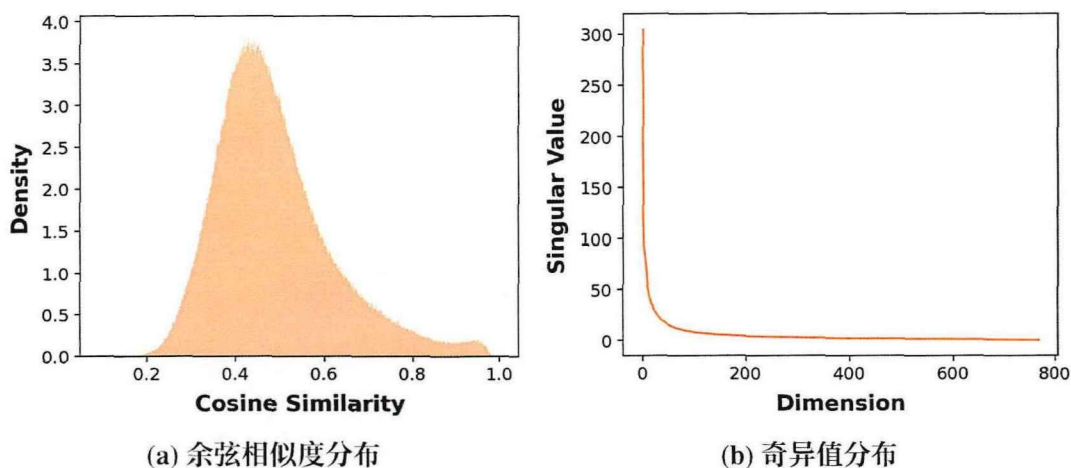


图 3.3 RoBERTa 在对比学习后的文档表征的余弦相似度分布与奇异值分布

3.5 评估指标

虽然现在已经有很多基于算法的 ICSs 被提出，但是缺乏对其进行评估的指标。为了对我们开发的基于自监督学习的 Financial-ICS 进行评估，我们从三个角度对先进的基于算法的 ICS 应该具备的特性进行了定义。首先，行业分类系统找到的同行公司应该是经济相关的，经济相关程度越高，说明该系统的性能越好。其次，基于算法的 ICSs 找到的同行公司应该与专家设计的 ICSs 中的行业结构存在一定的一致性。最后，由于基于算法的 ICSs 的本质是利用其表征模型生成特征丰富的公司经济表征并作相似度计算，因此对表征质量的评估也是对基于算法的 ICSs 评估的重要组成部分。我们从以上角度出发设计了三个自定义的指标。评估中需要使用的验证集的详细信息已经在第3.3节中给出，其中包含 10K 报告、SIC 编码、NAICS 编码以及 12 维度的股市数据向量 $\mathbf{s}$ 。

首先，经济相关的同行公司通常在股市数据上表现出相似的趋势性^[3]，例如，如果一个公司的股市趋势是增长的，那么其同行公司通常更大概率也是趋势增长的。这使得同行公司之间的股市数据相比随机选择公司通常具有更大的线性相关性。对于专家设计的 ICSs，例如 SIC 和 NAICS，我们将 SIC 编码或者 NAICS 编码完全相同的公司看作是同行公司。而对于基于算法的 ICSs，我们将它们产生的公司表征之间余弦相似度高于某一相似度阈值的公司认为是同行公司。我们为每个焦点公司计算其股市数据与其若干同行公司股市数据之间的 Pearson 相关系数，并求均值，作为我们的第一个评估指标。Pearson 相关系数越接近 1，意味着同行公司之间的经济相关性越高。这个评估指标主要用于反映行业分类系统所找到的同行公司与焦点公司之间的经济相关性。

第二，SIC 编码和 NAICS 编码是被广泛使用的专家设计的 ICSs，分别使用分层的 4 位和 6 位数字编码对行业进行分层，并为公司分配确定的行业编码。这些

编码可用作对基于算法的 ICSs 进行评估的人工标签。我们参考 STS 任务^[59-60]中评估指标的设计方法,设计了第二个评估指标。在 STS 任务中,每组句子都有一个相似度分数的人工标注,分数范围在 0 到 5 之间,分数越高表示这组句子之间的相似度越高。该任务使用每组句子表征的余弦相似度与相似度标签的 Pearson 相关系数或者 Spearman 相关系数来评估句子表征质量。我们使用 SIC 和 NAICS 分配给公司的 SIC 编码和 NAICS 编码来构建公司之间相似度分数的人工标签。我们计算了每对公司之间 SIC 编码和 NAICS 编码的最长公共前缀的长度,认为如果两个公司的编码具有更长的公共前缀,那么这两个公司更有可能是经济相关的同行公司,因此应该具有更高的相似度分数。公司间的标签相似度分数的计算公式如下:

$$\text{score}_{\text{sim}} = \max(0, \text{LCP}(c_1, c_2) - 1) \quad (3.1)$$

其中, LCP 代表最长公共前缀函数, c_1 和 c_2 分别代表两个公司的 SIC 编码或者 NAICS 编码。举例来说,微软公司的 SIC 是 7372,代表行业是服务-预包装软件(Services-Prepackaged Software);而推特和 Facebook 公司的 SIC 都为 7370,代表服务-计算机编程、数据处理等(Services-Computer Programming, Data Processing, Etc.)。因此,微软与推特的标签相似度分数为 2,推特和 Facebook 的标签相似度分数为 3。我们忽略了最长公共前缀为 1 情况下的相似度分数,这是因为 SIC 编码和 NAICS 编码使用最前面的两位进行主要行业部门的划分。两个公司的标签相似度越高,即它们在专家设计的分层行业中越靠近,那么使用基于算法的 ICSs 获取表征的余弦相似度也应该更高。以此为基础,我们计算公司经济表征之间的余弦相似度与我们按照规则计算的人工标注相似度分数之间的 Spearman 相关系数作为第二个评估指标。我们选择使用 Spearman 相关系数而不是 Pearson 相关系数,因为 Spearman 相关系数衡量的是排名而不是实际分数的线性相关系数,更符合此场景。Spearman 相关系数的指标越接近 1,那么该基于算法的 ICS 性能更佳。这个评估指标用于反映基于算法的 ICSs 搜索到的同行公司与专家设计的行业结构之间的一致性。

最后,我们设计了迁移学习^[61-62]任务对模型产生的表征质量进行评估。在句子表征质量评估中,迁移学习任务通常是对 STS 任务的补充。借鉴这个思路,我们构建了一个多分类任务,用以测试模型生成表征的可迁移性。我们将 SIC3 和 NAICS4 编码作为分类任务的标签,其中, SIC3 代表 SIC 编码中的前三位数字,而 NAICS4 代表 NAICS 编码中的前四位。将基于算法的 ICSs 中产生的公司表征作为逻辑回归分类器的输入,我们认为含有更丰富公司经济特征的表征应该更容易被线性分类,表现出更好的分类效果。在实验中,我们采用五折交叉验证来减少误差,并排除了样本数量小于 5 的类别。其中, SIC3 共 100 个类别, NAICS4 共 63 个类别。此外,我们选择使用 F1 作为评估指标而不是通常被使用

的准确率指标, 因为 SIC3 和 NAICS4 中的类别数很多, 且不同类别的样本数量差异很大, 使用准确率并不能很好的反映表征的综合性能。分类的 F1 值越高, 则认为该模型产生的表征蕴含的经济特征越丰富和清晰。

3.6 实验细节

3.6.1 基线

我们选择了 8 种方法作为我们设计的自监督学习的 Financial-ICS 的基线。首先, 我们选择了当前两种代表性也是最受欢迎的专家设计的 ICSs, SIC 和 NAICS, 作为实验的基线。

此外, 我们选择并设计了 6 种基于算法的 ICSs, 以下基于算法的 ICSs 使用 10K 报告数据集中的训练集部分作为算法的输入。其中, Random 方法随机产生 768 维度的表征向量。Bag-of-words 是目前最先进的基于算法的 ICS, TNIC^[4]所采用的方法。我们采用与 TNIC 相同的策略, 只使用文档中的名词和专有名词来构建词表, 并移除文档中出现频率高于 25% 的常见单词。我们最终得到的词表长度为 35K。对于每个文档, 我们生成一个与词表长度等长的二进制向量, 统计该单词是否在文档中出现。Avg. GloVe embeddings^[72]方法利用 GloVe 来生成词嵌入向量, 并通过计算整个文档的所有词嵌入向量的均值来产生最终的文档表征。GloVe 是传统表征学习的经典工作, 它利用数据集中单词与上下文词的共现统计关系来构建全局的词-词共现矩阵, 最后通过对这个共现矩阵进行矩阵分解获得每个单词的向量表示。我们将 GloVe 的嵌入维度设置为 768, 并将窗口大小设置为 15。对于 RoBERTa CLS token 和 Avg. RoBERTa embeddings 方法, 我们使用 RoBERTa 已经在通用领域语料库上完成预训练的检查点来初始化权重, 然后在 10K 报告数据集中进行了继续预训练, 而没有对其进行对比学习的微调。我们将文档切分成模型可以直接处理的文本段进行训练和推理。在推理阶段, 我们计算所有文本段 CLS token 的均值或者进行全局平均池化来生成文档表征。

3.6.2 超参数设置

我们的训练包含预训练和微调两个部分。我们在两张显存容量为 80 GiB 的 A100 GPU 上进行实验。在这两个阶段, 我们均使用了 AdamW^[84] 优化器进行训练。我们在 RoBERTa 的嵌入层、注意力矩阵和逐位置 FFN 中使用了 $p_{drop} = 0.1$ 的 dropout, 同时设置了 0.1 的权重衰减作为正则化项。我们开启了 gradient checkpointing^[85] 以减少显存消耗。

在预训练阶段, 我们将学习率设置为 $1e-4$, 通过线性学习率衰减来进行学习率的调整, 并对最前面的 5% 的步数进行 warmup。此外, 将批处理大小设置为

16, 梯度累加设置为 4, 进行了大约 100K 步的训练。在微调阶段, 我们将学习率设置为 $1e-6$, 采用常数学习率策略。此外, 将批处理大小设置为 64, 对比学习损失中的温度设置为 0.1, 进行了大约 200 步的训练。

在评估指标一中, 我们需要为基于算法的 ICSs 设置相似度阈值, 将其生成的公司表征之间的余弦相似度高于该阈值的公司视为同行公司。为此, 我们首先统计了专家设计的 ICSs 中每个公司平均匹配到的同行公司数量, 具体数据见表 3.2。可以看到, 在我们收集的验证集中, SIC 平均为每个公司找到 33 个同行公司, 而 NAICS 平均找到 49 个同行公司。因此, 我们选择为每个公司平均搜索 50 个同行公司作为基于算法的 ICSs 的相似度阈值。

