用于领域无关多标签分类的标签语义感知生成方法

Subhendu Khatuya, Shashwat Naidu, Saptarshi Ghosh, Pawan Goyal, Niloy Ganguly
Indian Institute of Technology Kharagpur, India
subha.cse143@gmail.com, shashwatnaidu07@gmail.com,
saptarshi@cse.iitkgp.ac.in, pawang@cse.iitkgp.ac.in, niloy@cse.iitkgp.ac.in

Abstract

文本数据的大量增长使得手动文档分类变得越来越困难。为了解决这个问题，我们引入了一种强大、高效的、领域无关的生成模型框架，用于多标签文本分类。我们的方法并不是将标签作为单纯的原子符号处理，而是利用预定义的标签描述，并通过输入文本生成这些描述进行训练。在推理过程中，生成的描述通过微调的句子变换器与预定义的标签进行匹配。我们将这一过程与双目标损失函数相结合，结合交叉熵损失和生成句子与预定义目标描述的余弦相似度，确保语义对齐和准确性。我们提出的模型 LAGAMC 因其对多种数据集的参数效率和多功能性而突出，使其非常适合实际应用。我们通过在所有评估的数据集中实现新的先进性能，超过几个强大基线，证明了我们提出模型的有效性。与所有数据集中的最接近基线相比，我们在 Micro-F1 中实现了 13.94% 的改进，在 Macro-F1 中实现了 24.85% 的改进。

1 介绍

文本分类自动化了对大型数据集的分析和组织，使高效的标记、分类和获取有价值的见解成为可能。在文本分类中，主要有两个类别：多类分类和多标签分类。多类分类为文本分配一个单一的类别，而多标签分类（MLTC）(Xiao et al., 2019) 则为一个文档分配多个相关的标签。现实世界的应用包括主题识别 (Yang et al., 2016) 、问答系统 (Kumar et al., 2016) 、情感分析 (Cambria et al., 2014) 、信息检索 (Gopal and Yang, 2010) 和文本分类 (Schapire and Singer, 2000) 等。

在多标签文本分类领域，包括传统机器学习、深度学习和混合模型在内的多种方法已被提出 (Chen et al., 2022) 。一些最新的前沿方法 (Huang et al., 2021; Xiao et al., 2019) 强调标签描述或元数据在提高模型性能和捕捉标签相关性方面的重要性。然而，这些方法通常在其通用性和适应性上面临限制。有些模型依赖于标签层次结构或元路径图 (Ye et al., 2024) ，尽管在某些情况下很有效，但却限制了可扩展性和灵活性。此外，一些模型被设计用于特定任务，例如法律或金融应用，为专用分类器整合标签描述 (Chalkidis et al., 2020; Khatuya et al., 2024) 。尽管这些方法有了进步，但仍局限于特定领域，这突显了需要一种更具通用性的方法来实现 MLTC。

大型语言模型（LLM）通过其广泛的预训练，能够基于文本模式理解相似性和关系 (Naveed et al., 2024) 。这促使我们提出了一种新颖的、领域无关的流水线，利用最近的生成模型，在参数高效的设置中采用低秩适应（LoRA）(Hu et al., 2021) 进行 MLTC 任务。不像之前将标签描述视为元数据的方法，我们提出的方法 LAGAMC 训练生成模型直接生成这些描述。这样我们就能够充分利用 LLM 的潜力，提供对标签关系及其与相应文本连接的更细致的表示。

为多标签数据集手动获取标签描述既费力又具有主观性。为了克服这一点，我们通过利用 GPT-3.5 (Brown et al., 2020) 和维基百科来自动化这个过程，从而能够在最小的人为干预下高效创建标签描述。通过在提供标签的语义信息的同时利用生成模型，我们的方法始终优于现有的最先进模型，在 Micro-F1 上取得了 9.32 % 的提升，并在 Macro-F1 上取得了 15.25 % 的提升，超过了最接近的基线。

我们进一步整合了一个双目标损失函数，将交叉熵损失与基于余弦相似度的损失结合在一起。这使得 Micro-F1 和 Macro-F1 分别进一步增加了 4.62 % 和 9.60 %。使用这种混合损失有助于在表示空间中使生成模型输出和预定义标签描述的嵌入更接近，从而更容易将生成的输出与最终标签映射。这种混合方法确保模型理解标签的区别，并避免对标记级匹配的过拟合，从而在各种数据集上提高性能。

对社交媒体、新闻、学术和医疗等不同领域的数据集的实证评估显示了我们方法的有效性 LAGAMC。在各种数据集中，我们的方法始终优于最先进的模型，与最接近的基线相比，在 Micro-F1 上实现了整体提升 13.94 %，

在 Macro-F1 上提升了 24.85 %。此外，我们的模型在处理稀有标签时表现出色，并展示了零样本能力，进一步增强了其实用性。总之，本文的主要贡献如下：

- 我们使用由 GPT-3.5 (Brown et al., 2020) 和维基百科指导的注释过程为缺乏其的数据集生成了标签描述。我们的数据集 [1] 和代码可通过这个匿名链接 [2] 公开获取。
- 我们开发了一种新颖的参数高效的生成方法用于 MLTC，利用标签描述来提高分类准确性。
- 我们引入了一个双目标损失函数，结合了一个基于语义相似度的损失来增强模型的语义理解。
- 我们的方法在不同领域中具有良好的泛化能力，这由各种数据集上的性能指标所证明。对于所有数据集，我们提出的模型 LAGAMC 相比基线取得了巨大的改进。此外，我们的模型在稀有标签上表现出色，并展示了零样本能力。

