通过预测表情符号理解文本情感

Ethan Gordon Nishank Kuppa Rigved Tummala Sriram Anasuri
College of Computing, Georgia Institute of Technology
Atlanta, GA 30332
{ egordon40, nkuppa3, rtummala6, sanasuri3 } @gatech.edu

Abstract

本项目使用四种深度学习架构探讨从短文本序列中预测表情符号：前馈网络、CNN、transformer 和 BERT。我们使用 TweetEval 数据集，并通过焦点损失和正则化技术来解决类别不平衡的问题。结果显示，由于 BERT 的预训练优势，其整体表现最佳，而 CNN 在罕见表情符号类别上表现出更高的效能。本研究显示了架构选择和超参数调优对于情感感知表情符号预测的重要性，有助于改善人机交互。

1. 介绍/背景/动机

1.1. 问题陈述和目标

该项目涉及构建各种深度学习模型，以能够有效预测与一段简短文本信息最匹配的表情符号。这个任务与情感分析密切相关，其目标是基于文本检测情感或情绪。

本项目的主要目标是训练机器学习模型以理解一连串词语背后的直接或间接含义，然后选择最适合的表情符号。一些表情符号的使用频率很高（如心形表情符号），而其他则比较偶尔（如圣诞树表情符号），这使得模型更难学习去预测那些不常见的表情符号。因此，挑战在于构建一个模型，该模型不仅仅是选择最常用的表情符号，而是学习将正确的表情符号匹配到每个独特的信息，以表达正确的情感。本项目的目标是：

1. 构建可以将短消息与表情符号以一种准确且类人方式匹配的模型。
2. 改进对不常使用的稀有表情符号的预测。
3. 比较不同的模型设计和超参数调整策略，以找出最佳方案。

1.2. 当前方法与限制

表情符号情感分类目前有多种方法：基于词典的方法、机器学习模型和基于变压器的模型。基于词典的方法通过给表情符号数据集分配预定义的情感分数，与机器学习方法相比较为基础且简单。它们基本上表现得像关键词查找系统，其中单词与表情符号进行匹配。近年来，经典的机器学习模型（如 SVM 和朴素贝叶斯）以及深度学习模型（如 CNN 和变压器）已被用于对大型数据集进行训练 [2] 。在移动键盘表情符号预测中，使用的是直接在设备上运行的轻量级机器学习模型，如 RNN [1] 。

即使采用这些先进的方法，许多问题仍然存在。这些模型在准确处理上下文时常常失败，尤其是在涉及讽刺或文化差异的时候。例如，哭泣的表情符号最近在社交媒体上被用来表达无法控制的笑声而不是悲伤，因此模型需要能够跟上同一表情符号的多重含义。总体而言，模型倾向于过度拟合常见模式，而较旧的基于词典的方法无法充分跟上表情符号使用的演变。总之，目前的方法在细微差别和真实的类似人类的表达方面仍显不足，本项目旨在解决这些问题。

1.3. 影响

此项目的影响范围包括帮助用户获得更准确和富有表现力的表情符号建议，并支持依赖于理解用户内容的消息应用和社交媒体平台的开发人员。如果表情符号预测变得更加准确和具备上下文感知，它可以通过更智能的建议改善用户体验，帮助平台更好地理解用户情感，用于内容审核或推荐。它还使系统能够适应新的表情符号趋势，为用户在数字通信平台上提供更个性化的体验。除便利性之外，这项研究表明，表情符号预测是评估情感分析架构的一个有效试验平台，提供的见解可以扩展到更广泛的自然语言处理应用中，其中情感的细微差别尤为重要。此项目的成功意味着缩小语言与情感之间的差距，同时推进我们对不同深度学习架构如何处理精确情感分类的理解，这对于以人为中心的人工智能非常有价值。

1.4. 数据集选择

本项目使用了来自 HuggingFace 的 Barbieri 等人的 TweetEval 表情符号预测数据集 。数据集简单地包括两列：第一列包含推文，第二列包含一个由 20 个表情符号之一表示的类别标签，这些表情符号反映了推文的情感。推文是本项目良好的文本表示，因为它们简短、非正式，并且通常具有直接的情感线索，使其成为研究人们在正常交流中如何将语言与表情符号配对的理想选择。