专家设计的 ICSs	平均的同行公司数量
SIC	33.08
NAICS	48.74

表 3.2 验证集中, SIC 和 NAICS 搜索到的同行公司的平均数量

3.7 实验结果与分析

我们使用了三个自定义指标和 8 个基线来评估我们提出的基于自监督学习的 Financial-ICS 的有效性。表 3.3 展示了 Financial-ICS 在三个指标中与各种行业分类系统的效果对比。从表中, 我们可以观察到, 即使随机选取公司对, 在同一年度的股市数据中仍存在着 0.2798 的 Pearson 相关系数。TNIC 使用的词袋方法展现了其有效性, 在三个指标上均超过了传统表征学习方法 GloVe。这是由于 GloVe 是一种词嵌入学习方法, 而 10K 报告中的平均文档长度达到 20K tokens 以上, 对词嵌入进行平均的操作使其性能不及词袋模型。此外, GloVe 模型仅使用了 768 维的向量来表示整个文档, 而词袋模型总共使用了长达 35K 维度的向量。RoBERTa 模型在仅进行预训练后, 由于其生成的表征受到各向异性的问题, 其三个指标效果都弱于 TNIC。其中, 对词嵌入进行全局平均池化的方案略优于使用 CLS token 的方案。然而, RoBERTa 模型在经过表征学习的微调后, 效果得到明显提升。无论是 SRoBERTa 还是基于 RoBERTa 的 Financial-ICS, 在三个指标上都明显优于 TNIC 和未经过微调的 RoBERTa 模型。这表明, 对预训练的模型进行微调可以明显地改善表征的质量, 增强对应 ICS 找到的同行公司的经济相关性。尽管基于 RoBERTa 的 Financial-ICS 通过使用预训练加对比学习微调的自监督学习方法在这些基于算法的 ICSs 基线中达到了最优的效果, 但距离专家设计的 ICSs, 包括 SIC 和 NAICS, 仍有一定的差距。

方法	股市 Pearson	相似度分数 Spearman		逻辑回归 F1	
		SIC	NAICS	SIC3	NAICS4
Random	0.2798	-0.0001	0.0004	0.0063	0.0115
Bag of words (TNIC)	0.4060	0.2913	0.2988	0.4936	0.5794
Avg. GloVe embeddings	0.3885	0.2904	0.2689	0.0601	0.0888
RoBERTa CLS token	0.3803	0.2802	0.2730	0.0579	0.0720
Avg. RoBERTa embeddings	0.3990	0.2859	0.2865	0.1280	0.1875
SRoBERTa	0.4177	0.3355	0.3559	0.5488	0.6333
SIC	0.4261	-	-	-	-
NAICS	0.4473	-	-	-	-
Financial-ICS (RoBERTa-SimCSE)	0.4197	0.3367	0.3441	0.6030	0.6730

表 3.3 Financial-ICS (RoBERTa-SimCSE) 与 8 种不同的行业分类系统的效果对比

数据增强方法	股市 Pearson	相似度分数 Spearman		逻辑回归 F1	
		SIC	NAICS	SIC3	NAICS4
无 (dropout)	0.4197	0.3367	0.3441	0.6030	0.6730
删除 1 个词	0.4198	0.3335	0.3416	0.5936	0.6724
删除 5% 的词	0.4124	0.3323	0.3364	0.5933	0.6694
不使用 dropout	0.4120	0.3315	0.3289	0.5901	0.6523
Masked LM 15%	0.4172	0.3312	0.3344	0.5896	0.6640
扰乱位置顺序	0.4080	0.3121	0.3154	0.5428	0.6208

表 3.4 在对比学习中使用不同数据增强方式的效果对比

3.8 数据增强方法的探索

数据增强方法在对比学习中扮演着重要角色。这一节中，我们将研究使用不同的数据增强方法对 RoBERTa 模型进行对比学习微调，以观察这对其构建的 Financial-ICS 性能的影响。我们准备了 6 种不同的数据增强方法来生成对比学习中的正样本对，包括 dropout、删除 1 个词、删除 5% 的词、不使用 dropout、对 15% 的 tokens 进行掩码以及扰乱位置顺序。其中，我们采用 $p_{drop} = 0.1$ 作为 dropout 的实验默认配置，扰乱位置顺序通过直接对位置嵌入的顺序进行扰乱来实现。在实验中，我们都使用全局平均池化作为文档表征的聚合函数。

从表 3.4 中可以看出，使用 dropout 即不使用任何复杂数据增强的表现最为出色。值得注意的是，扰乱位置顺序对性能产生了明显下降，由于 RoBERTa 每次的输入长度基本都是 512 tokens，即文档的分段长度，扰乱位置顺序对文本语义造成了破坏，从而导致效果不佳。

3.9 本章小结

在本章中，我们提出了自监督学习的 Financial-ICS，采用预训练加对比学习微调的方式来训练其中的表征模型。我们使用 RoBERTa 作为实验的模型，以验证该训练框架的有效性。由于 10K 报告数据集中的最大文本长度达到 100K tokens 以上，我们对长文档进行了分段处理。我们分析了 RoBERTa 模型在对比

学习微调前后生成的文档表征的分布,验证了对比学习可以有效缓解文档表征中的各向异性问题。此外,我们提出了三个自定义的指标,用于评估自监督学习的 Financial-ICS 搜索到的同行公司的经济相关性、其与专家设计的行业结构的一致性以及表征模型生成的文档表征质量。实验结果表明,使用自监督学习框架训练的 RoBERTa 模型,其构建的 Financial-ICS 效果明显优于仅经过预训练的 RoBERTa 模型构建的 ICS,并且超过了目前最先进的基于算法的 ICS, TNIC。但是与专家设计的 ICSs 相比,仍存在一定的差距。另外,我们还对对比学习期间采用的数据增强方法进行了探索,并发现使用 dropout 进行数据增强效果最佳。

第4章 基于高效 Transformer 的行业分类系统研究

4.1 引言

在第3章中,我们介绍了基于 RoBERTa 的 Financial-ICS 的自监督训练框架,并设计了三个指标对其进行评估。尽管基于 RoBERTa 的 Financial-ICS 超过了当前最先进的基于算法的 ICS, TNIC, 但是与专家设计的 ICSs 相比,仍存在较大差距。由于 10K 报告的最大文本长度达到 100K tokens 以上,使用 RoBERTa 模型进行分段表征再求均值并不是最佳的实现方法。目前已有的高效 Transformer 受到内存瓶颈或者位置编码的限制,无法直接处理如此长的文档。在本章中,我们提出了一种新的高效 Transformer 模型, LongBERT。通过结合移位块注意力和全局注意力, LongBERT 将自注意力机制的计算和内存复杂度降低至 $O(n)$, 能够处理长达 131K tokens 的文档。另外,为了解决长文档情况下对比学习的内存瓶颈问题,我们设计了内存优化的对比学习方法,以便能够使用长文档对 LongBERT 进行对比学习微调。

4.2 LongBERT

鉴于 10K 报告中的最大文本长度超过 100K tokens, 而 Transformer 中自注意力机制的计算和内存复杂度是序列长度的二次方。为解决 Transformer 在处理长文本时的限制,我们设计了一种更高效的,且具有双向视野域的高效 Transformer 模型, LongBERT。LongBERT 最长可以处理 131K tokens 的长文档,从而可以直接处理 10K 报告数据集中的任意文档。

我们以 RoBERTa 的模型结构作为基础,并对其自注意力机制和位置编码方式进行了改进。我们通过结合移位块注意力和全局注意力来替换传统的自注意力机制,并引入旋转位置嵌入 (rotary position embedding, RoPE)^[86] 作为 LongBERT 的位置编码方式来取代原模型可学习的绝对位置嵌入。

首先,我们对 RoBERTa 的注意力矩阵进行分析。我们随机选择了 100 个 10K 报告文档,并使用 RoBERTa 模型进行处理,收集模型各层 softmax 归一化后的注意力矩阵,然后进行均值运算。我们取数值的自然对数来优化图像的可视化结果,如图4.1所示。我们发现自注意力机制中的询问 (query) 更有可能与邻居位置的键 (key) 具有更高的注意力权重,这也体现了自然语言任务中的信息局部性原理。在一段文本中,某个 token 的含义通常由其周围 tokens 组成的上下文所决定,因此该 token 与邻居的 tokens 通常具有更紧密的联系,需要更多的关注。基于这一分析结果,我们引入了局部注意力。

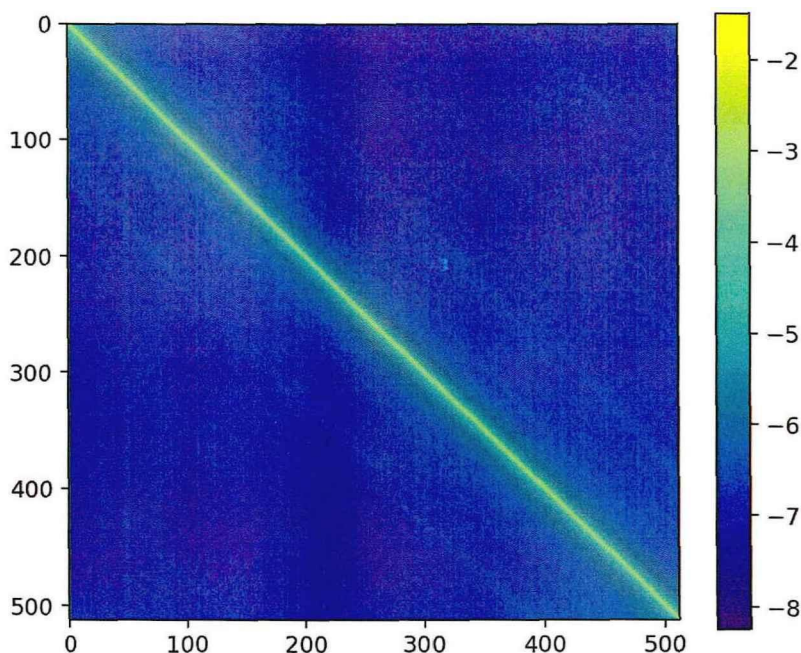


图 4.1 RoBERTa 的注意力矩阵分析

目前主要有三类局部注意力的方式，分别是块注意力（block attention）、窗口注意力（window attention）和扩张注意力（dilated attention）^[31,66]。块注意力通过将输入序列分段成不重叠的文本段，每个块中的所有查询只能关注当前局部块中的键值对，实现 $O(n)$ 的复杂度。但是缺点是文本块之间无法进行信息的流动。窗口注意力则规定每个询问只能关注其邻居若干节点的键值对。扩张注意力则借鉴扩张卷积神经网络^[87]的思想，在窗口注意力思想的基础上，使用带间隔扩张的窗口来增加窗口注意力的感受野，而不增加复杂度。窗口注意力和扩张注意力的缺点是，它们需要使用高度优化的 CUDA 内核^[31]的方式来进行实现，如果使用原生 torch 实现，则会因为其中复杂的张量计算而降低计算效率。

我们选择使用移位块注意力（shifted block attention）来实现局部注意力，因为这种注意力模式可以使用原生 torch 高效实现，而不需要自定义 CUDA 内核。首先，我们引入简单的块注意力。我们假设文档长度为 n ，并将文档划分成每段长度为 w 的文本段。每个 token 只能注意本文段内的 tokens。通过引入块注意力，我们降低了自注意力的复杂度为 $O(wn)$ ，由于 w 是一个常数，所以其复杂度为 $O(n)$ 。