在文献中，这项任务已经通过多种不同的技术方法来解决，从简单的单标签分类器方法到基于神经网络的模型，或者通过采用基于变压器的方法。解决这个问题的方法可以分为三类：问题的转换、算法的调整和基于神经网络的模型。这种问题的转换方法是算法无关的，即将多标签学习任务转化为单标签分类任务，但这在现实世界中拥有的数据量的情况下是计算上难以处理且不切实际的。第二种方法涉及通过修改某些特定的学习模型和算法来直接处理多标签数据。例如，使用在树的叶子上带有多个标签的决策树。基于神经网络的各种方法，如 CNN，RNN，CNN 和 RNN 的结合，注意力机制等，已经在文档表示中取得了巨大的成功。上述所有模型主要关注文档表示，并倾向于忽视标签语义和内容。最近，基于变压器的方法被用于解决这一任务。像 BERT、RoBERTa 这样的预训练变压器模型被用来建立一个基于神经网络的多标签文本分类方法。在文献中，作者提出在 RoBERTa 的最终层上构建多个注意力层以进行最终预测。一些方法，包括 DXML，EXAM，SGM，GILE 被提出以通过利用标签结构或标签内容来捕捉标签之间的相关性。

[1]数据：https://drive.google.com/drive/folders/1nrCKgmEtYrM1mQHIu3-eKXEtRdlIUDiO?usp=sharing

[2]代码：https://anonymous.4open.science/r/GenerativeMultiLabel_Classification-5415/README.md

2 相关工作

多标签文本分类已经通过多种技术来实现，包括扩展单标签分类器、采用神经网络架构，以及更近期的，利用基于变压器和预训练语言模型的方法。基于神经网络的方法表现出极大的成功，方法包括利用卷积神经网络（CNNs）(Liu et al., 2017)、循环神经网络（RNNs）(Liu et al., 2016) 以及混合的 CNN-RNN 模型 (Chen et al., 2017)。基于注意力的模型 (Yang et al., 2016; You et al., 2019; Adhikari et al., 2019) 也改善了文档表示，但常常忽视标签语义和依赖关系。

最近的方法利用了诸如 BERT (Devlin et al., 2019) 和 RoBERTa (Liu et al., 2019) 等预训练的 transformer，这些模型已经被调整以适应多标签任务。例如，(Ameer et al., 2023) 在 RoBERTa 的最后一层上应用多个注意力层，以增强标签关联学习。(Ma et al., 2023) 探索了各种设计用于减轻数据集中类别不平衡影响的损失函数。

大多数方法没有利用标签描述中包含的信息。一些利用标签描述来提升性能的工作 (Ye et al., 2024) 不具有可扩展性，因为它们需要额外的信息，如标签层次结构。此外，一些模型是为特定任务而构建的 (Khatuya et al., 2024)，这使得很难推广到来自不同领域的数据集。

还有一些流行的极端多标签分类框架，如 GalaXC (Saini et al., 2021)、SiameseXML (Dahiya et al., 2021a)、DEXA (Dahiya et al., 2023)、Renee (Jain et al., 2023)、InceptionXML (Kharbanda et al., 2024) 等。这些工作更多地关注于在大量标签中进行高效预测，但很少利用标签描述来提高性能。

3 数据集和标签描述生成

我们从社交媒体、新闻、学术和医疗领域收集了流行的多标签分类数据集用于我们的实验。

3.1 数据集概览

CAVES [社交媒体]：CAVES 数据集 (Poddar et al., 2022) 包含约 1 万条与 COVID-19 相关的反疫苗推文，并经过人工标注。

路透社 [新闻]：Reuters-21578 (Hayes and Weinstein, 1990) 包含来自 1987 年路透社新闻的文档，分布不均衡。

AAPD [学术文本]：Arxiv 学术论文数据集（AAPD）(Yang et al., 2018) 包含来自计算机科学领域的摘要。

SemEval [社交媒体]：SemEval-2018 任务 1C (Mohammad et al., 2018) 包含 2016-2017 年标注情感的英文推文。

PubMed [医疗保健]: 一个来自 BioASQ 9 任务 A [3] 的 PubMed 数据集的处理版本 [4]，可在 Kaggle [5] 上获得，并由生物医学专家手动用医学主题词（MeSH）进行标注。

3.2 通过标签描述增强数据集

在多标签分类任务中，许多现有的数据集仅提供最终的标签预测，而没有为每个标签提供详细描述。然而，标签描述对于改善上下文理解和提高模型性能至关重要。在本研究中，我们通过添加精炼的标签描述来弥补这一缺陷。在我们的研究中，来自社交媒体领域的 CAVES 数据集 (Poddar et al., 2022) 是唯一一个已经包含这些描述的数据集。

对于剩余的数据集，我们使用 GPT-3.5 (Brown et al., 2020) 结合预定义的维基百科定义生成标签描述。最初，我们从维基百科检索标签定义，称之为“初步描述”。为了更好地使这些定义与每个数据集的具体上下文对齐，我们通过提供初步的维基百科定义及数据集中预测结果中标签出现的两个相关例子来进行细化。然后提示模型为每个标签生成更符合上下文的描述，以确保更好地与数据集的上下文对齐。

例如，在 SemEval 数据集中，针对标签“Anger”使用了以下提示：
标签：愤怒
初步描述：愤怒，一种涉及恼怒和愤怒的情绪。
数据集：包含推文及其对应的情感注释。
数据集中的示例：
推文 1：“泪水和眼睛可以干涸，但我不会，我像灯泡里的电线一样燃烧。”
预测：愤怒
推文 2：“我们将在下一轮对曼城进行复仇。”
预测：愤怒
任务：为“愤怒”生成一个适合该数据集背景的标签描述。

Dataset	Train	Dev	Test	# Labels	Max Labels	Avg. Desc. Length
CAVES	6,957	987	1,977	12	3	28.17
SemEval	6,838	886	3,259	12	6	61.11
Reuters	6,769	1,000	3,019	90	11	13.41
AAPD	53,840	1,000	1,000	54	8	50.34
PubMed	40,000	10,000	10,000	14	13	91.40

