Software College Of
Northeastern University

基于改进 distilbert 实现评论文本情感分析

AUTHORS

李静薇 - 20220940

段国霄 - 20226720

罗亚静 - 20226948

陈禹彤 - 20226642

2024-10-23

Contents

1 引言 **1**
1.1 项目背景 1
1.2 项目动机 1

2 相关工作 **1**
2.1 Knowledge Distillation 介绍 1
2.2 DistilBERT 介绍 2
2.3 DistilBERT 的优势 3

3 项目信息 **4**
3.1 项目描述 4
3.2 实验环境 5
3.3 创新点 5
3.4 研究假设 6

4 实现过程与结果 **7**
4.1 模型配置 7
4.2 数据集说明 7
4.3 实现过程 9
4.4 性能分析 13

5 总结 **13**
5.1 主要发现 13
5.2 创新点及其潜在影响 13

List of Figures **I**

References **II**

1 引言

1.1 项目背景

自然语言处理（NLP）在近几年取得了飞速的发展，尤其是基于深度学习的预训练语言模型。BERT（Bidirectional Encoder Representations from Transformers）模型作为突破性模型之一，极大提高了多项自然语言任务的表现。然而，由于 BERT 的模型参数量大，推理速度较慢，因此 Hugging Face 提出了 DistilBERT，一个蒸馏版本的 BERT。DistilBERT 在保留 BERT 模型大部分性能的前提下，显著减少了模型的参数量，提高了推理速度，使其更适合资源受限的场景。

1.2 项目动机

情感分析任务广泛应用于商业决策和社会研究中，通过分析用户评论、社交媒体发言等内容，可以迅速获取公众对某个产品或话题的情感态度。然而，传统的 NLP 模型对文本序列信息的捕捉较为有限，而情感分析往往需要模型能够深刻理解文本中潜在的语义和序列关系。在现有的 DistilBERT 基础上，本项目希望通过增加创新点来进一步提升模型表现，使模型更加适应真实世界中的复杂文本数据，如冗长评论、情感模糊表达等，从而提升其在情感分析任务中的精度与鲁棒性。

2 相关工作

2.1 Knowledge Distillation 介绍

知识蒸馏（Knowledge Distillation）是一种模型压缩技术，通过将复杂的大型模型（称为教师模型）中的知识“蒸馏”到一个较小的模型（称为学生模型）中，从而保持较高的性能。蒸馏过程通过最小化教师模型输出的软标签（soft targets）和学生模型预测之间的差异，使得学生模型能够学习到教师模型的深层特征和预测模式。尽管学生模型规模较小，但它通过这种蒸馏方式可以接近甚至达到教师模型的精度和表现，同时显著减少了计算资源的需求。因此，知识蒸馏常用于在设备受限的环境中部署高效的机器学习模型，比如在移动设备或嵌入式系统中使用。

在图表 1 中展示了知识蒸馏的应用过程，教师模型（如 OpenAI GPT）经过训练后生成高质量的输出表示，学生模型（如 BlendCNN）通过蒸馏损失学习教师模型的知识。学生模型采用较简单的卷积神经网络架构，输入与教师模型相同的数据，并通过对比教师模型的输出（软标签）来优化自身性能。尽管学生模型规模较小，但通过知识蒸馏，它能在推理时保持高效的同时保留教师模型的知识和性能，特别适合计算资源有限的应用场景。

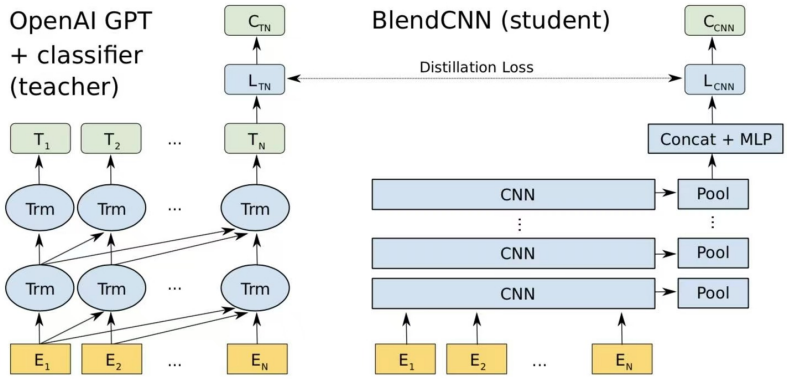

Figure 1: Knowledge Distillation 实际应用

2.2 DistilBERT 介绍

DistilBERT 是由 Hugging Face 团队提出的一种通过知识蒸馏技术压缩 BERT 模型的轻量级版本。它保持了 BERT 的双向 Transformer 结构，但通过减少模型层数和蒸馏学习策略，大幅度降低了模型的参数量和推理时间。