虽然这种方式降低了传统自注意力机制的复杂度，但是由于不同段之间缺乏信息交流，导致其在长文本训练中表现不佳。因此我们引入了移位块注意力。我们设计了两种块注意力的模式。第一种是之前提到的简单的块注意力模式，第二种是通过将块注意力的位置向前平移了 $w/2$ 个 tokens，即将文档分段的位置向前进行平移。两种注意力模式中的部分 tokens 在文本段之间出现重叠，从而实现

不同段之间信息的流动。

我们让多头注意力机制中的不同的注意力头分别使用这两种模式。一半的注意力头使用第一种模式，另一半使用第二种模式。在图4.2中，我们展示了 LongBert 的自注意力结构图，其中黄色的部分代表使用第一种模式的块注意力，绿色部分代表使用第二种模式的块注意力。在该图中，文本块的长度 $w = 4$ 。我们将两种模式的块注意力整合起来称之为移位块注意力。通过使用移位块注意力，我们可以在保持自注意力 $O(n)$ 复杂度的同时，增强模型局部注意力的覆盖范围。假设 LongBert 总共包含 L 个注意力层，那么局部注意力的视野将扩展到 $O(Lw)$ 。

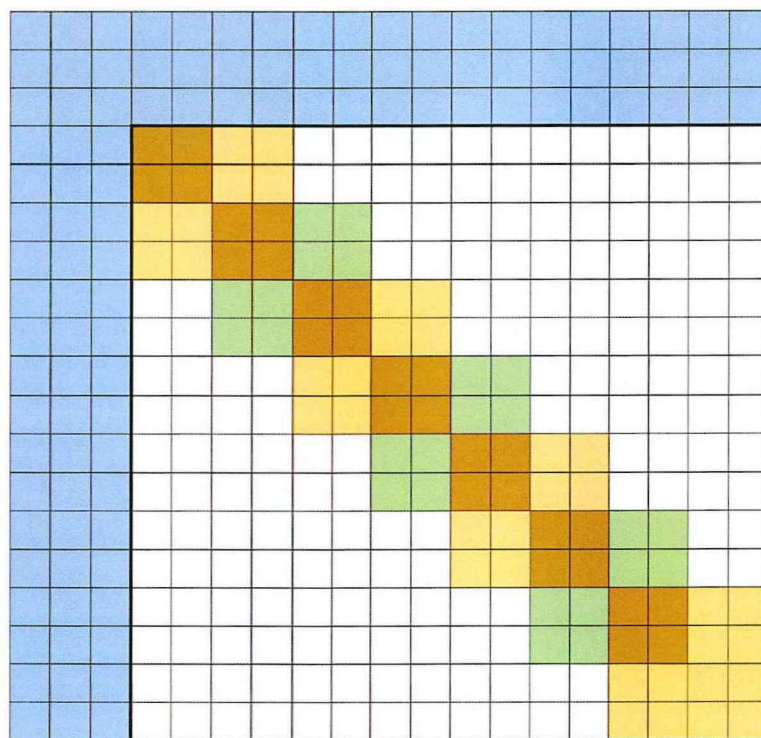


图 4.2 LongBert 的自注意力机制

尽管移位块注意力显著地增加了模型的局部注意力的视野范围，但是在 10K 报告数据集中，最长文档的 tokens 数达到 100K 以上。仅使用局部注意力的方式，我们仍无法实现对整个文档的全局视野。为了解决这个问题，我们引入了全局注意力，使模型学习长依赖的路径长度降至 $O(1)$ 。我们使用全局 tokens 的方式来实现全局注意力，希望它可以作为文本段之间进行信息流动的桥梁。全局 tokens 允许对所有 tokens 进行关注，同时，所有 tokens 也可以对全局 tokens 进行关注。我们采用了外部拓展的全局 tokens 的方案，即通过创建 g 个额外的可训练 tokens，而不是通过直接使用文本段中的 tokens 作为全局 tokens，这种方案可以使得 LongBert 具有更强的可拓展性，而不受到具体任务的影响。

然后，我们对提出的自注意力机制进行了计算和内存复杂度的分析。移位块

注意力的复杂度为 $O(wn)$, 因为对一部分注意力头的块注意力机制进行平移并不会改变其复杂度。全局注意力复杂度为 $O(2gn)$ 。因此, 总的复杂度为 $O(wn+2gn)$ 。在实验中, 我们将 w 设置为 512, g 设置为 16。因此复杂度可以简化为 $O(n)$ 。

我们使用批处理的方式来高效实现 LongBERT 的自注意力机制。LongBERT 中的 tokens 可以分为两个部分, 一部分是需要执行移位块注意力的内部 tokens, 另一部分是拓展的外部全局 tokens。我们分别对这两个部分的 tokens 进行处理。首先, 我们需要将内部 tokens 的长度填充到 w 的整数倍, 然后对采用第二种注意模式的注意力头进行旋转。具体地, 我们将这些注意力头的询问、键、值和掩码向后旋转 $w/2$ 个位置, 这样序列的最后的 $w/2$ 个 tokens 会旋转到序列最前面。我们对旋转到序列最前面的 tokens 进行掩码处理, 因为我们认为将句子开头和结尾混合在一起进行注意力计算, 会干扰局部注意力所需要学习的局部模式。全局的信息流动应该是全局 tokens 需要关心的。旋转完成后, 我们对序列进行分段, 并将拓展的全局 tokens 拼接在各个段上, 使得所有的内部 tokens 都可以关注到全局 tokens。通过这种方法, 我们可以将文本段数作为批处理的维度进行自注意力, 从而进行高效的并行运算。而全局 tokens 对所有内部 tokens 的关注可以直接通过完全注意力简单实现。LongBERT 的自注意力机制可以使用原生 torch 高效实现, 而不需要自定义 CUDA 内核, 这确保了该注意力机制的灵活性和可拓展性。

除了对自注意力机制进行了修改外, 我们还采用了旋转位置嵌入^[86]来替代原模型采用的可学习的绝对位置编码方式。目前主要有两种位置编码方式, 即绝对位置编码和相对位置编码。绝对位置编码为每个位置分配一个向量, 但每个位置的嵌入都是独立于其他位置的, 无法直接提供相对位置的信息。而相对位置编码使用 tokens 之间的相对位置关系和距离进行位置建模。因为我们处理的是长文本, 创建长度为 100K tokens 以上的可学习绝对位置编码会面临参数数量过大和收敛速度缓慢的问题, 而引入相对位置编码又会遇到相对位置矩阵的内存瓶颈限制。旋转位置嵌入结合了这两种编码方式的优点, 通过使用旋转矩阵对绝对位置信息进行编码, 同时能够在自注意力公式的向量内积中显式引入相对位置的依赖性。已经有实验表明, 该位置嵌入在长文本任务中具有更好的性能^[86], 且对序列长度具有更大的灵活性。

综上所述, LongBERT 通过同时引入移位块注意力和全局注意力实现了复杂度和长依赖学习能力的平衡。移位块注意力可以有效地利用自然语言中的信息局部性原理, 并实现 $O(n)$ 的计算和内存复杂度。其移位的设计扩展了局部注意力的感受野。而全局注意力在此基础上补充了长依赖学习能力。我们可以利用批处理技巧对其自注意力机制进行高效实现。此外, 我们还引入了旋转位置嵌入的方式对长文本进行编码, 增强 LongBERT 在处理长文本时的能力和灵活性, 并加

快了收敛速度。

4.3 自监督训练框架

我们对 LongBERT 的训练采用了与 RoBERTa 类似的自监督训练框架，即预训练加对比学习微调的范式，但存在一些差异。为了避免从头开始对 LongBERT 进行训练，我们使用了 RoBERTa 在通用领域语料库完成预训练的检查点权重对 LongBERT 进行初始化。由于 LongBERT 是 RoBERTa 的改进版本，主要升级了其中的注意力机制和位置编码方式，因此我们可以复用大部分的参数权重，这显著提升了模型的收敛速度。接着，我们对 LongBERT 在 10K 报告数据集中进行继续预训练，以增强模型对金融领域文本的理解能力，并让模型更好地适应新的注意力结构和位置编码。由于 LongBERT 最长可以处理 131K tokens 的长文本，即可以对 10K 报告数据集中的整个文档进行推理，因此我们不需要对文档进行分段处理。最后，我们对 LongBERT 进行对比学习微调。为了缓解长文本处理和微调期间增大负样本数量之间的矛盾，我们没有使用对比学习任务中常用的 InfoNCE 损失^[79]，而是设计了一种内存优化的长文本对比学习方法，用以微调 LongBERT。

4.4 内存优化的对比学习方法

对比学习是自监督训练框架中的重要环节。我们收集的 10K 报告数据集的平均长度在 20K tokens 以上，虽然 LongBERT 将 Transformer 结构中的自注意力机制的复杂度从 $O(n^2)$ 降低为 $O(n)$ ，但仍会受到内存瓶颈的限制，导致训练时的批处理大小过小。在预训练阶段，这并不是问题，因为我们可以通过梯度累加的技巧来间接增大总的批处理大小，从而提高训练的稳定性。然而，在对比学习微调阶段，由于需要引入正负样本来进行损失的计算，较小的批处理大小会导致负样本数量稀缺，甚至无法引入负样本而导致训练失败。例如，如果总的批处理大小只能达到 2，那么我们就无法创建负样本进行对比学习损失的计算，也就无法对模型进行优化。并且对比学习中负样本的数量往往对提升表征质量有帮助^[58]。因此，我们设计了内存优化的对比学习方法，使我们能够使用长文档对 LongBERT 进行对比学习微调。

我们采用了 dropout 对同一文档进行两次编码来产生正样本对，因为这种增强方法在第3.8节数据增强方法的探索中取得了最好的效果。此外，我们使用全局平均池化来生成文档向量。我们设计了一种新的对比学习损失：基于原型 (prototype) 的对比学习损失，可以从分类任务的角度来思考对比学习。我们设计

的对比学习损失如下：

$$\begin{aligned}
 p_i &= (h_{2i} + h_{2i+1})/2 \\
 label_i &= \lfloor i/2 \rfloor \\
 Logits &= \text{pairwise_sim}(H, P)/\tau \\
 loss &= \text{cross_entropy}(Logits, Labels)
 \end{aligned} \tag{4.1}$$

其中 $h_{2i}, h_{2i+1} \in \mathbb{R}^{d_m}$ 是正样本对, d_m 是模型产生文档表征的维度。由于正样本的表征是由同一个文档使用不同的 dropout 掩码生成的, 我们计算每个正样本对的均值来创建它们对应文档的原型 $p_i \in \mathbb{R}^{d_m}$, 并赋予这些原型各自的类别标签, 代表这些原型来自第 i 个文档。原型的个数和批处理中文档的数量相等, 也与类别标签的个数相等。 $H \in \mathbb{R}^{2n \times d_m}$, $P \in \mathbb{R}^{n \times d_m}$ 是由所有的 h, p 向量打包成的矩阵, 其中 n 为文档的数量。 pairwise_sim 是逐对的相似度函数, 这里我们选择余弦相似度进行计算。由于 Financial-ICS 在查找同行公司时, 使用文档表征的余弦相似度进行排序, 所以在损失函数中直接对其进行优化是合理的。 τ 为温度 (temperature) 超参数。整个对比学习任务可以被简单的定义为分类任务, 即将文档表征 h 分类到对应的原型中, 旨在增大样本表征与对应原型之间的余弦相似度, 降低与其他原型的余弦相似度。我们可以使用交叉熵损失来对其进行优化。交叉熵损失的输入 $Logits \in \mathbb{R}^{2n \times n}$, $Labels \in \mathbb{R}^{2n}$ 是由 label 构成的向量。基于原型的对比学习函数损失的优点是, 易于理解且可以处理每个正样本组中样本数量不同或者超过两个的情况。