Table 1: 数据集统计。最后三列显示了标签总数、每个样本的最大标签数以及每个数据集的平均标签描述长度。

评估指标：为了评估我们模型的性能，我们考虑了 1) Micro-F1 和 2) Macro-F1 指标。

[3] http://participantsarea.bioasq.org/general_information/Task9a/
[4] https://pubmed.ncbi.nlm.nih.gov/
[5] https://t.ly/FWWuQ

我们通过不同的基线验证我们的模型，这些基线从传统的基于 RNN 和 CNN 的方法，如 TextCNN (Kim, 2014)、TextRNN (Liu et al., 2016)、Attentive ConvNet (Yin and Schütze, 2018) 到基于 transformer 的方法，如 BERT (Devlin et al., 2019)、XLNet (Yang et al., 2020)、RoBERTa (Liu et al., 2019)、StarTransformer (Guo et al., 2022)。我们还将 LAGAMC 的性能与各种流行的极端多标签分类框架进行了比较，例如 (AttentionXML (You et al., 2019)、(GalaXC (Saini et al., 2021)、SiameseXML (Dahiya et al., 2021a)、DEXA (Dahiya et al., 2023)、DeepXML (Dahiya et al., 2021b) 和 Renee (Jain et al., 2023)。

我们还创建了我们自己的生成基线 T5-Base (Raffel et al., 2020) 和 T5-Large (Raffel et al., 2020)。最后，随着 ChatGPT (Brown et al., 2020) 的出现，我们很好奇地使用 gpt-3.5-turbo [6] API 在 500 个随机样本上使用相同的指令提示来检查它在该任务上的表现。

4 多标签分类的建议框架

所提出的 LAGAMC 框架用于多标签分类（图 1），分为两个阶段：监督生成阶段和无监督描述到标签匹配阶段。
问题表述：给定一个输入序列 $x_{input} = \{ x_1, x_2, ..., x_n \}$，任务是将文本分类标签 $Y_k = \{ y_1, y_2, ..., y_k \} \subset Y$ 分配给 x_{input}（待分类的文本），其中 $Y = \{ y_1, y_2, ..., y_p \}$ 包含该数据集的所有可能标签。我们采用基于提示的学习范式，生成在给定输入提示下生成的文本。

4.1 生成阶段

在第一阶段，我们将问题框定为一个生成任务，指导模型从给定文档中使用与任务相关的提示生成标签描述。我们构建一个包括三个组件的提示 x_{prompt}：指令（x_{inst}）：提供输入的简要概述（参见表 **??** 获取示例）。
任务描述（x_{desc}）：描述需要执行的确切任务。例如，在我们的工作中，当任务 x_{desc} 是 -
“任务：多标签文本分类
描述：为给定的文本生成标签描述。”
输入文本（x_{input}）：这是输入文本序列，在我们的情况下可以从论文摘要到推文。x_{prompt} 是通过将指令 x_{inst}、任务描述 x_{desc} 和输入文本 x_{input} 连接起来构建的。表 **??** 中列出了一个来自 SemEval 数据集的示例。

所提出的生成模型预计会生成一个文本回应 y_{target}，该回应是对应文本真实标签的预定义

[6] https://platform.openai.com/docs/models/gpt-3-5

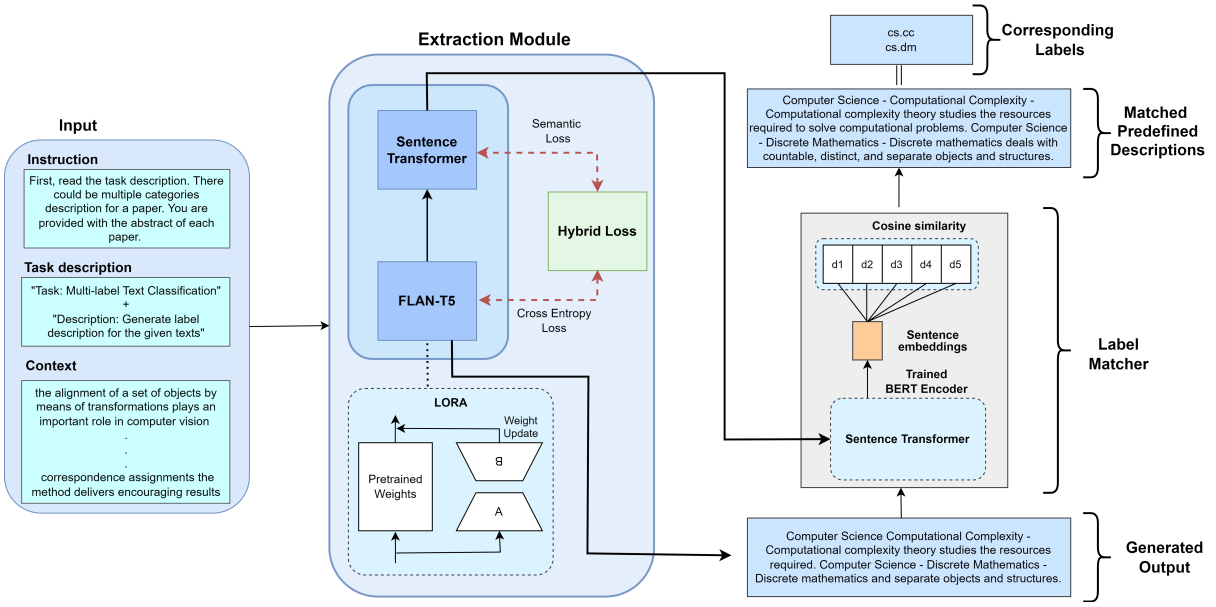

Figure 1: 我们提出的框架。提取模块将任务特定指令和要分类的输入文本作为输入。在该模块中，FLAN-T5 与句子转换器一起在双目标损失上进行训练。FLAN-T5 生成的标签描述随后进入标签匹配器，该匹配器使用训练过的句子转换器预测该文本的最终标签。