该数据集基于公开可用的推文，不需要进一步的预处理，因为数据的结构已经很简单且格式良好。用于本项目的数据集的表情符号配置包括 45,000 个训练样本、5,000 个验证样本和 50,000 个测试样本。本文介绍

的模型使用所有可用的训练样本进行训练。

该数据集存在类别不平衡，因为一些表情符号如红心比其他表情符号出现频率高得多。虽然这对准确预测比较少见的表情符号带来了挑战，但该数据集代表了现实世界和非正式语言，提供了多样化的数据来研究常见表情符号使用模式。

2. 方法

2.1. 设计选择与实施

为了开展该项目和探索各种方法，使用不同的架构创建了四个模型：BERT，前馈神经网络，transformer 和 CNN。评估四种截然不同的架构并比较其性能的方法在很多方面都是新颖的。首先，这些模型被选择来比较各个模型的不同优势：BERT 因其通过大规模预训练的迁移学习，前馈网络作为基准，CNN 用于学习局部语义模式，而 transformer 则通过注意机制来建模可能并不相邻的令牌之间的依赖关系。其次，本项目旨在探索由于某些表情符号主导数据集而导致的类别不平衡，值得观察哪个模型可以最好地处理这种不平衡并更准确地预测较少见的表情符号。需要注意的是，像前馈网络/transformer/CNN 这样的定制模型利用 pytorch 框架和 TweetTokenizer 来对我们的词语进行分词。

所有四种架构具有相同的初始设置：

1. 数据集加载：在导入相关库（例如 torch，torchtext，datasets 和 nltk）之后，将具有 “emoji” 配置的 TweetEval 数据集加载并分为训练、验证和测试集。
2. 分词和构建词汇表：使用来自 nltk 的预训练分词器，称为 TweetTokenizer，对推文进行分词。此分词器经过优化，特别适用于推文。之后，根据训练数据构建词汇表，并包括用于填充和未知词的特殊标记。
3. 编码：推文文本被编码为标记 ID，并进行填充或截断以符合固定长度 64。
4. 数据加载器：数据加载器为所有集合创建，以帮助在训练和测试期间进行批处理。

以下是对每种架构的更详细概述。将在第 3 节：实验和结果中给出理由。

- BERT: 我们使用一个经过预训练的 BERTweet 基础模型，为每个标记获取 768 维的上下文隐藏表示。然后，我们通过三种不同的注意力机制来处理这些表示：具有 8 个头的词级注意力、具有 4 个头的短语级注意力以及具有 2 个头的句子级注意力。添加了一个核大小为 3 的一维卷积层和自适应平均池化。每个注意力流经过掩蔽池化，以处理不同的序列长度并忽略填充。我们结合所有四个流，并使用一个融合层将 3072 维缩小到 768 维。应用层归一化、ReLU 和 dropout 后，我们将所有内容输入一个两层分类器，该分类器从 768 降至 384 到最终的表情符号类别，使用 ReLU 和 dropout。
- 前馈网络：这个网络首先使用嵌入层将输入标记映射到稠密向量。对序列应用最大池化以提取最显著的特征，将输入简化为固定大小的特征向量。然后将这个池化向量传递通过三个线性层，分别从 256 到 128 再到 64 维。使用 ReLU 激活、层归一化和 dropout 进行正则化。最终输出层会生成目标表情符号类别上的 logits。
- Transformer：该模型的起始部分类似于前馈网络，具有其嵌入层，并提供位置编码以保持标记的顺序。这些嵌入随后通过一个多层 Transformer 编码器，该编码器应用自注意力机制来建模标记之间可能的关系。我们选择使用 2 个 Transformer 编码器层，每层具有 2 个注意力头。在 Transformer 之后，我们在标记维度上应用最大池化操作，以获得固定长度的输出，并聚合每个标记的最强激活。然后将这些结果通过具有 128 维输入的全连接分类器和 ReLU 激活，以生成最终的表情符号分类。
- CNN: 所提出的卷积神经网络采用多核结构。其设计目的是捕捉情感基础的表情符号分类所需的不同 n-gram 模式。该实现使用三个并行卷积层，卷积核大小为 3、4 和 5，这允许进行三元组、四元组、五元组特征提取，并与推文文本的不同语义粒度级别相关联。对每个卷积输出应用全局最大池化，确保情感表达的平移不变性。关键超参数包括 128 维嵌入，每层 128 个滤波器，0.3 的 dropout，以及解决类别不平衡的采用平衡类别权重的焦点损失（gamma=1.5）。