除了对比学习损失的设计, 我们还对其进行了内存消耗方面的优化。为了减少内存占用, 我们仅对其中的 s 对正样本和它们的原型进行带梯度的损失计算, 而其他的样本和原型则在无梯度的上下文环境中进行计算, 仅作为负样本的形式提供表征信息。我们可以将这些不需要计算梯度的样本分开计算, 并以更小批次进行处理, 从而显著降低内存消耗。在我们的实验中, 由于内存的限制, 我们将 s 设置为 1, 但是总的批处理大小可以被设置为 16, 即负样本的数量为 14。通过这种方法, 我们增大了每个批次中对比学习负样本的数量。由于对比学习微调的收敛很快, 所以这样的设计并不会对微调时长造成较大的影响。

4.5 实验细节

4.5.1 基线

在第3.6.1节的基础上, 我们引入了两个新的基线, 总共选择了 11 个基线来评估我们提出的基于 LongBERT 的 Financial-ICS 的有效性。其中, SimCSE-RoBERTa 即为我们在3章中提出的自监督学习的 Financial-ICS。而 SimCSE-

Longformer^[31,52]是我们新增加的基线。与 SimCSE-RoBERTa 类似, SimCSE-Longformer 也使用 dropout 作为对比学习的数据增强方法, 并使用 SimCSE 中的 InfoNCE 损失^[79]进行优化。Longformer 是当前热门的高效 Transformer, 其计算和内存复杂度为 $O(n)$, 最大能处理 4096 tokens 的长文本, 因此我们仍需要对文档进行分段。在实验中, 我们使用 512 tokens 的窗口注意力, 并将 CLS token 作为全局 token 进行全局注意力。Avg. LongBERT embeddings 是只经过 10K 报告继续预训练的 LongBERT 模型, 而没有对其进行对比学习微调。我们在输出层使用全局平均池化来生成文档表征。LongBERT 可以直接处理完整的 10K 报告文档, 而无需进行分段处理。基于 LongBERT 的 Financial-ICS 是我们提出的新的基于高效 Transformer 的行业分类系统。该系统利用 LongBERT 来提取文档中的公司经济表征, 采用预训练加对比学习微调的自监督训练框架, 并利用内存优化的对比学习方法进行微调。

4.5.2 超参数设置

我们的训练包含预训练和微调两个部分。我们在两张显存容量为 80 GiB 的 A100 GPU 上进行实验。在这两个阶段, 我们均使用了 AdamW^[84] 优化器进行训练。我们设置了 LongBERT 的移位块注意力中的文本段长度 $w = 512$, 外部拓展的全局 tokens 数 $g = 16$, 并在 LongBERT 的嵌入层、注意力矩阵和逐位置 FFN 中使用了 $p_{drop} = 0.1$ 的 dropout, 同时设置了 0.1 的权重衰减作为正则化项。我们开启了 gradient checkpointing^[85] 以减少显存消耗。

在预训练阶段, 我们将学习率设置为 $1e-4$, 通过线性学习率衰减来进行学习率的调整, 并对最前面的 5% 的步数进行 warmup。为了节约显存使用, 我们使用了 deepspeed ZeRO2^[88] 的分布式训练方式。此外, 将批处理大小设置为 1, 梯度累加设置为 16, 进行了大约 20K 步的训练。在微调阶段, 我们将学习率设置为 $1e-6$, 采用常数学习率策略。此外, 将批处理大小设置为 16, 其中包括 2 个样本进行梯度的计算, 而另外的 14 个样本在无梯度环境下进行计算。我们将对比学习损失中的温度设置为 1, 进行了大约 300 步的训练。

我们沿用了第3.6.2节中对于相似度阈值的选择, 将为每个公司平均选择 50 个同行公司作为所有基于算法的 ICSs 的相似度阈值。

4.6 实验结果与分析

我们使用了三个自定义指标和 11 个基线来评估我们提出的基于 LongBERT 的 Financial-ICS 的有效性。其中评估指标一可以对基于算法的 ICSs 和专家设计的 ICSs 进行评估, 而指标二、三由于使用了专家设计的 ICSs 作为人工标签, 因

方法	股市 Pearson	相似度分数 Spearman		逻辑回归 F1	
		SIC	NAICS	SIC3	NAICS4
Random	0.2798	-0.0001	0.0004	0.0063	0.0115
Bag of words (TNIC)	0.4060	0.2913	0.2988	0.4936	0.5794
Avg. GloVe embeddings	0.3885	0.2904	0.2689	0.0601	0.0888
RoBERTa CLS token	0.3803	0.2802	0.2730	0.0579	0.0720
Avg. RoBERTa embeddings	0.3990	0.2859	0.2865	0.1280	0.1875
SRoBERTa	0.4177	0.3355	0.3559	0.5488	0.6333
SimCSE-RoBERTa	0.4197	0.3367	0.3441	0.6030	0.6730
SimCSE-Longformer	0.4235	0.3998	0.3808	0.6244	0.6982
SIC	0.4261	-	-	-	-
NAICS	0.4473	-	-	-	-
Avg. LongBERT embeddings	0.4154	0.3189	0.2939	0.2186	0.3403
Financial-ICS (LongBERT-based)	0.4289	0.4198	0.4194	0.6334	0.7086

表 4.1 Financial-ICS 与 11 种不同的行业分类系统的效果对比

此只能对基于算法的 ICSs 进行评估。三个指标分别反映了行业分类系统找到的同行公司与焦点公司之间的经济相关性、基于算法的 ICSs 搜索到的同行公司与专家设计的行业结构之间的一致性，以及基于算法的 ICSs 中产生的公司经济表征的质量。表4.1展示了 Financial-ICS 与各种不同的行业分类系统在三个指标上的效果对比。接下来，我们将对新加入的方法的实验结果进行描述和分析。

在第3.7节中，我们已经提到，SimCSE-RoBERTa 的方法超过了当前最先进的基于算法的 ICS，TNIC，但是与专家设计的 ICSs，如 SIC、NAICS 相比，仍存在较大差距。SimCSE-Longformer 在 SimCSE-RoBERTa 的基础上对表征模型进行了改进，将模型可以直接处理的文本长度从 512 tokens 提升至 4096 tokens。从实验结果中可以看出，SimCSE-Longformer 在三个指标中的效果都要优于 SimCSE-RoBERTa。Avg. LongBERT embeddings 由于只经过了预训练，其生成的表征存在各向异性问题，导致其性能与经过微调的 Financial-ICS 存在较大差距，但是其在三个指标上的效果依旧超过 Avg RoBERTa embeddings，在部分指标上优于 TNIC。我们提出的基于 LongBERT 的 Financial-ICS，由于可以直接对 10K 报告文档进行处理，优于所有基于最先进算法的 ICSs 和专家设计的 SIC，尽管距离 NAICS 仍有差距，但是在4.8.1节相似度阈值的实验中，当相似度阈值提升至每个公司平均搜索 20 个同行公司时，其性能超过了 NAICS。

4.7 消融实验

我们对训练 Financial-ICS 中引入的三个方法进行消融实验，分别是基于原型的对比学习损失、内存优化的对比学习方法和 LongBERT 模型。由于内存限制，我们无法在无内存优化的对比学习方法下使用 LongBERT 进行训练，因此，我们逐步在 RoBERTa 下引入这些技巧，以观察它们对 Financial-ICS 效果的影响。

从表4.2中可以看出，基于原型的对比学习损失能够达到与 SimCSE^[52]中的

InfoNCE 损失类似的效果。这种方法可以从分类的角度来看待对比学习任务,并且可以处理每个正样本组中样本数量不同或超过两个的情况。内存优化的对比学习方法通过对一部分样本对在无梯度的环境下进行推理,仅作为负样本提供表征信息,从而降低内存消耗。我们在批处理大小为 16 的设置下进行实验,并仅对其中的 1 组样本对进行梯度计算,发现实验结果只会发生略微下降,但却可以显著降低内存消耗,从而能够引入 LongBERT 进行对比学习微调。而 LongBERT 的引入明显提高了 Financial-ICS 的效果,特别是指标二的 Spearman 系数有了大幅增长。因此,增大模型能够处理的最大文本长度可以显著提升长文档表征学习的性能。

方法	股市 Pearson	相似度分数 Spearman		逻辑回归 F1	
		SIC	NAICS	SIC3	NAICS4
RoBERTa	0.4197	0.3367	0.3441	0.6030	0.6730
+ 基于原型的对比学习损失	0.4194	0.3358	0.3454	0.6082	0.6701
+ 内存优化的对比学习方法	0.4195	0.3342	0.3428	0.6012	0.6710
+ LongBERT	0.4289	0.4198	0.4194	0.6334	0.7086

表 4.2 基于 LongBERT 的 Financial-ICS 的消融实验

4.8 参数敏感性

我们对 Financial-ICS 进行了参数敏感性的实验,包括搜索同行公司的相似度阈值、批处理大小、生成文档表征的聚合方法、对比学习微调阶段的 dropout 与温度,以及 LongBERT 能处理的最大上下文长度。

4.8.1 相似度阈值

在我们自定义的评估指标一中,基于算法的 ICSs 需要在搜索同行公司时对相似度阈值进行设置,以确定哪些公司被认为是同行公司。实验中,我们默认设置每个公司平均搜索 50 个同行公司作为相似度阈值。在这里,我们进行了实验,以探究 Financial-ICS 使用不同相似度阈值对指标一的影响。

从表 4.3 可以看出,设定更高的相似度阈值可以找到经济相关性更高的同行公司。当相似度阈值被设置为每个公司平均搜索 20 个同行公司时,其性能超过了 NAICS。

平均搜索的同行公司数量	股市 Pearson
10	0.4550
20	0.4478
50	0.4289

表 4.3 Financial-ICS 使用不同相似度阈值对评估指标一的影响

4.8.2 批处理大小

Financial-ICS 采用了内存优化的对比学习方法来增加对比学习中负样本的数量。表4.4展示了使用不同的批处理大小对 Financial-ICS 效果的影响。在实验中,若要使用对比学习的方法进行微调,需要保证批处理的大小大于等于4才能引入正负样本,以确保实验成功运行。结果表明,在批处理大小较小时,增大批处理大小可以显著提升 Financial-ICS 的效果,不过当批处理大小超过16后,提升效益不再明显。

批处理大小	股市 Pearson	相似度分数 Spearman		逻辑回归 F1	
		SIC	NAICS	SIC3	NAICS4
4	0.4245	0.3989	0.3965	0.6287	0.6995
8	0.4271	0.4138	0.4102	0.6321	0.7056
16	0.4289	0.4198	0.4194	0.6334	0.7086
32	0.4284	0.4238	0.4187	0.6359	0.7081