标签描述的串联。因此，如果预期输出具有 k 个标签 = { y_1 , y_2 , ..., y_k }，那么 y_{target} = { y'$_1$. y'$_2$ ..... y'$_k$ }，其中 y'$_i$ 表示 i^{th} 标签的预定义标签描述（通过停顿符串联和分隔）。y_{target} 的示例可以在表格 **??** 的目标列中看到。在这个生成阶段，我们为训练数据集中的每个数据点制定 x_{prompt} 和 y_{target} 。我们将这个 x_{prompt} 作为输入，并以 y_{target} 为目标提供给我们的模型。

在文本生成任务中，像 FLAN-T5 这样的模型通过交叉熵（CE）损失进行训练。交叉熵损失在词元级别上操作，这意味着它仅在序列中的每个位置奖励精确匹配。其主要限制在于无法处理使用不同词元的语义等价句子。因此，我们在训练生成模型时在损失函数中加入了一个基于语义相似度的项，以防止模型过度拟合于精确匹配。使用这种混合损失有助于在表示空间中将生成模型的输出嵌入和预定义标签描述的嵌入拉近，从而更容易将生成的输出映射到最终标签。我们将混合损失函数定义如下：

$$\mathcal{L}_{\text{hybrid}} = \lambda \cdot \mathcal{L}_{\text{CE}} + (1 - \lambda) \cdot \mathcal{L}_{\text{semantic}} \quad (1)$$

其中：$\mathcal{L}_{\text{CE}} = -\sum_{t=1}^{T} y_t \log(\hat{y}_t)$ 表示传统的交叉熵损失，该损失是在标记级别计算的。这里，y_t 是在位置 t 的真实标记，而 $\hat{y}_t$ 是在同一位置的标记的预测概率。

$\mathcal{L}_{\text{semantic}} = 1 - \text{CosSim}(v_{\text{gen}}, v_{\text{target}})$ 是语义相似性损失，其中 v_{gen} 和 v_{target} 分别表示由句子转换器为生成模型的生成输出和目标 y_{target} 生成的嵌入。$CosSim$ 表示生成和目标嵌入之间的余弦相似度。在学习过程中，允许句子转换器进行训练和调整。λ 是一个可学习参数，用于动态调整交叉熵损失和语义相似性损失之间的平衡。

4.2 标签匹配阶段

在推理过程中，我们采用标签匹配模块根据生成的描述和预定义描述之间的相似性来分配标签。我们利用生成阶段训练的句子转换器来获取生成句子 $\{gendesc_1, gendesc_2, \ldots, gendesc_k\}$ 和预定义标签描述的嵌入。对于每个生成的句子 $gendesc_i$ ，我们计算它与所有标签嵌入的余弦相似度，并选择相似度最高的标签作为最终预测 $predLabel_i$ 。即使生成的描述偏离了预定义标签，该方法也能确保强大的匹配。

为了评估生成方法在该任务中的可行性，我们对多种生成模型进行全面评估，它们在大小和训练策略上有所不同。首先，我们对 T5-Base（220M 参数）和 T5-Large（780M 参数）进行微调，以生成目标描述。接下来，我们微调 FLAN-T5 Large (Longpre et al., 2023; Chung et al., 2022) ，该模型受益于在超过 1.8K 个基于指令的任务上进行的广泛预训练。为了高效地进行微调，我们对所有生成模型应用低秩适应（LoRA） (Hu et al., 2021) ，仅更新模型参数的 0.08 %。我们的模型和基准的可训练参数的详细信息见表 2 。

5 实验设置

对于我们所有的模型变体（在 NVIDIA A100 80G GPU 上进行），我们从 Huggingface 库获取预训练检查点[7]。对于使用 LoRA 训练模型来说，可训练分解矩阵的秩设置为 2。FLAN-T5-Large 模型经过 20 个周期的指令调整，批量大小为 8，学习率为 $2e-4$，使用 LoRA（训练时间：56 分钟/周期，推理时间：2 分钟/样本）。这些超参数是基于验证集上最佳宏 F1 结果选择的。输入长度由每个数据集的平均输入标记数设置，而输出长度基于每个数据集的平均标签描述长度设置。

6 结果与讨论

我们在表格 2 中报告了我们提出的生成模型变体和各种基准的结果，涵盖所有数据集。首先，我们针对 MLTC 任务对预训练的 transformer 进行微调，比如 BERT、XLNet、RoBERTa，添加了一个投射层。接下来，我们与专门为 MLTC 任务设计的基准模型进行比较，比如 GalaXC、DeepXML、Renee 等。我们观察到在这些基准模型中，为 MLTC 任务专门设计的模型的性能优于微调的 transformer。为了检查我们提出的生成框架的鲁棒性，我们尝试使用不同的基础模型，比如 T5-Base、T5-Large。我们设计的生成基准在所有数据集上都优于最佳基准。我们还通过提供与 LAGAMC 相同的提示报告了 ChatGPT 的性能。