DistilBERT 通过知识蒸馏技术，将 BERT 模型中的重要信息压缩到一个更小的模型中。知识蒸馏是一种用大模型（称为教师模型）指导小模型（称为学生模型）学习的技术。在这个过程中，学生模型（DistilBERT）学习教师模型的预测输出，从而保留原始模型的关键信息。同时它通过减少 BERT 的 Transformer 层数，从 BERT 的 12 层减少到 6 层，使其更加轻量。尽管如此，它通过知识蒸馏保留了相对较高的性能。另外，它通过减少 40% 的参数量和 60% 的推理时间，使其成为在实际应用场景中非常高效的选择，同时其准确率仅下降了 2% 左右。

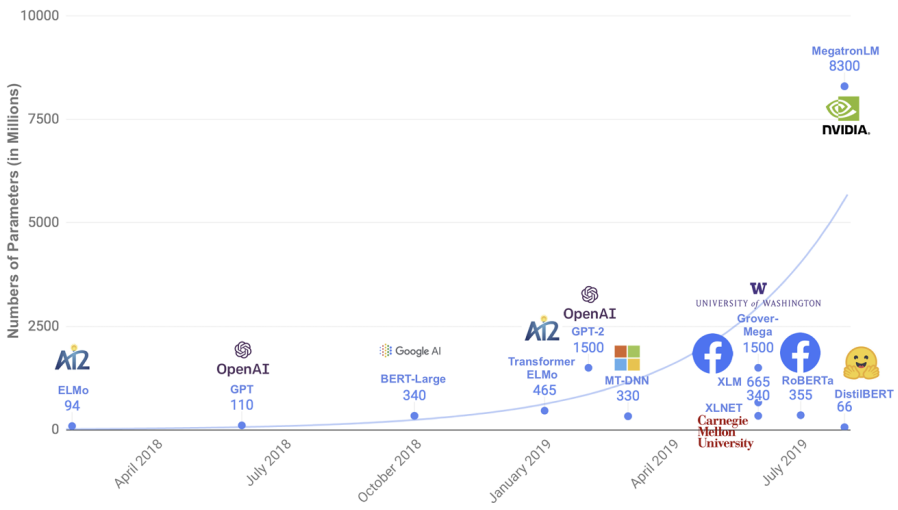

Figure 2: DistilBERT 的参数与其他模型的对比 [1]

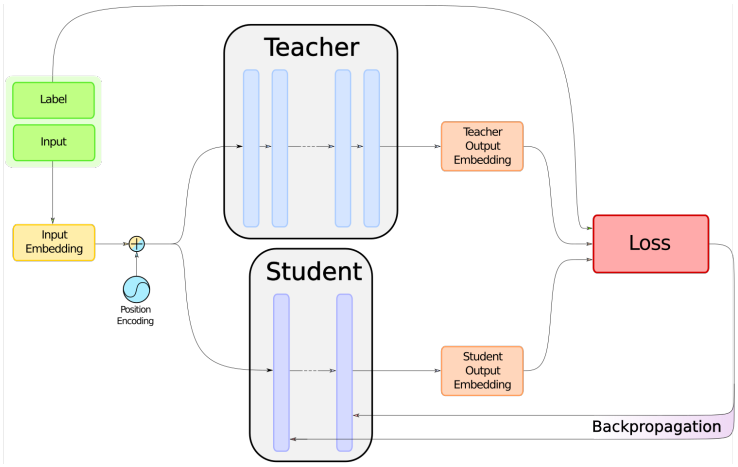

Figure 3: distilbert 的基本架构

2.3 DistilBERT 的优势

根据图表 3 的数据显示，DistilBERT 在 GLUE 基准测试中的表现相当于 BERT，保留了约 97% 的性能。这一结果强调了其在处理多种下游任务时的有效性，尤其是在情感分析和问答任务中，表明模型即便在减少参数的情况下依然能提供高质量的预测。