表 4.4 使用不同的批处理大小对 Financial-ICS 效果的影响

4.8.3 聚合方法

Financial-ICS 需要将 10K 报告文本转换成文档表征的形式以进行同行公司搜索。因此,我们需要在 LongBERT 的输出层对词嵌入进行聚合来生成文档表征。我们探索了在对比学习微调和评估中使用不同的聚合方法对实验结果的影响,包括使用全局 tokens 的均值、不含全局 tokens 的全局平均池化以及所有 tokens 的全局平均池化。从表4.5中,我们发现在训练和评估中,使用全局平均池化的效果更佳。

聚合方法	股市 Pearson	相似度分数 Spearman		逻辑回归 F1	
		SIC	NAICS	SIC3	NAICS4
Global tokens avg.	0.4155	0.3364	0.3453	0.5843	0.6735
First-last avg. (不含全局 tokens)	0.4256	0.4139	0.4084	0.6302	0.7010
First-last avg.	0.4289	0.4198	0.4194	0.6334	0.7086

表 4.5 使用不同的聚合方法生成文档表征对 Financial-ICS 效果的影响

4.8.4 dropout 与温度

我们使用 dropout 的方式来产生对比学习中的正样本对,并且在对比学习损失中使用温度 τ 来控制样本间相似度的敏感性。我们研究了在微调阶段使用不同 p_{drop} 和 τ 对训练效果的影响。如表4.6和表4.7所示,选择 $p_{drop} = 0.1$, $\tau = 1$ 的效果最佳。

4.8.5 最大上下文长度

引入更长的上下文长度来进行长文档的表征学习通常可以带来更好的效果。因此,我们对 LongBERT 在预训练、微调和推理阶段能处理的最大 tokens 数进

p_{drop}	股市 Pearson	相似度分数 Spearman		逻辑回归 F1	
		SIC	NAICS	SIC3	NAICS4
0	0.4206	0.3925	0.3783	0.6239	0.7054
0.05	0.4203	0.3918	0.3924	0.6330	0.7066
0.1	0.4289	0.4198	0.4194	0.6334	0.7086
0.2	0.4254	0.4072	0.3934	0.6310	0.7044
0.3	0.4204	0.4033	0.3997	0.6291	0.7016
0.5	0.4174	0.3683	0.3500	0.5387	0.6383

表 4.6 使用不同的 p_{drop} 对 Financial-ICS 效果的影响

τ	股市 Pearson	相似度分数 Spearman		逻辑回归 F1	
		SIC	NAICS	SIC3	NAICS4
0.01	0.4263	0.3786	0.3741	0.6245	0.6861
0.1	0.4266	0.4145	0.4065	0.6294	0.7072
1	0.4289	0.4198	0.4194	0.6334	0.7086
10	0.4174	0.4074	0.3924	0.6293	0.7040

表 4.7 使用不同的温度 τ 对 Financial-ICS 效果的影响

行了限制，以探究模型能处理的最大上下文长度对 Financial-ICS 效果的影响。从表4.8中可以看出，如我们预想的一样，使用更长的上下文长度有利于提升 Financial-ICS 的性能。

Max Length	股市 Pearson	相似度分数 Spearman		逻辑回归 F1	
		SIC	NAICS	SIC3	NAICS4
512	0.4189	0.3649	0.3408	0.6088	0.6706
4K	0.4218	0.3858	0.3754	0.6211	0.6858
20K	0.4226	0.4078	0.3895	0.6297	0.7013
131K	0.4289	0.4198	0.4194	0.6334	0.7086

表 4.8 LongBERT 能处理的最大上下文长度对 Financial-ICS 效果的影响

4.9 推理速度与内存消耗

LongBERT 通过将移位块注意力和全局注意力相结合，将时间和内存复杂度降低到了 $O(n)$ ，这种注意力模式可以通过批处理技巧进行高效的实现。在这一节中，我们对 LongBERT 在真实场景下处理长文本的时间和内存效率进行了实验。我们的实验环境是 80 GiB 显存的 A100 GPU。

首先我们比较了 RoBERTa、Longformer 和 LongBERT 在 10K 报告的 demo 集中使用不同的批处理大小的推理速度。由于 demo 集的平均长度为 23K tokens，对于 RoBERTa 和 Longformer，我们在推理前将文档分段成模型能处理的最大长度，分别为 512 tokens 和 4096 tokens。从表4.9中可以看出，当批处理大小为 1 时，RoBERTa 的推理速度非常慢，这是由于批处理大小太小而无法充分利用 GPU 的

并行运算能力导致。当批处理大小大于等于 4 时，RoBERTa 达到了三个模型中最优的推理速度。LongBERT 在批处理大小为 1 时就展现了其出色的处理速度。尽管 Longformer 和 LongBERT 的计算复杂度都为 $O(n)$ ，但是由于 LongBERT 在注意力实现上的简洁性，其推理速度是 Longformer 的 1.87 倍。

Batch Size	RoBERTa	Longformer	LongBERT
1	2.99	3.03	7.61
4	7.74	3.74	7.49
16	9.83	4.07	-
64	10.37	4.07	-
256	10.73	-	-

表 4.9 RoBERTa、Longformer 和 LongBERT 在不同批处理大小下的推理速度

然后，我们比较了使用完全注意力（full attention）和 LongBERT 中的注意力机制推理不同序列长度文本的内存消耗。如图 4.3 所示，完全注意力由于其 $O(n^2)$ 的内存复杂度，在序列长度超过 512 时，其内存消耗呈现急剧上升的趋势。当序列长度达到 4K 时，就需要 25 GiB 以上的显存资源进行推理。而 LongBERT 由于其自注意力 $O(n)$ 的内存复杂度，可以从图中明显地看到，其内存消耗随序列的长度增加呈现线性增长。当推理 131K tokens 的长文本时，只需要 14.4 GiB 的显存资源，这通常是可以接受的。

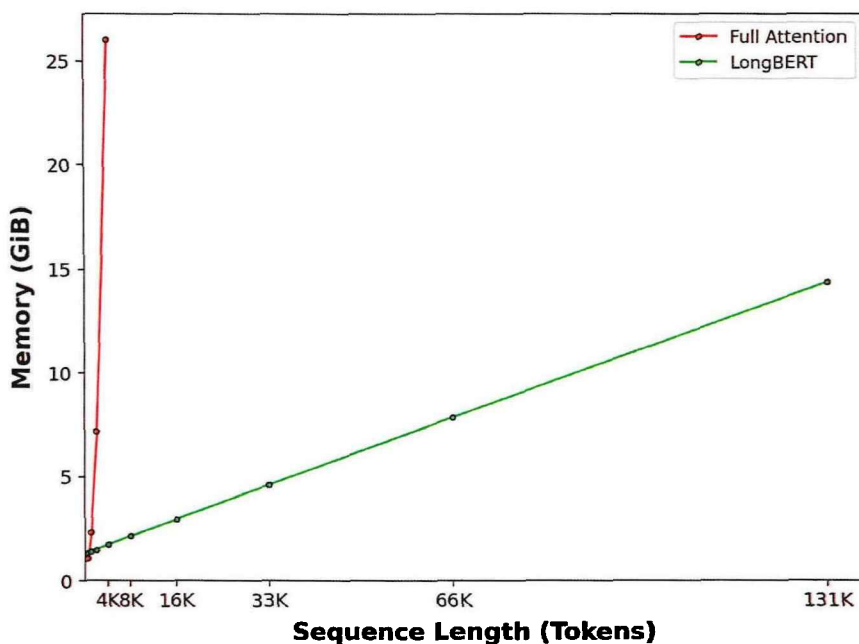


图 4.3 比较使用完全注意力和 LongBERT 中的注意力机制推理不同序列长度时的内存消耗

4.10 Financial-ICS 演示

Financial-ICS 已经在三个自定义的指标中展现了其相对于其他行业分类系统的先进性。此外，我们还提供了 Financial-ICS 的演示，对结果进行直观定性的分析。

图4.4中，我们展示了 Financial-ICS 针对微软和 Visa 公司找到的同行公司的示例。我们使用 10K 报告中的 demo 集来生成所有公司的表征。在表格中，第一行代表焦点公司，其下是按余弦相似度降序排序的搜索到的同行公司。从图中可以看出，Financial-ICS 找到的同行公司通常与相应的焦点公司具有相同的 SIC 编码。这表明，虽然 LongBERT 的训练没有使用 SIC 编码作为监督信号，但 Financial-ICS 与专家设计的 ICS 对公司经济相关性的判断是一致的，这也间接说明了 10K 报告中的产品与业务描述部分是基于算法的 ICSs 的理想输入。与专家设计的 ICSs 不同的是，Financial-ICS 保留了相似度分数和同行公司的顺序信息，这些信息能够为金融研究中对同行公司的选择提供更细粒度的控制，例如可以控制选择与焦点公司最相似的公司来构建研究的基线或者作为对照等。此外，我们还注意到了一些例外情况，其中同行公司与焦点公司拥有不同的 SIC 编码，比如微软和 Zendesk 公司、Visa 和 Euronet worldwide 公司。我们认为这是由于公司业务多样性所导致的。基于算法的 ICSs 可以提供为每个公司查找其独特的同行公司的能力，而不受分层的行业结构的限制。最后，Financial-ICS 具有很好的可拓展性。如果我们需要引入新的公司的 10K 报告，我们只需要使用 LongBERT 对其报告进行处理，产生公司的经济表征，然后将其加入行业分类系统中的表征列表中即可。如果需要更新已有公司的 10K 报告，也只需要更新对应公司的表征向量，这种可拓展性使得 Financial-ICS 可以自动化地完成对行业分类系统更新迭代的大部分过程。

总之，Financial-ICS 有效地解决了专家设计的 ICSs 中存在的限制，且找到的同行公司具有很好的经济相关性，与专家设计的 ICSs 具有可比较的能力。

4.11 本章小结

在本章中，我们提出了 LongBERT，通过结合移位块注意力和全局注意力，成功将自注意力机制的计算和内存复杂度从 $O(n^2)$ 降低至 $O(n)$ ，这使得 LongBERT 能够直接处理 10K 报告数据集中的任意文档，而这是现有高效 Transformer 无法做到的。我们可以通过使用批处理技巧对其自注意力机制进行高效实现，使其在真实场景下的推理速度比同为 $O(n)$ 复杂度的 Longformer 快 1.87 倍。此外，我们提出了内存优化的对比学习方法。该方法仅对批处理中的少量样本对和原型进行梯度计算，而其他样本对和原型则在无梯度环境下进行计算，仅作为负样本的

形式提供表征信息。实验表明,内存优化的对比学习方法仅会对模型效果产生略微下降,但却可以显著降低内存消耗,从而能够使用长文档对 LongBERT 进行对比学习微调。我们提出了基于 LongBERT 的 Financial-ICS,并对其性能进行了评估。实验结果表明,基于 LongBERT 的 Financial-ICS 在性能上超过了一系列基于最先进算法的 ICSs,包括 RoBERTa-SimCSE 和 Longformer-SimCSE,并且在默认相似度阈值设置下超过了专家设计的 SIC。当我们将相似度阈值提升至每个公司平均搜索 20 个同行公司时,其性能超过了 NAICS。

CIK	Firm Name	SIC	SIC Description	Cosine Similarity
789019	Microsoft Corp.	7372	Services-Prepackaged Software	1.0000
1660134	Okta, Inc.	7372	Services-Prepackaged Software	0.9622
1463172	Zendesk, Inc.	7374	Services-Computer Processing & Data Preparation	0.9605
1447669	Twilio	7372	Services-Prepackaged Software	0.9598
1104855	Support.com, Inc.	7374	Services-Computer Processing & Data Preparation	0.9594
1372612	Box, Inc.	7372	Services-Prepackaged Software	0.9591
1066194	eGain	7372	Services-Prepackaged Software	0.9587
1108524	Salesforce, Inc.	7372	Services-Prepackaged Software	0.9575
1470099	MobileIron Inc.	7372	Services-Prepackaged Software	0.9570
849399	Symantec Corporation	7372	Services-Prepackaged Software	0.9567
1448056	New Relic	7372	Services-Prepackaged Software	0.9556
...				