在平均情况下，与给定数据集的最佳基线相比，LAGAMC 在 Micro-F1 上提高了约 13.94 %，在 Macro-F1 上提高了约 24.85 %。性能最好的提升发生在 SemEval[8]，其次是 CAVES 数据集。LAGAMC 在从推文情感到医学领域数据集、学术文本的各个领域，与各自的 SOTA 模型相比，展示了其适应性和多功能性。表 2 的最后一列比较了不同基线包括我们训练的生成模型的可训练参数数量。我们表现最好的模型 LAGAMC 以及我们的生成基线，通过显著减少的可训练参数数量相较于大多数最接近的竞争基线，展示了参数效率。为了理解标签描述的重要性，我们进行了一个实验，把原子标签而不是它们的描述作为目标。因此，标签匹配模块现在比较生成的和真实标签的嵌入（而非描述）。从表 4 中，我们观察到 Macro-F1 平均下降了 36.04 %。我们通过比较我们提出的混合损失函数与标准交叉熵损失的性能，评估语义损失的重要性。结果总结在表 4 中，显示在只使用交叉熵损失时，Micro-F1 和 Macro-F1 分别下降了 3.96% 和 7.33% 。现在我们展示了 LAGAMC 的不同分析和消融。为了评估我们所提模型的零样本能力，我们构建了一个在训练中未见过标签的测试数据集。具体来说，从每个数据集中随机选择了 4-5 个标签作为未见标签，仅出现在测试实例中。然后，我们在修改后的数据集上训练模型，以评估其预测未见标签的能力。如图 2 所示，在全部训练的情况下，所有数据集的平均宏观 F1 得分为 83.45，而在零样本设置中为 70.61，表明在更具挑战性的零样本情景中仍表现出色。这减少了重新训练的需要，并加速了在实际环境中的部署。

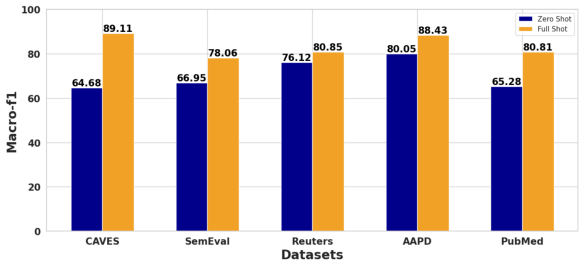

Figure 2: LAGAMC 的零样本性能，宏平均 F1 得分达到 70.61。

6.1 最不常见标签的表现

我们评估模型在最不常见标签上的表现，因为这对于现实应用至关重要，其中稀有标签可能代表重要事件。我们从每个数据集的训练集中识别出最不常见的 15 % 标签，并计算在测试样本中这些稀有标签集作为真实标签的 Macro F1 分数。如图 3 所示，我们的模型在所有数据集上均表现出优于最接近的基准模型的性能，Macro-f1 平均提高了 22 %。

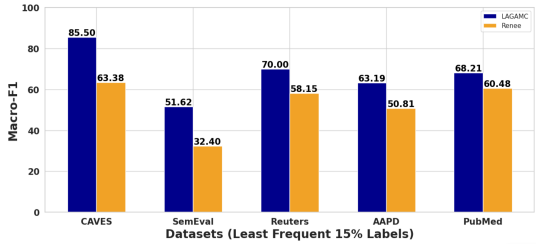

Figure 3: 对最不常见标签的模型性能比较：我们提出的模型相比于最接近的基线表现出更优越的性能。

我们还评估了最近的 LLM，例如 Llama-2-7b 和 Mistral-7B，用于多标签分类。表格中显示的初步结果是令人鼓舞的，因为这些模型优于所有基线方法（除了 PubMed 上的 Micro-F1 接近最佳基线）。然而，它们的准确性低于我们提出的使用 FLAN-T5 的方法。由于 GPU 资

[7] https://huggingface.co/

[8] 公共排行榜可在 https://paperswithcode.com/sota/emotion-classification-on-semeval-2018-task 查看

	CAVES		SemEval		Reuters		AAPD		PubMed		# TP(M)
Models	Mi-F1	M-F1	Mi-F1	M-F1	Mi-F1	M-F1	Mi-F1	M-F1	Mi-F1	M-F1	
Baselines											
BERT	70.36	65.29	70.70	56.30	87.73	34.98	71.30	55.90	85.05	70.99	110
XLNet	71.61	63.83	58.01	35.31	88.54	51.99	70.07	58.39	85.33	70.81	110
RoBERTa	71.34	63.82	59.82	40.55	88.27	42.63	69.14	54.88	85.19	70.54	125
TextCNN	55.48	39.64	54.55	39.51	81.89	33.96	67.71	49.85	82.62	66.4	3.88
TextRNN	57.70	42.17	52.42	37.94	81.72	33.26	69.28	52.27	83.11	67.72	3.86
StarTransformer	53.86	35.98	51.42	38.96	80.22	36.39	68.22	49.36	82.35	67.35	3.84
AttentiveConvNet	54.22	38.15	51.61	37.21	79.77	31.86	68.11	49.21	82.65	66.33	3.91
ChatGPT	57.22	44.47	41.61	27.21	69.77	35.86	48.11	29.21	42.65	26.33	-
AttentionXML	67.12	52.83	61.55	48.32	78.43	43.57	69.01	56.28	84.39	69.11	112
GalaXC	69.84	56.12	62.34	50.79	81.23	46.39	70.65	58.17	85.91	71.34	41
SiameseXML	72.16	59.89	65.42	52.84	84.22	48.99	72.48	60.43	86.42	72.98	115
DEXA	74.81	62.35	66.57	54.32	86.07	51.12	74.21	63.12	87.25	74.22	134
DeepXML	77.53	65.84	68.76	55.98	88.52	53.79	75.63	64.58	88.41	75.96	161
Renee	79.46	67.93	71.18	58.43	90.29	66.21	77.82	66.04	89.74	78.12	82
Proposed											
T5-Base	86.84	71.22	83.12	70.13	90.89	69.23	82.25	71.22	89.87	78.56	22.32
T5-Large	88.33	81.89	84.35	72.23	91.12	71.32	84.23	71.59	89.90	79.35	22.68
LAGAMC	92.46	89.11	87.81	78.06	96.48	80.85	95.64	88.43	89.93	80.81	22.69