Model	**Score**	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	WNLI
ELMo	68.7	44.1	68.6	76.6	71.1	86.2	53.4	91.5	70.4	56.3
BERT-base	79.5	56.3	86.7	88.6	91.8	89.6	69.3	92.7	89.0	53.5
DistilBERT	77.0	51.3	82.2	87.5	89.2	88.5	59.9	91.3	86.9	56.3

Figure 4: DistilBERT retains 97% of BERT performance [1]

在图表 5 中，我们可以看到 DistilBERT 在模型大小和推理速度上的明显优势。图表显示，DistilBERT 显著较小，同时在 CPU 上进行推理时速度更快，尤其是在批处理大小为 1 的情况下。这一特性使其在资源受限的环境中具有更高的适用性和效率。

Model	# param. (Millions)	Inf. time (seconds)
ELMo	180	895
BERT-base	110	668
DistilBERT	66	410

Figure 5: DistilBERT is significantly smaller while being constantly faster. [1]

最后，图表 6 展示了 DistilBERT 在下游任务上的准确性与 BERT 的比较，结果表明，DistilBERT 在 IMDb 电影评论情感分析和 SQuAD 1.1 的准确性方面表现出色。这些结果进一步验证了 DistilBERT 作为一个轻量级、有效的自然语言处理工具的潜力，使其成为许多实际应用的首选。

Model	IMDb (acc.)	SQuAD (EM/F1)
BERT-base	93.46	81.2/88.5
DistilBERT	92.82	77.7/85.8
DistilBERT (D)	-	79.1/86.9

Figure 6: DistilBERT yields to comparable performance on downstream tasks. [1]

3 项目信息

3.1 项目描述

本项目旨在利用 DistilBERT 预训练模型，构建一个情感分析系统，主要应用于文本分类任务。具体而言，项目采用 Yelp 的评论数据集（yelp_review_full），对模型进行微调，以便自动识别用户评论的情感倾向（如正面或负面）。通过结合 TensorFlow 架构和 Hugging Face 提供

的 distilbert-base-uncased 模型，我们可以在提高计算效率的同时保持较高的预测准确性。该系统的最终目标是优化文本分类的表现，并探索如何通过创新性技术提升模型在下游任务中的表现。

3.2 实验环境

- **架构**：TensorFlow
- **GPU**：NVIDIA GeForce RTX 4050 Laptop GPU，Google COLAB

3.3 创新点

1. **数据增强策略**: 使用领域自适应增强，将同样类型不同领域的数据集融合在一起构建新的数据集，增强模型在多领域的泛化能。我们将探索通过数据增强的方式，进一步提升模型在处理噪声数据和多样化输入时的表现。例如，加入同义词替换、拼写错误模拟等数据增强技术，可以帮助模型更好地应对非标准化文本输入，增强模型的鲁棒性和泛化能力。

2. **原始嵌入和位置编码的结合方式**：在传统 BERT 模型中，位置编码用于表征输入序列中每个单词的位置信息。然而，本项目通过创新性地结合原始嵌入和位置编码，试图提高模型对长文本的理解能力。通过这种方法，模型不仅可以捕捉词与词之间的上下文关系，还能更准确地对不同位置的单词进行处理。这种结合有助于增强情感分析的精度，特别是在处理具有复杂结构的文本时。

3. **前馈神经网络中添加卷积层**: 在前馈神经网络中添加卷积层能够有效地捕获输入数据的局部特征。卷积层通过局部感受野对输入进行处理，提取局部区域的特征，这种方式比传统的全连接层更能捕捉到数据中的空间结构信息。卷积操作能够减少参数数量，增强模型的泛化能力，同时保留重要的局部信息。这种结合使得网络在处理图像、文本等数据时，能够更好地理解和分析局部模式，提高整体性能。