(a) 微软公司

CIK	Firm Name	SIC	SIC Description	Cosine Similarity
1403161	Visa Inc.	7389	Services-Business Services, NEC	1.0000
1141391	Mastercard Inc.	7389	Services-Business Services, NEC	0.9683
1123360	Global Payments Inc.	7389	Services-Business Services, NEC	0.9590
1029199	Euronet Worldwide	6099	Functions Related To Depository Banking, NEC	0.9564
1175454	Fleetcor Technologies, Inc.	7389	Services-Business Services, NEC	0.9524
721683	Total System Services, Inc.	7389	Services-Business Services, NEC	0.9507
1144354	Heartland Payment Systems, Inc.	7389	Services-Business Services, NEC	0.9442
1559865	Evertec, Inc.	7374	Services-Computer Processing & Data Preparation	0.9423
1671013	Cardtronics	7389	Services-Business Services, NEC	0.9357
1633917	PayPal Holdings, Inc.	7389	Services-Business Services, NEC	0.9321
1140184	The Western Union Company	7389	Services-Business Services, NEC	0.9281
...				

(b) Visa 公司

图 4.4 Financial-ICS 找到的同行公司

第5章 行业分类系统的可解释性分析

5.1 引言

随着深度学习的不断发展,特别是最近 Transformer 模型在自然语言处理领域表现出的强大性能,自然语言处理正逐渐放弃过去透明的、可解释的白盒技术,如规则、决策树和逻辑回归等,转而采用强大但难以解释的黑盒模型。尽管这些模型的出现显著提高了模型的效果,但逐渐变得不可解释。这种达到结果的过程的模糊性带来了很多问题,例如削弱了人们对模型结果的信任等。

在金融领域中,系统的可解释性对于应用来说至关重要。由于 LongBERT 是一种基于 Transformer 架构的深度学习模型,属于复杂的黑盒模型,其预测过程缺乏透明性。因此,我们需要对其进行可解释性的分析,增加用户对 Financial-ICS 预测结果的信任。SHAP^[65,89]是如今备受欢迎且适用范围广泛的解释方法之一。它提供了可加性的特征重要性度量,可以帮助我们了解文档中各个 token 对模型预测的影响程度。然而,SHAP 对表征模型并不兼容,它的通用实现 Kernel SHAP 是一种模型无关的方法,虽然可以很好适配 BERT 系列的模型,但是需要一个返回概率的分类器函数。在本章中,我们提出了一种对表征模型进行 SHAP 解释的方法,并对 LongBERT 进行了可解释性分析,以增强用户对 Financial-ICS 的信任。

5.2 背景

模型预测的准确性和可解释性^[63-64]之间的矛盾是自然语言处理中面临的重要挑战。特别是在一些领域,如医疗、金融和法律等,对模型预测的可解释性有着比较高的要求。例如,当医生使用深度学习模型来做出医疗方案的决策时,我们通常希望模型可以提供决策背后的原因。这可以影响医生对模型预测的信任程度,并影响是否接纳模型提出的建议。对模型预测进行解释有助于用户建立对人工智能系统的信任,同时增进用户对模型运作的理解,提供更多信息帮助用户做出最终的决策。为了解决这个问题,已经有不少的研究人员开始研究自然语言模型的可解释性问题。

对模型预测的解释可以分为两种分类方式。首先,可以根据解释是针对模型的单次预测还是整个模型的行为将其分为局部解释和全局解释。局部解释指的是对模型在特定输入上的预测进行解释或者提供信息,而全局解释则是对模型的整体预测过程进行解释,与具体的输入无关。其次,可以根据解释是在模型预测过程中产生还是需要在预测后执行额外的操作将其分为自解释和事后解释。

自解释是指模型会在预测的同时生成解释，我们可以直接利用模型预测过程中产生的信息来理解预测过程，而事后解释则需要在模型预测后执行额外的操作，来对模型进行解释。

决策树和基于规则的模型属于全局自解释模型。可以直接对模型是如何工作的提供解释，而无需针对具体输入，并在模型预测过程中生成解释。Transformer^[8]是如今自然语言处理中最重要的模型之一，其中的注意力机制也是常见的可解释方法，属于局部自解释模型。它需要有确定的输入，并在预测过程中产生解释。另外，LIME (Local Interpretable Model-agnostic Explanations)^[90]通过对输入进行扰动来学习代理模型，属于局部事后解释模型。代理模型是指通过学习第二个更容易解释的模型，例如线性模型，来解释模型的预测。由于LIME借助于代理模型的机制与具体模型无关，因此具有很好的灵活性。SHAP (SHapley Additive exPlanations) 值^[65]的思想来自于Shapley值，Shapley值是合作博弈论中广泛使用的方法。SHAP的核心思想是计算特征对模型输出的边际贡献，从而对模型进行解释。SHAP提供了可加性的特征重要性度量，通过将每个特征的存在与否对模型预测值的影响量化地关联起来，从而解释了如何从不含任何特征信息的预测基准值到最终的输出，获取每个特征对最终结果的贡献。

5.3 表征模型的 SHAP 解释方法

SHAP 方法是目前最受欢迎的解释性方法之一，其通用实现 Kernel SHAP 是模型无关的，因此可以用于对 LongBERT 的预测行为进行可解释性分析。然而，我们训练的 LongBERT 是一个表征模型，而 Kernel SHAP 需要一个返回概率的分类器函数，导致两者的不兼容。因此，我们设计了适用于表征模型的 SHAP 解释方法。

为了对 Financial-ICS 中的表征模型 LongBERT 的预测行为进行解释，我们需要对其输入进行扰动，并分析不同 tokens 对模型预测的影响。我们通过将 10K 报告中的一部分文本段替换为“...”来实现对输入的扰动。由于 SHAP 解释方法与嵌入模型的不兼容，我们提出了一种新的方法，通过观察原始输入和扰动后输入的模型预测之间的余弦相似度大小，并将其映射到 0-1 之间的，从而可以利用 SHAP 对表征模型进行可解释性分析。如果扰动前后表征向量的余弦相似度较小，则意味着替换掉的 tokens 对预测有着重要的积极影响。其转换公式如下：

$$\begin{aligned} s &= \text{pairwise_sim}(h, h_0)/\tau \\ p &= \text{sigmoid}(s) \end{aligned} \quad (5.1)$$

其中 $h_0 \in \mathbb{R}^{d_m}$ 代表未对输入进行扰动时模型产生的表征向量， h 表示对输入进行扰动后模型产生的表征向量。 τ 代表温度，我们在实验中将其设置为 1。标量

s 经过 sigmoid 函数得到置信度 p , 使其范围映射到 0-1 之间, 从而转化成分类任务以便使用 SHAP 进行处理。通过对模型输入进行扰动并观察置信度 p 的变化, 我们可以判断每个 token 对输出的贡献程度。

5.4 实验与分析

我们使用适用于表征模型的 SHAP 解释方法对训练后的 LongBERT 进行了可解释性分析, 图5.1展示了模型对苹果公司和 NVIDIA 公司的 10K 报告预测进行解释的结果, 其中只展示了 10K 报告中的部分文本。我们使用黄色标注正向影响的 tokens, 绿色标注负向影响的 tokens。从图中我们可以观察到, 像“Apple”、“IOS”和“macOS”等 tokens 对苹果公司 10K 报告的模型预测结果具有明显的正向影响。而对于 NVIDIA 公司, 类似于“NVIDIA”、“GPU”和“AI”等 tokens 对模型的输出产生了显著的正向影响。通过对 LongBERT 进行可解释性分析, 我们可以了解每个 token 对模型预测的贡献, 这为 Financial-ICS 搜索同行公司提供了支持, 提高了用户对系统的信任, 并给出了更多可以参考的信息帮助用户做出决策。

5.5 本章小结

在本章中, 我们提出了一种对表征模型进行 SHAP 解释的方法。我们使用“...”替换文本段来对输入进行扰动, 通过观察扰动前后模型生成表征向量之间的余弦相似度大小, 来衡量替换的 tokens 对输出的贡献程度。其中, 较小的相似度代表替换掉的 tokens 对输出具有更积极的影响。由于 Kernel SHAP 需要一个返回概率的分类器函数, 我们通过将相似度分数映射到 0-1 之间来适配 SHAP。我们利用这一方法对 LongBERT 模型进行了可解释性分析, 以了解 10K 报告输入中各个 token 对输出的贡献程度, 从而提升用户对 Financial-ICS 的信任。

The Company designs, manufactures and markets mobile communication and media devices and personal computers , and sells a variety of related software, services, accessories and third-party digital content and applications. The Company 's products and services include iPhone®, iPad®, Mac®, Apple Watch®, AirPods®, Apple TV®, HomePod™, a portfolio of consumer and professional software applications, iOS, macOS®, watchOS® and tvOS™ operating systems, iCloud ®, Apple Pay® and a variety of other accessory, service and support offerings. The Company sells and delivers digital content and applications through the iTunes Store®, App Store®, Mac App Store, TV App Store, Book Store and Apple Music® (collectively “Digital Content and Services”). The Company sells its products worldwide through its retail stores , online stores and direct sales force, as well as through third-party cellular network carriers, wholesalers, retailers and resellers. In addition, the Company sells a variety of third-party Apple-compatible products, including application software and various accessories, through its retail and online stores. The Company sells to consumers, small and mid-sized businesses and education, enterprise and government customers. The Company's fiscal year is the 52- or 53-week period that ends on the last Saturday of September. The Company is a California corporation established in 1977.

(a) 苹果公司

Starting with a focus on PC graphics, NVIDIA invented the graphics processing unit, or GPU, to solve some of the most complex problems in computer science. We have extended our focus in recent years to the revolutionary field of artificial intelligence, or AI. Fueled by the sustained demand for better 3D graphics and the scale of the gaming market, NVIDIA has evolved the GPU into a computer brain at the intersection of virtual reality, or VR, high performance computing, or HPC, and AI.

The GPU was initially used to simulate human imagination , enabling the virtual worlds of video games and films. Today , it also simulates human intelligence, enabling a deeper understanding of the physical world. Its parallel processing capabilities, supported by up to thousands of computing cores, are essential to running deep learning algorithms. This form of AI, in which software writes itself by learning from data, can serve as the brain of computers, robots and self-driving cars that can perceive and understand the world . GPU-powered deep learning continues to be adopted by thousands of enterprises to deliver services and features that would have been impossible with traditional coding.