Table 2: 基于多个数据集的 Micro F1 和 Macro F1 分数的性能评估。最佳性能以粗体显示，最强的基线结果以下划线显示。最后一列 TP (M) 表示大约的可训练参数数量（以百万为单位）。

源有限，我们无法完全微调这些模型或进行广泛的 LoRA 微调超参数搜索，这可能导致性能较低。我们也期望增加上下文长度可以改善结果。这些发现表明我们的方法是有效的，并可以应用于其他 LLM 进行多标签分类。我们还用 Llama-3.1-7B 评估了我们的方法，并观察到在 Micro-F1 和 Macro-F1 中分别提高了近 1 和 2 个百分点。

为了评估我们方法的稳健性，我们通过从每个数据集中随机选择 500 个训练样本和 100 个测试样本来创建了一个统一的数据集。所有数据集的平均 Macro-F1 得分为 83.45，相比于混合数据集的 78.38 分。尽管从 181 个真实标签中进行预测（表 1 中的标签列总和），我们的模型在性能上仅下降了 5.07%，而最接近的基线下降了 14.3%。

描述长度与性能：我们根据标签描述的串联长度来评估模型的性能。为此，我们将标签描述分组到具有相同测试样本数量的桶中。较长的描述对应更多的实际标签，增加了预测的复杂性。Figures 4 和 5 显示对于非常长的描述，性能略有下降，这一趋势在不同数据集之间一致。

实际与预测标签数量：对于每个数据集，我们分析具有给定标签数量的样本的数量，并将其与预测具有相同标签数量的样本数量进行比较（表格 5）。我们分析了最多五个标签数量。该模型倾向于为 SemEval 和 Caves 数据集提供单

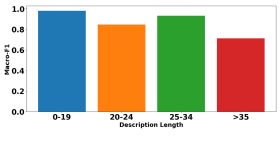

Figure 4: 洞穴

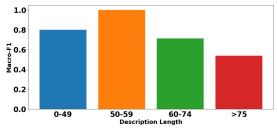

Figure 5: 语义评估

Figure 6: 标签描述长度的影响

标签预测。

我们研究了模块的计算效率，并提出了一种基于阈值的方法，以防止标签分配来自幻觉文本。

计算效率：Label Matcher 模块（第 4.2 节）通过计算句子和标签嵌入之间的余弦相似度来分配标签。使用 NumPy 的 (Harris et al., 2020) 矩阵运算进行并行计算可显著提高效率。例如，对于 10,000 个句子和 1,000 个标签使用 1,024 维的嵌入，矩阵方法在 0.089 秒内完成，而顺序方法则需要 0.354 秒。在最坏情况下，当所有 1,000 个标签都出现在一个实例中时，推理仅需 0.007 秒（矩阵方法）而顺序方法为 0.043 秒。

预测中的幻觉：在标签匹配过程中，每个输

Dataset	Model	Mi-F1 / M-F1
CAVES	Ours	92.46 / 89.11
	Ours with threshold	91.46 / 88.55
	Llama-2-7B with threshold	90.97 / 87.13
	Llama-2-7B w/o threshold	89.67 / 83.65
	Mistral-7B with threshold	90.17 / 86.52
	Mistral-7B w/o threshold	88.59 / 82.25
SemEval	Ours	87.81 / 78.06
	Ours with threshold	86.18 / 77.60
	Llama-2-7B with threshold	86.54 / 77.11
	Llama-2-7B w/o threshold	84.67 / 75.65
	Mistral-7B with threshold	85.54 / 76.25
	Mistral-7B w/o threshold	84.45 / 74.45
Reuters	Ours	96.48 / 80.85
	Ours with threshold	94.48 / 76.15
	Llama-2-7B with threshold	93.62 / 74.65
	Llama-2-7B w/o threshold	92.18 / 72.88
	Mistral-7B with threshold	95.17 / 76.43
	Mistral-7B w/o threshold	93.55 / 74.15
AAPD	Ours	95.64 / 88.43
	Ours with threshold	94.46 / 86.73
	Llama-2-7B with threshold	93.46 / 86.11
	Llama-2-7B w/o threshold	89.12 / 75.97
	Mistral-7B with threshold	92.76 / 86.07
	Mistral-7B w/o threshold	87.57 / 74.15
PubMed	Ours	89.93 / 80.81
	Ours with threshold	89.22 / 78.91
	Llama-2-7B with threshold	89.92 / 79.71
	Llama-2-7B w/o threshold	86.75 / 77.45
	Mistral-7B with threshold	89.67 / 79.55
	Mistral-7B w/o threshold	87.65 / 78.35

Table 3: 使用不同的 LLM 在多个数据集上的性能比较（Mi-F1 / M-F1）。

Dataset	Models	Mi-F1 / M-F1
CAVES	Ours	92.46 / 89.11
	w/o Semantic loss	89.67 / 85.25
	w/o Label Description	67.63 / 62.98
SemEval	Ours	87.81 / 78.06
	w/o Semantic loss	85.53 / 74.79
	w/o Label Description	64.24 / 54.35
Reuters	Ours	96.48 / 80.85
	w/o Semantic loss	94.97 / 78.12
	w/o Label Description	68.24 / 43.12
AAPD	Ours	95.64 / 88.43
	w/o Semantic loss	86.94 / 73.13
	w/o Label Description	65.27 / 53.28
PubMed	Ours	89.93 / 80.81
	w/o Semantic loss	86.77 / 74.12
	w/o Label Description	67.29 / 53.21

Table 4: 跨数据集进行有、没有标签描述和没有语义损失的性能比较。结果突出显示了标签描述和语义损失对于多标签分类的重要性。

Labels	Caves	SemEval	Reuters	AAPD	PUBMED
1	(1386, 1562)	(288, 456)	(2592, 2583)	(-, -)	(-, -)
2	(579, 369)	(1486, 1367)	(279, 308)	(642, 690)	(50, 105)
3	(12, 46)	(1078, 1055)	(86, 63)	(264, 225)	(376, 535)
4	(-, -)	(395, 316)	(32, 32)	(69, 62)	(1438, 1503)
5	(-, -)	(11, 60)	(17, 15)	(23, 21)	(2200, 2404)