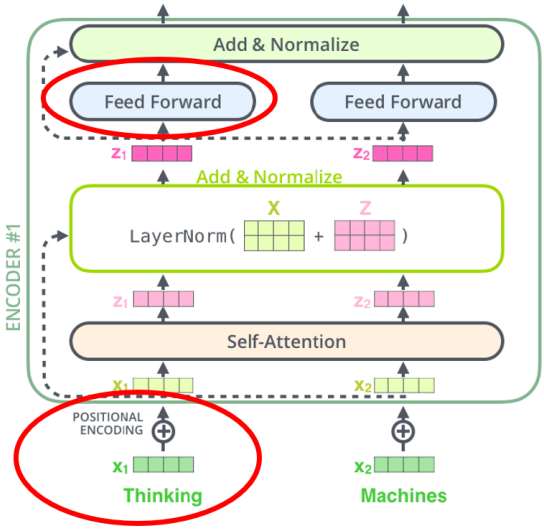

Figure 7: 模型架构的创新点体现

3.4 研究假设

1. 结合原始嵌入和位置编码的方式能够提升模型在处理长文本时的情感分类准确性。

2. 通过引入数据增强技术，模型在应对具有噪声的数据时表现会有显著提升，从而提高对不同风格和结构的文本的理解能力，增强对真实世界场景下评论文本的适应性。

3. 引入卷积层能够显著提高模型对输入数据局部特征的提取能力，从而改善在任务中的性能。这种网络结构在处理序列数据（如文本或时间序列）时，假设能够更好地捕捉上下文关系和局部依赖性，最终导致在分类或回归任务中的准确率提升。

4. DistilBERT 模型在微调后，能够在情感分析任务中实现高效的计算性能，并在保持较高预测准确性的同时，减少计算成本和模型大小。

4 实现过程与结果

4.1 模型配置


```
{
  "activation": "gelu",
  "architectures": [
    "DistilBertForMaskedLM"
  ],
  "attention_dropout": 0.1,
  "dim": 768,
  "dropout": 0.1,
  "hidden_dim": 3072,
  "initializer_range": 0.02,
  "max_position_embeddings": 512,
  "model_type": "distilbert",
  "n_heads": 12,
  "n_layers": 6,
  "pad_token_id": 0,
  "qa_dropout": 0.1,
  "seq_classif_dropout": 0.2,
  "sinusoidal_pos_embds": false,
  "tie_weights_": true,
  "transformers_version": "4.10.0.dev0",
  "vocab_size": 30522
}
```


Figure 8: distilbert 模型的配置信息

4.2 数据集说明

本项目使用 Yelp/yelp_revi_full 数据集和 SetFit/sst5 数据集。

Yelp/yelp_revi_full 来自 Hugging Face 数据集库，是一个著名的情感分析数据集，常用于训练和评估模型在处理用户评论时的表现。该数据集包含大量的用户评论，内容丰富，主要用于文本分类任务，特别是情感分析。该数据集共有 650,000 条用户评论，其中每条评论都已被

标注为情感标签。评论文本主要是关于餐馆、商店等服务行业的评价，涵盖了广泛的用户体验，评论从几句话到几段不等。它共分为 5 个情感等级（标签从 1 到 5），表示用户对服务的评分。1 表示非常负面，5 表示非常正面。每个情感等级的评论数量大致相等，确保模型在训练时能够接触到不同类型的情感。

label class label 5 classes	text string · *lengths* 1 5.64k
4 5 stars	dr. goldberg offers everything i look for in a general practitioner. he's nice and easy to talk to without being patronizing; he's always on time in seeing…
1 2 star	Unfortunately, the frustration of being Dr. Goldberg's patient is a repeat of the experience I've had with so many other doctors in NYC -- good doctor,…
3 4 stars	Been going to Dr. Goldberg for over 10 years. I think I was one of his 1st patients when he started at MHMG. He's been great over the years and is really…
3 4 stars	Got a letter in the mail last week that said Dr. Goldberg is moving to Arizona to take a new position there in June. He will be missed very much. \n\nI think…
0 1 star	I don't know what Dr. Goldberg was like before moving to Arizona, but let me tell you, STAY AWAY from this doctor and this office. I was going to Dr.…
4 5 stars	Top notch doctor in a top notch practice. Can't say I am surprised when I was referred to him by another doctor who I think is wonderful and because he went…