(b) NVIDIA 公司

图 5.1 LongBERT 对 10K 报告预测的可解释性分析结果

第6章 总结与展望

6.1 研究成果总结

本文提出了 Financial-ICS, 一种新的基于算法的 ICS。该系统是基于 Long-BERT 模型构建的, 可以直接处理超过 100K tokens 的 10K 报告长文本, 以生成蕴含公司经济特征的高质量表征。借助 Transformer 强大的语言理解能力, Financial-ICS 在三个自定义指标中均优于当前最先进的基于算法的 ICS, TNIC。在默认相似度阈值下, Financial-ICS 超越了当前广泛使用的专家设计的 SIC, 当我们将相似度阈值提升至每个公司平均搜索 20 个同行公司时, 其性能超过了最先进的专家设计 ICS 之一, NAICS。本次研究的主要贡献如下。

首先, 我们设计了完整的自监督训练框架来对 Financial-ICS 中的表征模型进行训练。我们使用公司 10K 报告作为数据集, 而没有引入任何标签数据。与以往的句子表征学习研究不同的是, 我们针对的是长文档的表征学习。我们采用预训练加对比学习微调的范式, 预训练增强了模型对金融领域文本的理解能力, 而对比学习微调成功缓解了模型生成文档表征中存在的各向异性问题^[48], 从而产生高质量的文档表征。我们展示了使用自监督训练框架对多个 Transformer 架构的模型进行训练后的结果, 包括: RoBERTa、Longformer 和 LongBERT。结果表明其成功改善了模型生成表征的各向异性问题, 且基于这些模型构建的基于算法的 ICSs 的效果均优于 TNIC。

第二, 我们设计并开发了 LongBERT, 通过结合移位块注意力和全局注意力, 将自注意力的计算和内存复杂度从 $O(n^2)$ 降低至 $O(n)$ 。LongBERT 最长可以处理 131K tokens 的长文本, 即可以直接对 10K 报告数据集中的任意文档进行处理, 并且具有双向的上下文理解能力。其中移位块注意力的设计扩展了传统块注意力的局部感受野, 并通过引入全局注意力, 实现了 $O(1)$ 的长依赖学习的操作路径长度。由于 LongBERT 自注意力机制的简洁设计, 我们可以借助批处理技巧对其进行高效实现, 使得在真实场景下的推理速度比时间和内存复杂度同为 $O(n)$ 的 Longformer^[31] 快 1.87 倍。此外, LongBERT 外部拓展的全局 tokens 的设计以及不需要自定义 CUDA 内核就可以高效实现的自注意力机制, 使得该模型具有很强的可拓展性, 可以应用于各种长文本场景中。LongBERT 的出现为高效 Transformer 的家族^[29] 引入了新成员, 并且打破了其能够处理的最大文本长度的限制。

第三, 我们设计了内存优化的对比学习方法, 用于在长文本情况下微调 LongBERT。尽管使用 LongBERT 降低了自注意力机制的内存复杂度至 $O(n)$, 但是训练平均长度超过 20K tokens 的长文本仍需要大量的内存, 这使得对比学习期间

的负样本数量非常有限,甚至导致训练失败的情况。为了解决这个问题,我们设计了一种新的对比损失函数,即基于原型的对比学习损失,并对其进行内存消耗方面的优化。传统对比学习损失通过拉近正样本对的距离,拉远负样本对的距离进行优化。基于原型的对比学习损失从分类的角度思考对比学习,将一组正样本向它们的原型靠拢,而向其他的原型远离。这种对比学习损失的优点是易于理解并且可以处理每个正样本组中样本数量不同或者超过两个的情况。接着,我们对其进行了内存消耗方面的优化。我们只对其中的少数正样本对和它们的原型进行梯度的计算,而将其他的样本对和原型在无梯度上下文中进行计算,只作为负样本的形式提供表征信息。因此,在同等计算环境下,可以在同一批次中引入更多的负样本,从而能够使用长文档对 LongBERT 进行对比学习微调。通过采用这种方法,我们为长文档的表征学习提供了解决方案。

第四,我们提出了三个自定义指标,用于对 Financial-ICS 的性能进行评估。具体包括以下内容:第一,使用股市数据对 Financial-ICS 找到的同行公司的经济相关性进行评估;第二,利用各公司的 SIC 编码和 NAICS 编码对 Financial-ICS 找到的同行公司与专家设计的 ICSs 中的行业结构的一致性进行评估;第三,借助迁移学习任务对 LongBERT 生成的表征进行多分类逻辑回归,评估其生成的表征质量。鉴于目前缺少针对基于算法的 ICSs 进行评估的指标,我们希望本文提出的评估指标能够成为基于算法的 ICSs 评估的标准,或为其提供启发。

最后,我们开发了基于 LongBERT 的 Financial-ICS,这是在三个指标中表现最出色的基于算法的 ICS,并且其性能超过了当前广泛使用的专家设计的 ICS, SIC。当相似度阈值提升至每个公司平均搜索 20 个同行公司时,其性能超过了专家设计的 NAICS。此外,我们提出了表征模型的 SHAP 解释方法。通过对输入中不同 tokens 进行扰动,并观察扰动前后模型生成的文档表征之间的余弦相似度大小,以分析其对模型结果的影响程度。我们对 Financial-ICS 中的表征模型 LongBERT 进行了可解释性分析,以了解模型在预测过程中各个 token 的贡献程度,这可以为用户提供更多的信息帮助他们决策。我们设计的 Financial-ICS 为金融领域的研究和分析提供了更先进的基于算法的 ICS,可以快速且有效地找到经济相关的同行公司。此外,Financial-ICS 可以为找到的同行公司提供相似度分数并进行排序,这在真实的应用场景中非常有用,很多研究需要选择与焦点公司最相似的公司来构建基线或者作为对照。并且,Financial-ICS 可以根据最新的 10K 报告进行快速地更新迭代,这些都是专家设计的 ICSs 无法做到的。

6.2 局限性与展望

本研究提出了若干方法来将 Transformer 应用于基于算法的 ICS 的构建中, 包括自监督训练框架、注意力高效的 LongBERT 模型、内存优化的对比学习方法以及表征模型的可解释性分析。但仍存在一定的不足, 接下来我们将介绍本研究的局限性和未来的展望。

首先, 我们使用的是完全自监督的方式对模型进行训练, 这是 Financial-ICS 的优势, 例如降低数据的标注成本和提供更强的可拓展性等, 但是也存在其局限性, 例如其性能可能不如有监督的情况。已经有研究表明, 在对比学习中引入监督信号可以提升性能^[52]。因此, 未来可以探索在对比学习微调过程中引入监督信号, 进一步提升行业分类系统的性能。

其次, 我们在对比学习中使用 dropout 作为数据增强的方法, 以生成正样本对。然而, 这种方法并未充分利用基于原型的对比学习损失的优势, 即可以在正样本组中样本数量不同或者超过两个的情况下进行训练。此外, 我们还可以研究更有效的适用于长文档对比学习的数据增强方法。因此, 未来的研究可以通过引入更多更有效的数据增强方法, 来提升对比学习后模型的表征质量, 从而进一步改善基于算法的 ICS 的性能。

最后, 我们可以探索更广泛的应用场景。目前, 我们使用上市公司每年提交到美国证券交易委员会的 10K 报告作为模型的文本输入, 以对我们提出的表征模型和行业分类系统进行训练和评估。未来, 我们可以将研究的范围扩展到国内的公司, 并将其财报作为输入, 建设国内的基于算法 ICS。这将使我们可以更好的分析国内的商业环境、市场竞争情况和就业环境等, 为企业运营者、监管者和投资者提供更准确和实时的信息, 帮助他们做出更精准的决策。

参考文献

- [1] BIZJAK J, LEMMON M, NGUYEN T. Are all ceos above average? an empirical analysis of compensation peer groups and pay design[J]. Journal of financial economics, 2011, 100(3): 538-555.
- [2] LI F, LUNDHOLM R, MINNIS M. A measure of competition based on 10-k filings[J]. Journal of Accounting Research, 2013, 51(2): 399-436.
- [3] BHOJRAJ S, LEE C M, OLER D K. What's my line? a comparison of industry classification schemes for capital market research[J]. Journal of Accounting Research, 2003, 41(5): 745-774.
- [4] HOBERG G, PHILLIPS G. Text-based network industries and endogenous product differentiation[J]. Journal of Political Economy, 2016, 124(5): 1423-1465.
- [5] LEE C M, MA P, WANG C C. Search-based peer firms: Aggregating investor perceptions through internet co-searches[J]. Journal of Financial Economics, 2015, 116(2): 410-431.
- [6] SPRENGER T O, WELPE I M. Tweets and peers: defining industry groups and strategic peers based on investor perceptions of stocks on twitter[J]. Algorithmic Finance, 2011, 1(1): 57-76.
- [7] JAFFE A B. Technological opportunity and spillovers of r&d: evidence from firms' patents, profits and market value[M]. national bureau of economic research Cambridge, Mass., USA, 1986.
- [8] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.
- [9] DEVLIN J, CHANG M W, LEE K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[A]. 2019. arXiv: 1810.04805.
- [10] BROWN T, MANN B, RYDER N, et al. Language models are few-shot learners[J]. Advances in neural information processing systems, 2020, 33: 1877-1901.
- [11] LEWIS M, LIU Y, GOYAL N, et al. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension[A]. 2019. arXiv: 1910.13461.
- [12] DOSOVITSKIY A, BEYER L, KOLESNIKOV A, et al. An image is worth 16x16 words: Transformers for image recognition at scale[A]. 2021. arXiv: 2010.11929.
- [13] LIU Z, LIN Y, CAO Y, et al. Swin transformer: Hierarchical vision transformer using shifted windows[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2021: 10012-10022.
- [14] LIU Z, HU H, LIN Y, et al. Swin transformer v2: Scaling up capacity and resolution[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022: 12009-12019.

- [15] DONG L, XU S, XU B. Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition[C]//2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2018: 5884-5888.
- [16] GULATI A, QIN J, CHIU C C, et al. Conformer: Convolution-augmented transformer for speech recognition[A]. 2020. arXiv: 2005.08100.
- [17] LIU Y, OTT M, GOYAL N, et al. Roberta: A robustly optimized bert pretraining approach [A]. 2019. arXiv: 1907.11692.
- [18] LAN Z, CHEN M, GOODMAN S, et al. Albert: A lite bert for self-supervised learning of language representations[A]. 2020. arXiv: 1909.11942.
- [19] SANH V, DEBUT L, CHAUMOND J, et al. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter[A]. 2020. arXiv: 1910.01108.
- [20] CLARK K, LUONG M T, LE Q V, et al. Electra: Pre-training text encoders as discriminators rather than generators[A]. 2020. arXiv: 2003.10555.
- [21] JOSHI M, CHEN D, LIU Y, et al. Spanbert: Improving pre-training by representing and predicting spans[J]. Transactions of the association for computational linguistics, 2020, 8: 64-77.
- [22] WANG W, BI B, YAN M, et al. Structbert: Incorporating language structures into pre-training for deep language understanding[A]. 2019. arXiv: 1908.04577.
- [23] ZHANG Z, HAN X, LIU Z, et al. Ernie: Enhanced language representation with informative entities[A]. 2019. arXiv: 1905.07129.
- [24] RADFORD A, NARASIMHAN K, SALIMANS T, et al. Improving language understanding by generative pre-training[M]. OpenAI, 2018.
- [25] RADFORD A, WU J, CHILD R, et al. Language models are unsupervised multitask learners [J]. OpenAI blog, 2019, 1(8): 9.
- [26] HOCHREITER S, SCHMIDHUBER J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.
- [27] CHUNG J, GULCEHRE C, CHO K, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling[A]. 2014. arXiv: 1412.3555.
- [28] CHO K, VAN MERRIENBOER B, GULCEHRE C, et al. Learning phrase representations using rnn encoder-decoder for statistical machine translation[A]. 2014. arXiv: 1406.1078.
- [29] TAY Y, DEGHANI M, BAHRI D, et al. Efficient transformers: A survey[J]. ACM Computing Surveys, 2022, 55(6): 1-28.
- [30] CHILD R, GRAY S, RADFORD A, et al. Generating long sequences with sparse transformers [A]. 2019. arXiv: 1904.10509.
- [31] BELTAGY I, PETERS M E, COHAN A. Longformer: The long-document transformer[A].