Table 5: 基于标签数量的实际样本计数和预测样本计数的比较。每个单元格 (x, y) 表示特定标签计数下的实际样本数 (x) 和预测样本数 (y)。

出句子根据余弦相似度被分配到最近的标签。然而，LLM 生成的文本可能包含虚构的句子，导致错误的预测。为了缓解这一问题，我们规定了一个最低相似度阈值，确保只有当句子的最高相似度分数超过设定值时才被分配标签。我们的分析发现，0.4 的阈值是最优的。如表 3 所示，这略微降低了我们模型（基于 FLAN-T5）的性能，但提高了较大 LLM 如 Llama-2 和 Mistral 的结果，可能是因为它们倾向于生成更长的输出。

6.2 模型组件的消融研究

我们对最佳模型进行消融实验以评估模块的重要性。对于标签匹配器，用 Sentence-T5-xxl 或 Sentence-BERT-L12 替换经过微调的 Sentence-BERT-Transformer 会降低性能（表 6 ）。类似地，在没有任务特定指令的情况下，对 FLAN-T5-Large 进行指令微调会导致所有数据集上的性能下降，突出了指令对齐的重要性。

我们通过将标签描述整合到两个强大的基线上（BERT 和 DeepXML）中进行了额外的实验。对于基于 BERT 的多标签分类基线，我们采用了一种联合编码策略，即输入文本和标签描述都使用共享的 BERT 编码器进行编码。这使得模型能够学习标签语义和文本表征之间的交互。

对于支持元数据合并的 DeepXML，我们引入了标签描述作为辅助特征。此外，在聚类阶段，我们用相应的描述替换了标签名称，以根据语义内容影响标签划分。

我们在 CAVES 和 SemEval 数据集上评估了这些修改后的模型。结果如表格 7 所示。我们观察到，结合标签描述与最初版本的 BERT 和 DeepXML 相比，性能有了一定的一致但适中的提升。然而，我们提出的带有标签匹配模块的生成方法取得了显著更好的性能，证明了将标签语义整合到预测过程中的设计优势。

为了描述我们模型所犯的错误，我们检查模型何时预测错误标签或提供了部分真实标签。我们注意到，我们的模型有时会在处理复杂输入时遇到困难。表格 8 中提到的第一个例子是关于一家处理黄金的公司相关的新闻，但新闻摘要是关于该公司的收购。这让我们的模型感到困惑，因为尽管" 黄金" 一词被多次提及，但新闻的主要主题是关于收购而不是关于黄金商品。

有时由于输入上下文的限制，我们的模型可能无法准确预测所有相应的标签。相反，它倾向于预测标签的子集。我们的模型通过准确区分相关的标签（如 “喜悦”、“爱” 和 “乐观”）而优于最佳基准模型，这些标签常常同时出现。

Model	CAVES (M-F1)	SemEval (M-F1)	Reuters (M-F1)	AAPD (M-F1)	PubMed (M-F1)
Ours	89.11	78.06	80.85	88.43	80.81
w S-BERT-L12	82.80	74.30	58.60	71.40	74.10
w ST5-xxl	83.20	74.70	56.70	71.40	75.00
w/o Instruction	81.30	74.80	56.40	71.60	74.30

Table 6: 消融研究结果：LAGAMC 及其变体的 M-F1 分数。

虽然基准模型对约 33% 的样本误判为“喜悦爱乐观”，但我们的模型能正确预测所有这样的样本。一个这样的例子展示在表 8 的最后一行。

7 结论

在这项工作中，我们提出了一种参数高效的生成方法，该方法配备了双重损失目标以解决多标签分类的挑战性问题。我们的方法引入了一种新颖的、与领域无关的框架，足够灵活，可以适应各种应用。通过利用生成建模和判别监督，该方法有效地捕捉标签间的相关性并增强预测的鲁棒性。通过大量实验，我们将我们的方法与几种专门为该任务设计的最新模型和强基线系统进行了比较。我们的结果表明，LAGAMC 获得了显著的性能提升，显示了其在多个评估指标上的优越性。此外，我们进行了详细的消融研究和实证分析，以验证框架中每个组件的贡献。

我们提出的框架的一个限制是，这种方法依赖于标签描述的可用性，而这些描述可能并不总是易于获取，并需要在缺失时进行生成。此外，它尚未在极端多标签分类数据集上进行测试。

Model	CAVES Mi-F1 / M-F1	SemEval Mi-F1 / M-F1
BERT	70.36 / 65.29	70.70 / 56.30
BERT + Label Desc.	73.00 / 67.50	72.20 / 58.00
DeepXML	77.53 / 65.84	68.76 / 55.98
DeepXML + Label Desc.	79.50 / 67.20	70.12 / 57.30

Table 7: 有无标签描述的基线性能。

Dataset	Abstract	Ground Truth	Ours	Renee
Reuters	CRA SOLD FOREST GOLD FOR 76 MLN DLRS... It also owns an undeveloped gold project.	acq	acq, gold	acq, gold
CAVES	The covid vaccine is not a vaccine... the next round of manufactured flu.	conspiracy ineffective side-effect	conspiracy	side-effect
SemEval	I used to make the peanut butter energy balls all the time. My famjam loved them!	joy, love	joy, love	joy, love, optimism

Table 8: 展示我们的模型和最近基准（Renee）预测错误的例子。

References

Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. 2019. Docbert: Bert for document classification.