2.2. 预期和遇到的问题

我们预期到在推文中会有不同令牌数量（即不同长度的推文）的挑战。为了克服这个挑战，我们简单地添加填充令牌来统一匹配我们模型的输入数据，特别是在处理批量数据时。在处理这个问题时，我们也预期到由于领域知识和其他工作的先前尝试，准确率可能有限，因为推文中经常包含非正式对话、词语使用不准确，以及由于文化变化而改变的句子上下文。我们还面临一个主要的类别不平衡问题，即:heart: 表情符号有 10,000+ 实例，而其他一些例如:heart_eyes: 和:joy: 有大约 4,500+ 实例，相比之下:grin: 这样的小表情符号仅出现 1153 次。因此，我们最初尝试使用常规的交叉熵损失，但发现结果不太理想，使用 focal loss 时性能和梯度流动要好得多。

3. 实验与结果

每种架构的成功通过标准分类指标来衡量，即：

- 准确性: 正确分类的推文占总数的百分比。虽然这是一个简单的指标，但在类别不平衡的数据集中（如 TweetEval 数据集）可能会产生误导。
- 损失：同时跟踪训练损失和验证损失。重要的是，验证时使用了 focal loss，因为它旨在通过关注更困难的样本来处理类别不平衡问题。

- 精确率、召回率和 F1-分数：这些指标提供了每个类性能的更具体视图。

3.1. BERT

我们的多尺度注意力模型使用了各种调试后的超参数。我们尝试了更高的丢弃值，但没有发现太多好处，因此我们在融合层选择了 0.3 的丢弃率，而分类器的丢弃率为 0.2，以帮助减少过拟合。我们使用专门为社交媒体文本设计的预训练 768 维 BERTweet 嵌入，由于内存限制，批量大小为 16。我们使用了 AdamW 优化器，其保守学习率为 2e-5，权重衰减为 0.01，因为对于微调预训练模型，AdamW 比标准 Adam 提供了更好的正则化。我们还尝试了更高的权重衰减值和不同的学习率，但变化显示出很少的改善，因此我们坚持了我们的保守方法。最后，我们决定采用 3 个周期，梯度裁剪为 1.0，因为任何进一步的周期都会导致严重的过拟合。这种多尺度注意力架构以不同的粒度捕捉不同的语言模式。具有 8 头的词级注意力专注于精细的标记关系，需要更多的注意力容量。具有 4 头的短语级注意力处理局部块，识别出表明表情符号使用模式的词组合。使用两个头的句子级注意力机制捕捉了对整体情感和远程依赖的更广泛理解。核大小为 3 的一维卷积层提取了注意力机制可能遗漏的互补特征，特别适用于社交媒体文本中的序列模式。这导致了一个能够处理多层次语言复杂性的平衡架构。我们的模型评估注重准确率和加权 F-1 得分，以解决严重的类别不平衡问题。最终模型达到了 44% 的准确率和 0.45 的加权 F-1 得分，相较于基线方法有显著提升。模型在具有明显模式的表情符号上表现优异：心形表情符号（F-1：0.81）、圣诞树（F-1=0.71）和美国国旗表情符号（F-1=0.62），显示了情感、季节和政治背景标记的有效捕捉。然而，类别不平衡仍然具有挑战性，像眨眼吐舌头表情（F-1=0.11）和露齿笑的表情（F-1=0.11）等稀有类别基本被忽略。这表明虽然架构能捕捉到常见的情感模式，但在处理不太常见表情符号的微妙上下文差异时存在困难。我们的学习曲线显示，通过 3 个 epoch，训练损失从 2.43 下降到 1.77，而验证损失在第 1 个 epoch 后稳定在约 2.32。训练和验证损失之间的差距表明存在轻微的过拟合，但比典型的 transformer 架构更加可控。