2020. arXiv: 2004.05150.
- [32] ZAHEER M, GURUGANESH G, DUBEY K A, et al. Big bird: Transformers for longer sequences[J]. Advances in neural information processing systems, 2020, 33: 17283-17297.
- [33] KITAEV N, ŁUKASZ KAISER, LEVSKAYA A. Reformer: The efficient transformer[A]. 2020. arXiv: 2001.04451.
- [34] ZHOU H, ZHANG S, PENG J, et al. Informer: Beyond efficient transformer for long sequence time-series forecasting[C]//Proceedings of the AAAI conference on artificial intelligence: Vol. 35. 2021: 11106-11115.
- [35] LIU P J, SALEH M, POT E, et al. Generating wikipedia by summarizing long sequences[A]. 2018. arXiv: 1801.10198.
- [36] WANG S, LI B Z, KHABSA M, et al. Linformer: Self-attention with linear complexity[A]. 2020. arXiv: 2006.04768.
- [37] KATHAROPOULOS A, VYAS A, PAPPAS N, et al. Transformers are rnns: Fast autoregressive transformers with linear attention[C]//International conference on machine learning. PMLR, 2020: 5156-5165.
- [38] CHOROMANSKI K, LIKHOSHERSTOV V, DOHAN D, et al. Rethinking attention with performers[A]. 2022. arXiv: 2009.14794.
- [39] TAY Y, BAHRI D, METZLER D, et al. Synthesizer: Rethinking self-attention for transformer models[C]//International conference on machine learning. PMLR, 2021: 10183-10192.
- [40] TOLSTIKHIN I O, HOULSBY N, KOLESNIKOV A, et al. Mlp-mixer: An all-mlp architecture for vision[J]. Advances in neural information processing systems, 2021, 34: 24261-24272.
- [41] DAI Z, YANG Z, YANG Y, et al. Transformer-xl: Attentive language models beyond a fixed-length context[A]. 2019. arXiv: 1901.02860.
- [42] RAE J W, POTAPENKO A, JAYAKUMAR S M, et al. Compressive transformers for long-range sequence modelling[A]. 2019. arXiv: 1911.05507.
- [43] DING S, SHANG J, WANG S, et al. Ernie-doc: A retrospective long-document modeling transformer[A]. 2021. arXiv: 2012.15688.
- [44] WU Y, RABE M N, HUTCHINS D, et al. Memorizing transformers[A]. 2022. arXiv: 2203.08913.
- [45] ZHANG H, GONG Y, SHEN Y, et al. Poolingformer: Long document modeling with pooling attention[C]//International Conference on Machine Learning. PMLR, 2021: 12437-12446.
- [46] WU Q, LAN Z, QIAN K, et al. Memformer: A memory-augmented transformer for sequence modeling[A]. 2022. arXiv: 2010.06891.
- [47] WANG T, ISOLA P. Understanding contrastive representation learning through alignment and uniformity on the hypersphere[C]//International Conference on Machine Learning. PMLR,

- 2020: 9929-9939.
- [48] ETHAYARAJH K. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings[A]. 2019. arXiv: 1909.00512.
 - [49] LI B, ZHOU H, HE J, et al. On the sentence embeddings from pre-trained language models [A]. 2020. arXiv: 2011.05864.
 - [50] WANG L, HUANG J, HUANG K, et al. Improving neural language generation with spectrum control[C]//International Conference on Learning Representations. 2019.
 - [51] REIMERS N, GUREVYCH I. Sentence-bert: Sentence embeddings using siamese bert-networks[A]. 2019. arXiv: 1908.10084.
 - [52] GAO T, YAO X, CHEN D. Simcse: Simple contrastive learning of sentence embeddings[A]. 2022. arXiv: 2104.08821.
 - [53] WU X, GAO C, ZANG L, et al. Esimcse: Enhanced sample building method for contrastive learning of unsupervised sentence embedding[A]. 2022. arXiv: 2109.04380.
 - [54] YAN Y, LI R, WANG S, et al. Consert: A contrastive framework for self-supervised sentence representation transfer[A]. 2021. arXiv: 2105.11741.
 - [55] JIANG T, JIAO J, HUANG S, et al. Promptbert: Improving bert sentence embeddings with prompts[A]. 2022. arXiv: 2201.04337.
 - [56] CHUANG Y S, DANGOVSKI R, LUO H, et al. Diffcse: Difference-based contrastive learning for sentence embeddings[A]. 2022. arXiv: 2204.10298.
 - [57] WANG H, DOU Y. Sncse: Contrastive learning for unsupervised sentence embedding with soft negative samples[C]//International Conference on Intelligent Computing. Springer, 2023: 419-431.
 - [58] HE K, FAN H, WU Y, et al. Momentum contrast for unsupervised visual representation learning[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 9729-9738.
 - [59] AGIRRE E, BANEJA C, CER D, et al. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation[C]//SemEval-2016. 10th International Workshop on Semantic Evaluation; 2016 Jun 16-17; San Diego, CA. Stroudsburg (PA): ACL; 2016. p. 497-511. ACL (Association for Computational Linguistics), 2016.
 - [60] CER D, DIAB M, AGIRRE E, et al. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation[C/OL]//Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). Association for Computational Linguistics, 2017. <http://dx.doi.org/10.18653/v1/S17-2001>. DOI: 10.18653/v1/s17-2001.
 - [61] CHEN T, KORNBLITH S, NOROUZI M, et al. A simple framework for contrastive learning of visual representations[C]//International conference on machine learning. PMLR, 2020: 1597-

- 1607.
- [62] CONNEAU A, KIELA D. Senteval: An evaluation toolkit for universal sentence representations[A]. 2018. arXiv: 1803.05449.
 - [63] DANILEVSKY M, QIAN K, AHARONOV R, et al. A survey of the state of explainable ai for natural language processing[A]. 2020. arXiv: 2010.00711.
 - [64] GURRAPU S, KULKARNI A, HUANG L, et al. Rationalization for explainable nlp: A survey [J]. *Frontiers in Artificial Intelligence*, 2023, 6.
 - [65] LUNDBERG S M, LEE S I. A unified approach to interpreting model predictions[J]. *Advances in neural information processing systems*, 2017, 30.
 - [66] LIN T, WANG Y, LIU X, et al. A survey of transformers[J]. *AI Open*, 2022.
 - [67] 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.
 - [68] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]//*Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016: 770-778.
 - [69] BA J L, KIROS J R, HINTON G E. Layer normalization[A]. 2016. arXiv: 1607.06450.
 - [70] MIKOLOV T, CHEN K, CORRADO G, et al. Efficient estimation of word representations in vector space[A]. 2013. arXiv: 1301.3781.
 - [71] MIKOLOV T, SUTSKEVER I, CHEN K, et al. Distributed representations of words and phrases and their compositionality[J]. *Advances in neural information processing systems*, 2013, 26.
 - [72] PENNINGTON J, SOCHER R, MANNING C D. Glove: Global vectors for word representation[C]//*Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014: 1532-1543.
 - [73] XU L, XIE H, LI Z, et al. Contrastive learning models for sentence representations[J]. *ACM Transactions on Intelligent Systems and Technology*, 2023, 14(4): 1-34.
 - [74] QIU X, SUN T, XU Y, et al. Pre-trained models for natural language processing: A survey[J]. *Science China Technological Sciences*, 2020, 63(10): 1872-1897.
 - [75] KALYAN K S, RAJASEKHARAN A, SANGEETHA S. Ammus : A survey of transformer-based pretrained models in natural language processing[A]. 2021. arXiv: 2108.05542.
 - [76] SU J, CAO J, LIU W, et al. Whitening sentence representations for better semantics and faster retrieval[A]. 2021. arXiv: 2103.15316.
 - [77] CHEN T, KORNBLITH S, SWERSKY K, et al. Big self-supervised models are strong semi-supervised learners[J]. *Advances in neural information processing systems*, 2020, 33: 22243-22255.

- [78] SENNRICH R, HADDOW B, BIRCH A. Improving neural machine translation models with monolingual data[A]. 2016. arXiv: 1511.06709.
- [79] VAN DEN OORD A, LI Y, VINYALS O. Representation learning with contrastive predictive coding[A]. 2019. arXiv: 1807.03748.
- [80] SRIVASTAVA N, HINTON G, KRIZHEVSKY A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. The journal of machine learning research, 2014, 15(1): 1929-1958.
- [81] REIMERS N, BEYER P, GUREVYCH I. Task-oriented intrinsic evaluation of semantic textual similarity[C]//Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. 2016: 87-96.
- [82] WOLF T, DEBUT L, SANH V, et al. Huggingface's transformers: State-of-the-art natural language processing[A]. 2020. arXiv: 1910.03771.
- [83] SUN C, QIU X, XU Y, et al. How to fine-tune bert for text classification?[C]//Chinese computational linguistics: 18th China national conference, CCL 2019, Kunming, China, October 18–20, 2019, proceedings 18. Springer, 2019: 194-206.
- [84] LOSHCHILOV I, HUTTER F. Decoupled weight decay regularization[A]. 2019. arXiv: 1711.05101.
- [85] CHEN T, XU B, ZHANG C, et al. Training deep nets with sublinear memory cost[A]. 2016. arXiv: 1604.06174.
- [86] SU J, LU Y, PAN S, et al. Roformer: Enhanced transformer with rotary position embedding [A]. 2023. arXiv: 2104.09864.
- [87] YU F, KOLTUN V. Multi-scale context aggregation by dilated convolutions[A]. 2016. arXiv: 1511.07122.
- [88] RASLEY J, RAJBHANDARI S, RUWASE O, et al. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters[C]//Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020: 3505-3506.
- [89] KOKALJ E, ŠKRLJ B, LAVRAČ N, et al. Bert meets shapley: Extending shap explanations to transformer-based classifiers[C]//Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation. 2021: 16-21.
- [90] RIBEIRO M T, SINGH S, GUESTRIN C. " why should i trust you?" explaining the predictions of any classifier[C]//Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. 2016: 1135-1144.