Iqra Ameer, Necva Bölücü, Muhammad Hammad Fahim Siddiqui, Burcu Can, Grigori Sidorov, and Alexander Gelbukh. 2023. Multi-label emotion classification in texts using transfer learning. *Expert Systems with Applications*, 213:118534.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Erik Cambria, Daniel Olsher, and Dheeraj Rajagopal. 2014. Senticnet 3: A common and common-sense knowledge base for cognition-driven sentiment analysis. volume 2.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Guibin Chen, Deheng Ye, Zhenchang Xing, Jieshan Chen, and Erik Cambria. 2017. Ensemble application of convolutional and recurrent neural networks for multi-label text categorization. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2377–2383.

Xiaolong Chen, Jieren Cheng, Jingxin Liu, Wenghang Xu, Shuai Hua, Zhu Tang, and Victor S Sheng. 2022. A survey of multi-label text classification based on deep learning. In *International Conference on Adaptive and Intelligent Systems*, pages 443–456. Springer.

Hyung Won Chung et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Kunal Dahiya, Ananye Agarwal, Deepak Saini, K Gururaj, Jian Jiao, Amit Singh, Sumeet Agarwal, Purushottam Kar, and Manik Varma. 2021a. Siamesexml: Siamese networks meet extreme classifiers with 100m labels. In *International conference on machine learning*, pages 2330–2340. PMLR.

Kunal Dahiya, Deepak Saini, Anshul Mittal, Ankush Shaw, Kushal Dave, Akshay Soni, Himanshu Jain,

Sumeet Agarwal, and Manik Varma. 2021b. Deepxml: A deep extreme multi-label learning framework applied to short text documents. In *Proceedings of the 14th ACM international conference on web search and data mining*, pages 31–39.

Kunal Dahiya, Sachin Yadav, Sushant Sondhi, Deepak Saini, Sonu Mehta, Jian Jiao, Sumeet Agarwal, Purushottam Kar, and Manik Varma. 2023. Deep encoders with auxiliary parameters for extreme classification. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 358–367.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Siddharth Gopal and Yiming Yang. 2010. Multilabel classification with meta-level features. *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*.

Qipeng Guo, Xipeng Qiu, Pengfei Liu, Yunfan Shao, Xiangyang Xue, and Zheng Zhang. 2022. Star-transformer.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature*, 585(7825):357–362.

Philip J. Hayes and Steven P. Weinstein. 1990. Construe/tis: A system for content-based indexing of a database of news stories. In *Conference on Innovative Applications of Artificial Intelligence*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Yi Huang, Buse Giledereli, Abdullatif Köksal, Arzucan Özgür, and Elif Ozkirimli. 2021. Balancing methods for multi-label text classification with long-tailed class distribution.

Vidit Jain, Jatin Prakash, Deepak Saini, Jian Jiao, Ramachandran Ramjee, and Manik Varma. 2023. Renee: End-to-end training of extreme classification models. *Proceedings of Machine Learning and Systems*, 5.

Siddhant Kharbanda, Atmadeep Banerjee, Devaansh Gupta, Akash Palrecha, and Rohit Babbar. 2024. Inceptionxml: A lightweight framework with synchronized negative sampling for short text extreme classification.

Subhendu Khatuya, Rajdeep Mukherjee, Akash Ghosh, Manjunath Hegde, Koustuv Dasgupta, Niloy Ganguly, Saptarshi Ghosh, and Pawan Goyal. 2024. Parameter-efficient instruction tuning of large language models for extreme financial numeral labelling. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7391–7403, Mexico City, Mexico. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification.

Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *International conference on machine learning*, pages 1378–1387. PMLR.

Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep learning for extreme multi-label text classification. *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Shayne Longpre et al. 2023. The flan collection: Designing data and methods for effective instruction tuning.

Long Ma, Zeye Sun, Jiawei Jiang, and Xuan Li. 2023. PAI at SemEval-2023 task 4: A general multi-label classification system with class-balanced loss function and ensemble module. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 256–261, Toronto, Canada. Association for Computational Linguistics.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. Semeval-2018 task 1: Affect in tweets. In *Proceedings of the 12th international workshop on semantic evaluation*, pages 1–17.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2024. A comprehensive overview of large language models.

Soham Poddar, Azlaan Mustafa Samad, Rajdeep Mukherjee, Niloy Ganguly, and Saptarshi Ghosh. 2022. Caves: A dataset to facilitate explainable classification and summarization of concerns towards

covid vaccines. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Colin Raffel et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*.

Deepak Saini, Arnav Kumar Jain, Kushal Dave, Jian Jiao, Amit Singh, Ruofei Zhang, and Manik Varma. 2021. Galaxc: Graph neural networks with label-wise attention for extreme classification. In *Proceedings of the Web Conference 2021*, pages 3733–3744.

Robert Schapire and Yoram Singer. 2000. Boostexter: A boosting-based system for text categorization. *Machine Learning - ML*, 39:135–168.

Lin Xiao, Xin Huang, Boli Chen, and Liping Jing. 2019. Label-specific document representation for multi-label text classification. In *Proceedings of the 2019 conference in EMNLP-IJCNLP*, pages 466–475.

Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. 2018. Sgm: Sequence generation model for multi-label classification.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2020. Xlnet: Generalized autoregressive pretraining for language understanding.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of NAACL: human language technologies*, pages 1480–1489.

Hui Ye, Rajshekhar Sunderraman, and Shihao Ji. 2024. Matchxml: An efficient text-label matching framework for extreme multi-label text classification.

Wenpeng Yin and Hinrich Schütze. 2018. Attentive convolution: Equipping cnns with rnn-style attention mechanisms. *Transactions of the Association for Computational Linguistics*, 6:687–702.

Ronghui You, Zihan Zhang, Ziye Wang, Suyang Dai, Hiroshi Mamitsuka, and Shanfeng Zhu. 2019. Attentionxml: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification. *Advances in neural information processing systems*, 32.