Figure 9: Yelp/yelp_revi_full 数据集

SST-5（Stanford Sentiment Treebank 5）数据集是一个用于情感分析的标准数据集，包含来自电影评论的句子，旨在帮助研究人员和开发者在自然语言处理（NLP）任务中评估和比较模型的性能。该数据集包含五个情感类别：非常消极、消极、中立、积极和非常积极，允许模型捕捉到细微的情感差异。它的数据来源于 Stanford Sentiment Treebank，包括对电影评论的解析，句子被标注为其情感极性。数据集不仅提供了原始句子，还包含了句子的树结构，以便研究者能够利用句子结构信息进行更深入的分析。由于其丰富的情感标注和结构信息，SST-5 成为情感分析领域中的重要基准数据集，广泛应用于各种机器学习和深度学习模型的训练和评估中。

text string · *lengths* 4 283	label int64 0 4	label_text string · *classes* 5 values
a stirring , funny and finally transporting re-imagining of beauty and the beast and 1930s horror films	4	very positive
apparently reassembled from the cutting-room floor of any given daytime soap .	1	negative
they presume their audience wo n't sit still for a sociology lesson , however entertainingly presented , so they trot out…	1	negative
the entire movie is filled with deja vu moments .	2	neutral
this is a visually stunning rumination on love , memory , history and the war between art and commerce .	3	positive
um , no. .	2	neutral

Figure 10: SetFit/sst5 数据集

4.3 实现过程

1. 在项目的初始阶段，加载 Yelp/yelp_review_full 数据集。通过 Hugging Face 的 datasets 模块，能够高效地获取并加载大规模的文本数据。加载数据后，进行必要的预处理操作。

```
tokenizer = AutoTokenizer.from_pretrained("C:/Users/10520/Desktop/huggingface_model/distilbert")


def tokenize_function(examples):
    return tokenizer(examples["text"], padding="max_length", truncation=True)

tokenized_datasets = dataset.map(tokenize_function, batched=True)
```

Figure 11: 加载数据集 数据预处理

2. 为了进一步提升模型的效果和表现，在 DistilBERT 的基础上进行了自定义模型的设计。在保持原有模型结构的基础上，加入了创新点：

   - **重写 TFEmbedding**：我们对模型中的 TFEmbedding 进行了改写，将原本的位置编码和原始嵌入由简单的相加操作改为串联操作。这种操作使得模型能够更好地保留位置信息和嵌入信息的独立性，进而提升对复杂文本的理解能力。

```
class CustomTFEmbeddings(tf.keras.layers.Layer):
```

Figure 12: 重写源代码中的 TFEmbedding

   - **添加线性层**：为了优化情感分类的表现以及避免和后面的层发生形状冲突，我们在模型后端添加了一层线性层，使得模型在输出分类结果前可以进行进一步的特征提取和优化，同时结合后的向量可以更改自己的形状，作为自注意力层的正确输入。这样可以提高模型在处理具有不同情感倾向的文本时的准确性。

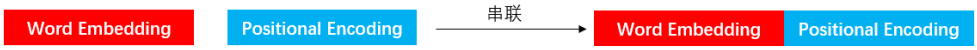

Figure 13: 相加变为串联

```python
# 使用串联操作
final_embeddings = tf.concat([inputs_embeds, position_embeds], axis=-1)

final_embeddings = self.LayerNorm(final_embeddings)
final_embeddings = self.dropout(final_embeddings, training=training)

# 添加线性层
final_embeddings = self.linear_layer(final_embeddings)
```

Figure 14: 从相加改为串联操作 添加线性层

- **重写源代码中的 TFFFN**：在原有的两个线性层中间添加一个卷积层。这种设计有利于模型更好地关注和提取局部特征，因而实现更有效的特征学习和分类。

```python
# 第一个线性层
self.lin1 = tf.keras.layers.Dense(
    config.hidden_dim, kernel_initializer=get_initializer(config.initializer_range), name="lin1"
)

# 添加卷积层
self.conv1d = tf.keras.layers.Conv1D(
    filters=config.hidden_dim,  # 卷积核的个数应该匹配输入的维度
    kernel_size=3,  # 卷积核的大小，你可以根据需求调整
    padding='same',  # 保持输入和输出的长度相同
    activation='relu',  # 激活函数
    name="conv1d"
)

# 第二个线性层
self.lin2 = tf.keras.layers.Dense(
    config.dim, kernel_initializer=get_initializer(config.initializer_range), name="lin2"
)
```