3.2. 前馈网络

关于神经网络架构，三个从 256 到 128 再到 64 维度的线性层被证明是最佳设置，因为更多层会导致过拟合，而更少的层会导致欠拟合。最大池化通过提取序列中最具信息量的特征也被证明是有用的，从而有效地突出了每条推文中最重要的词汇。关于超参数，使用 128 的嵌入维度来表示每个标记，这为推文提供了足够的表达能力。为了减少过拟合，丢弃率设置为 0.3（实验表明这个值效果良好，而较低的值则几乎没有好处）。训练时使用的批量大小为 32，应用 Adam 优化器，采用 5e-4 的小学习率和 1e-4 的权重衰减来稳定学习过程，改进梯度更新，并将过拟合最小化。模型训练 10 个周期，因为实验表明再多的周期会导致严重过拟合。图 1 显示了前馈网络的学习曲线。曲线显示，尽管过拟合从第 6 个周期开始，但其程度很轻微。总体而言，验证损失接近于训练损失，并且两条曲线都呈下降趋势，因此学习过程是稳定的。实验表明，使用焦点损失对实现如此良好的学习曲线至关重要，因为使用类别加权交叉熵损失会导致训练损失和验证损失之间的差距大得多。这可能是因为焦点损失更适合于这种具有偏斜分布和多个稀有类别的数据集。表格 2 提供了该神经网络的分类报告。报告指出，该网络仅达到 28 % 的整体准确率和 0.28 的加权 F1 分数。类别不平衡显而易见，因为模型对:heart: 类别的预测明显过高。许多稀有类别，如 13、15 和 16，几乎被忽视，F1 分数接近于零。只有少数中等频率的类别表现较好，而类别 17 的 F1 分数为 0.61 脱颖而出。尽管使用焦点损失来缓解类别不平衡，但其他技术如随机过采样可能进一步缓解不平衡。

Label	Precision	Recall	F1-Score	Support
:heart:	0.91	0.73	0.81	10798
:heart_eyes:	0.44	0.24	0.31	4830
:joy:	0.56	0.43	0.49	4534
:two_hearts:	0.21	0.15	0.18	2605
:fire:	0.62	0.53	0.57	3716
:blush:	0.16	0.17	0.17	1613
:sunglasses:	0.23	0.20	0.21	1996
:sparkles:	0.32	0.36	0.34	2749
:blue_heart:	0.17	0.23	0.19	1549
:kiss:	0.15	0.32	0.20	1175
:camera:	0.33	0.51	0.40	1432
:flag-us:	0.58	0.68	0.62	1949
:sunny:	0.43	0.81	0.56	1265
:purple_heart:	0.11	0.17	0.14	1114
:wink:	0.13	0.18	0.15	1306
:100:	0.24	0.37	0.29	1244
:grin:	0.10	0.12	0.11	1153
:christmas_tree:	0.65	0.79	0.71	1545
:camera_with_flash:	0.37	0.27	0.31	2417
:stuck_out_tongue_winking_eye:	0.09	0.13	0.11	1010
Accuracy			0.44	50000
Macro Avg	0.34	0.37	0.34	50000
Weighted Avg	0.48	0.44	0.45	50000