Figure 15: 重写源代码中的 TFFFN

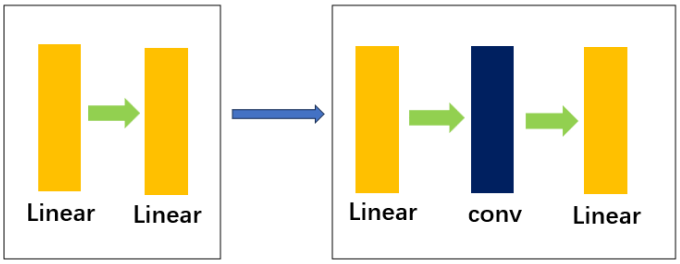

Figure 16: FFN 变化

3. 替换掉原有的类，通过 python 面向对象和动态加载的特性，替换掉源代码中的对应类

```
from transformers.models.distilbert import modeling_tf_distilbert


modeling_tf_distilbert.TFEmbeddings = CustomTFEmbeddings

modeling_tf_distilbert.TFFFN = CustomTFFFN
```

Figure 17: 替换掉源代码中的对应类

4. 在自定义模型和超参数设置完成后，加载 DistilBERT 预训练模型，并基于预处理后的数据进行模型训练，同时对超参数进行调优。

   - 使用 transformer 自带的 from_pretrained 加载模型架构、权重及配置
   - 最初使用 SGD 优化器，最终使用 AdamW 优化器
   - 受限于显卡内存，取不会出现 OOM 的最大值 32
   - 损失函数使用 transformer 库当中 distilbert 自带的损失函数（交叉熵函数）

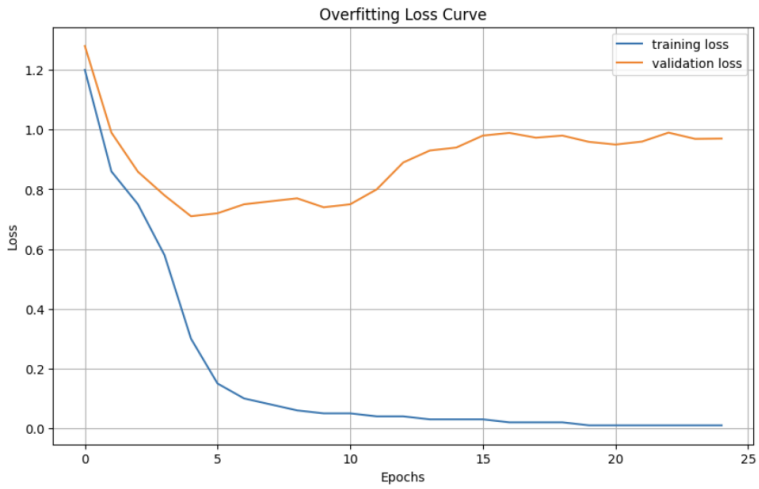

Figure 18: 模型初期的损失曲线

   - 一开始在训练时使用的 SGD 优化器，也没有额外设定任何调参辅助。训练损失快速下降，说明模型在训练集上表现非常优异，但是验证损失却逐步升高，说明模型在验证集上表现不良，该模型呈现出明显的过拟合状态。

- 之后将优化器改为了具有权重衰减的 AdamW 优化器，同时添加了早停回调函数加快调参效率，使用了 ReduceLROnPlateau 调度器，让模型在训练过程中自动调整学习率，同时避免局部最优解的问题。经过多轮不断的调试，模型最终呈现收敛的状态，在训练集和验证集上都表现良好。如图 15 和图 16 是调整后模型的损失曲线和准确率曲线。