Table 1. 表情符号分类报告（BERT 模型）

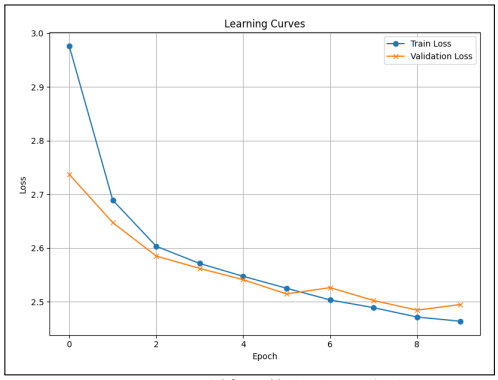

Figure 1. 前馈网络的学习曲线

Label	Precision	Recall	F1-Score	Support
:heart:	0.95	0.50	0.66	10798
:heart_eyes:	0.19	0.08	0.11	4830
:joy:	0.21	0.45	0.29	4534
:two_hearts:	0.17	0.18	0.17	2605
:fire:	0.22	0.37	0.28	3716
:blush:	0.08	0.02	0.03	1613
:sunglasses:	0.08	0.18	0.11	1996
:sparkles:	0.09	0.02	0.04	2749
:blue_heart:	0.11	0.03	0.04	1549
:kiss:	0.10	0.21	0.13	1175
:camera:	0.26	0.17	0.20	1432
:flag-us:	0.29	0.12	0.17	1949
:sunny:	0.21	0.67	0.32	1265
:purple_heart:	0.00	0.00	0.00	1114
:wink:	0.07	0.09	0.08	1306
:100:	0.02	0.00	0.00	1244
:grin:	0.05	0.00	0.00	1153
:christmas_tree:	0.57	0.64	0.61	1545
:camera_with_flash:	0.36	0.37	0.37	2417
:stuck_out_tongue_winking_eye:	0.04	0.13	0.06	1010
Accuracy			0.28	50000
Macro Avg	0.20	0.21	0.18	50000
Weighted Avg	0.35	0.28	0.28	50000

Table 2. 表情符号分类报告（前馈网络）

3.3. 变压器

Transformer 模型使用了各种调整后的超参数。我们决定使用 0.3 的 dropout 来尝试对抗过拟合，我们会在后面讨论；而较高的 dropout 值并没有显示出很大优势。类似于我们的前馈网络，我们选择了 128 的嵌入维度，以便赋予我们的 token 足够的特征表示。与前馈网络相似，我们选择了 32 的批量大小。为了进一步对抗过拟合，我们使用了 AdamW 优化器并配以 1e-4 的中等学习率和 4e-5 的权重衰减。与 Adam 优化器相比，使用 AdamW 优化器取得了略微更好的结果，因为权重更新与梯度脱钩，导致更好的泛化能力。我们还尝试了更高的权重衰减值以及更低/更高的学习率，但变化对对抗过拟合的改善效果不大。最后，我们定下了 15 个 epoch，但在验证数据损失上设置了一个提前停止检查，以尝试在训练过程中减少过拟合。

从表 3 中，我们看到整体准确率在 0.28 和 0.30 之间有小幅提升，宏观和加权 F1 分数也略微更好，这表明某些少数类上的表现有所改善。值得注意的是，诸如:purple_heart:、:fire:、:camera: 和:flag-us: 这些类在精度和召回率上显示出显著的提高，有些类如:purple_heart: 从零提升到了可测量的 F1 分数。然而，这些进步通常伴随着权衡，因为有些类（如:joy: 和:two_hearts:）的召回率和 F1 分数有所下降。总体而言，虽然从前馈架构转换到变压器编码架构确实带来了某些特定类别的改进，但单纯的架构改变在调优后并未显著提升整体性能。主要影响在于改进了之前表现不佳、F1 分数为 0 的类别的表现，但通常会略微牺牲其他类别的表现。

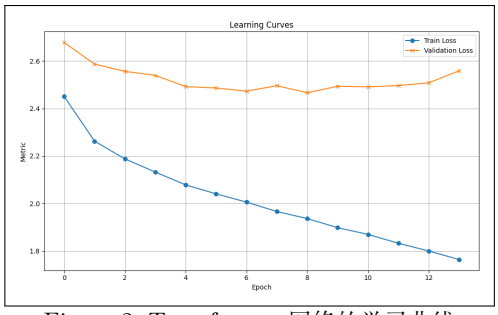

Figure 2. Transformer 网络的学习曲线

最后，我们查看了学习曲线，如图 2 所示，显示了训练/验证数据在历元中的损失。我们发现损失低于前馈网络的损失，但我们的训练/验证损失之间存在一个间隙，并且随着历元的增加而继续恶化。这表明我们的模型过拟合，因为验证损失保持稳定而训练损失继续减少。尽管进行了广泛的超参数调整以减少模型复杂性（减少 transformer 层数、减小嵌入维度、减少注意力头数）并调整学习率和权重衰减，但泛化间隙仍然存在。尽管如此，transformer 仍然表现出强大的性能。通过更好的正则化或数据增广来抵消过拟合和类别不平衡，transformer 可能会比前馈网络表现出更显著的优势。

Label	Precision	Recall	F1-Score	Support
:heart:	0.92	0.66	0.77	10798
:heart_eyes:	0.18	0.11	0.14	4830
:joy:	0.25	0.31	0.28	4534
:two_hearts:	0.17	0.06	0.09	2605
:fire:	0.46	0.28	0.35	3716
:blush:	0.09	0.03	0.05	1613
:sunglasses:	0.17	0.08	0.10	1996
:sparkles:	0.20	0.12	0.15	2749
:blue_heart:	0.09	0.11	0.10	1549
:kiss:	0.09	0.25	0.13	1175
:camera:	0.25	0.49	0.33	1432
:flag-us:	0.47	0.35	0.40	1949
:sunny:	0.26	0.43	0.33	1265
:purple_heart:	0.05	0.13	0.08	1114
:wink:	0.06	0.24	0.10	1306
:100:	0.08	0.08	0.08	1244
:grin:	0.05	0.10	0.07	1153
:christmas_tree:	0.51	0.70	0.59	1545
:camera_with_flash:	0.33	0.07	0.12	2417
:stuck_out_tongue_winking_eye:	0.04	0.08	0.05	1010
Accuracy			0.30	50000
Macro Avg	0.24	0.23	0.21	50000
Weighted Avg	0.38	0.30	0.32	50000

Table 3. 表情符号分类报告（Transformer 模型）

3.4. 卷积神经网络

模型初始过拟合，测试准确率为 35 %，训练准确率为 72 %，因此对模型进行了重新设计以获得更平衡的结果和更健康的训练模式。调整过程包括将学习率从 1e-4 加快到 1e-3，训练轮数增加到 20，但引入了早停机制，耐心参数设为 3（早于默认的 5）。然而，准确率会在验证数据上达到顶峰，然后在 5 个训练轮数后下降，同时训练准确率迅速上升。但由于早停机制的作用，训练在大约 8 个轮数时被中断，因此由于模型学习过快并开始过拟合，决定将训练轮数减少到 5。最终结

果是训练和测试的最终准确率非常平衡，分别为 31.35 %（训练）和 32.7 %（测试）。

我们的模型在关注准确性的同时，也专注于加权 F-1 分数，以考虑类别不平衡。最终的 CNN 整体准确率为 33 %，加权 F-1 分数为 0.34，相较于 transformer 和前馈基线略有改进。学习曲线 [3] 表明，在整个五个 epoch 中，模型进行了稳定的学习，训练损失保持一致。在第二个 epoch 之后，两个曲线之间有一个非常小的差距，提示过拟合，但两个曲线都呈下降趋势，验证了正则化策略的有效性。