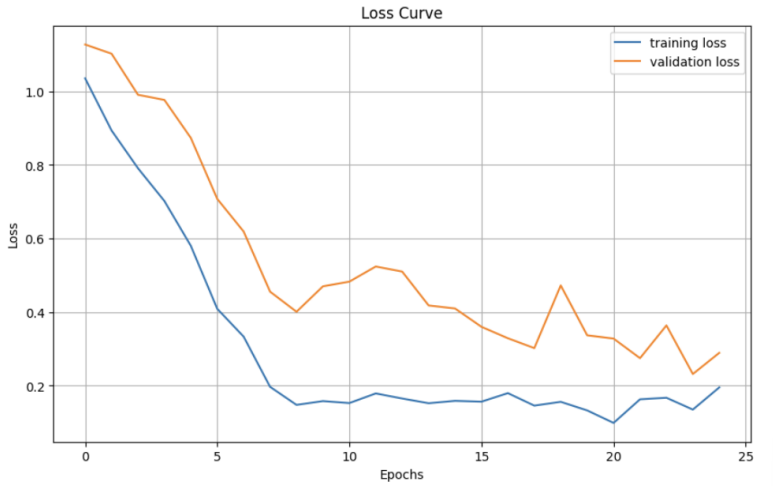

Figure 19: 最终模型的损失曲线

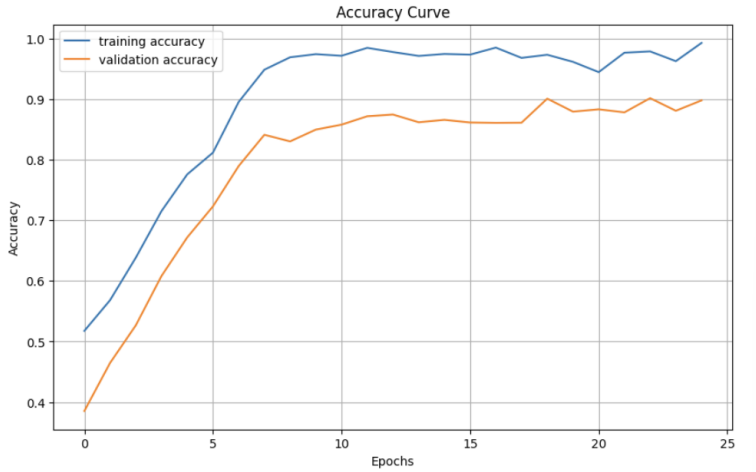

Figure 20: 最终模型的准确率曲线

4.4 性能分析

最终使用数据集的测试集进行性能评估。通过 model.evaluate 得到以下结果，由于调参过程不够细致，准确率欠佳，但是整体上来说模型表现良好。

Loss值	Accuracy值
0.213	0.897

Figure 21: 在测试集上的性能表现

5 总结

5.1 主要发现

在本项目中，我们成功开发了基于 DistilBERT 模型的情感分析系统，并使用 Yelp 数据集进行了微调。该项目表明，即使使用像 DistilBERT 这样的小型模型，我们也能够在文本分类和情感分析任务中实现高准确率。通过对嵌入和位置编码技术的改进，我们增强了模型对评论中细微上下文信息的理解，从而提高了情感分类的性能。

5.2 创新点及其潜在影响

本项目的一个关键创新是通过拼接的方式将原始嵌入与位置编码相结合，而不是传统的相加。这一方法使模型能够保留更细致的语义和位置细节，特别是在处理长且复杂的评论时，显著提升了预测的准确性。此外，我们还关注数据增强技术，如同义词替换和拼写噪声模拟，进一步提高了模型在噪声环境中的鲁棒性。这不仅增强了模型在不同数据集上的泛化能力，也突显了在现实应用中，数据往往不完美时部署该解决方案的潜力。

这些创新点强调了结合新技术与现有模型的重要价值，从而推动性能的边界，同时保持效率。本项目的见解和方法有潜力影响未来的情感分析及其他自然语言处理任务的研究与应用，尤其是在计算资源有限但仍需高准确率的场景中。

List of Figures

1 Knowledge Distillation 实际应用 2
2 DistilBERT 的参数与其他模型的对比 [1] 3
3 distilbert 的基本架构 3
4 DistilBERT retains 97% of BERT performance [1] 4
5 DistilBERT is significantly smaller while being constantly faster. [1] 4
6 DistilBERT yields to comparable performance on downstream tasks. [1] 4
7 模型架构的创新点体现 6
8 distilbert 模型的配置信息 7
9 Yelp/yelp_revi_full 数据集 8
10 SetFit/sst5 数据集 8
11 加载数据集 数据预处理 9
12 重写源代码中的 TFEmbedding 9
13 相加变为串联 9
14 从相加改为串联操作 添加线性层 10
15 重写源代码中的 TFFFN 10
16 FFN 变化 10
17 替换掉源代码中的对应类 11
18 模型初期的损失曲线 11
19 最终模型的损失曲线 12
20 最终模型的准确率曲线 12
21 在测试集上的性能表现 13

References

[1] “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter".” https://arxiv.org/abs/1910.01108. Accessed:2 Oct 2019.