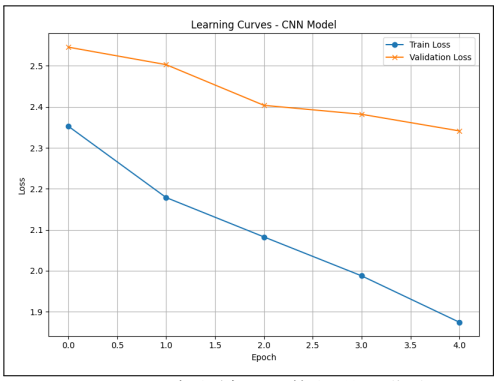

Figure 3. 卷积神经网络的学习曲线

对每个类别 [4] 结果的深入分析让我们洞察了模型的不同性能模式，更多地了解了模型的能力和局限性。对于包含明显词汇模式的表情符号，CNN 的预测表现非常出色——圣诞树的 F-1 得分为 0.64，表明模型捕捉到了季节性术语；美国国旗表情符号得分为 0.46，显示出对政治背景有很强的理解；火焰表情符号也表现不错，得分为 0.43，因为它能够识别现代俚语的使用。类别不平衡依然是一个持续的问题，数据集中有 22 % 的样本包含标准的心形表情符号，导致模型对该表情符号有很大的偏向（F-1=0.75）。这种不平衡导致了在语义上相似的表情符号上有问题，比如紫色心形（F-1=0.09）、双重心形（F-1=0.10）和蓝色心形（F-1=0.11），这告诉我们，虽然模型能够识别情感情绪，但在细微的上下文理解上却存在困难。尽管使用了焦点损失函数，不平衡也影响了不太常见的表情符号预测，像眨眼舌头表情符号和脸红表情符号（F-1 都为 0.07）几乎被忽略，因为缺少训练样本。这种性能差距反映在宏平均 F-1 得分 0.24 中，远低于加权平均值 0.34。结果表明，虽然卷积结构在预测具有明确语言信号的表情符号方面表现良好，但它们在捕获需要更深层次上下文理解的微妙情感差异方面仍存在困难，而不仅仅是局部 n-gram 模式。

这项综合研究分析了四种不同的深度学习架构用于从给定文本预测表情符号，深入了解了模型设计与情感分类任务表现之间的关系。在这四个模型中，BERT 表现出明显的最佳表现，准确率为 44 %，权重 F-1 分数为 0.45。该设计采用复杂的多尺度注意力机制以及预训练的社交媒体表示法，在具有明显模式的表情符号上表现出色，如心形 (F1=0.81) 和圣诞树 (F1=0.71)。

Label	Precision	Recall	F1-Score	Support
:heart:	0.96	0.62	0.75	10798
:heart_eyes:	0.27	0.06	0.10	4830
:joy:	0.37	0.26	0.31	4534
:two_hearts:	0.20	0.07	0.10	2605
:fire:	0.38	0.48	0.43	3716
:blush:	0.10	0.06	0.07	1613
:sunglasses:	0.10	0.18	0.13	1996
:sparkles:	0.25	0.19	0.21	2749
:blue_heart:	0.10	0.13	0.11	1549
:kiss:	0.12	0.21	0.15	1175
:camera:	0.23	0.62	0.34	1432
:flag-us:	0.42	0.50	0.46	1949
:sunny:	0.30	0.68	0.42	1265
:purple_heart:	0.06	0.18	0.09	1114
:wink:	0.07	0.14	0.10	1306
:100:	0.12	0.16	0.13	1244
:grin:	0.09	0.13	0.10	1153
:christmas_tree:	0.56	0.76	0.64	1545
:camera_with_flash:	0.27	0.05	0.09	2417
:stuck_out_tongue_winking_eye:	0.05	0.09	0.07	1010
Accuracy			0.33	50000
Macro Avg	0.25	0.28	0.24	50000
Weighted Avg	0.40	0.33	0.34	50000

Table 4. 表情符号分类报告（卷积网络）

在我们的实验过程中，所有模型遇到的另一个一致问题是由于数据集中样本数量的心形表情符号的主导地位导致的类别不平衡，无论我们实施了何种焦点损失或类别权重策略，都产生了偏差。所有模型在具有明显独特词汇模式的表情符号上表现出一致的强劲表现，而在语义相似的变体表情符号（例如，不同的心形变体）上表现较差，突显了当前方法在更精确的情感分类上的局限性。这些结果和分析可以用于人机交互和用户体验/用户界面实现，从智能手机键盘改进到社交媒体内容理解，其中 BERT 架构显示出最有潜在使用价值，而 CNN 紧随其后。研究证明，表情符号预测是评估模型架构在情感分析中的有效测试方法，清晰地表明设计必须与任务特征和数据属性相一致，以实现最佳性能。未来的工作可以探索更好的数据增强、对比学习以及结合不同优势的混合模型，以解决稀有类别和语义相似性所带来的持续问题。

References

[1] Francoise Beaufays, Kanishka Rao, Rajiv Mathews, and Swaroop Ramaswamy. Federated learning for emoji prediction in a mobile keyboard. https://arxiv.org/abs/1906.04329, 2019. 1

[2] Dev Priy Kulshreshtha, Upasna, Shivam Kumar Tiwari, and Aditya Dayal Tyagi. Emoji sentiment understanding through feature-based machine learning methods. International Journal of Creative Research Thoughts, 13(4):787–792, 2025. 1