arXiv:2303.06586v1 [cs.IR] 12 Mar 2023

通过对比学习实现应用问题主动优先级排序

Moghis Fereidouni*、Adib Mosharrof*、Umar Farooq†、A.B. Siddique*
* 计算机科学系，肯塔基大学，† 独立研究员 电子邮箱：mfe261@uky.edu、amo304@g.uky.edu、ufarooq.cs@gmail.com、siddique@cs.uky.edu

摘要—移动应用商店会以用户评论的形式产生海量数据，这些数据是用户需求和情感的重要来源；借助此类评论，应用开发者可以主动处理其应用程序中的问题。然而，只有少量评论反映了普遍存在的问题和情感，因此有必要自动识别重要评论。遗憾的是，现有的大多数文本排序和流行度预测工作都侧重于可获取其他信号的社交情境，这使得这些工作在应用评论场景中难以奏效。在本文中，我们提出了一种新的框架，即PPrior，通过识别重要评论（即预测在给定时间窗口内会获得大量投票数的评论），实现应用问题主动优先级排序。由于与社交帖子不同，用户的社交网络特征不可用，因此预测获得大量投票数的评论颇具挑战性。此外，还存在类别不平衡问题，因为大量用户评论几乎不会获得投票。PPrior 采用预训练 T5 模型，并分三个阶段运行。第一阶段以自监督方式使预训练 T5 模型适应用户评论数据。第二阶段利用对比训练，学习用户评论的通用且任务无关表示。第三阶段使用半径邻居分类器进行最终预测。该阶段还使用 FAISS 索引，以实现可扩展性和高效搜索。为开展广泛实验，我们从 Google Play 获取了一个包含超过 210 万条用户评论的大型数据集。实验结果表明，与多种最先进的方法相比，所提出的框架具有良好效果。此外，PPrior 在预测重要评论方面的准确率可与经验丰富的应用开发者相媲美。

索引词—应用评论、应用分析、应用问题自动优先级排序。

一、引言

近几年，移动应用的普及率呈指数级增长。根据最近的一项研究，用户每天使用移动应用的时间超过四小时，美国 [1]。此类用户活动会产生有价值的用户反馈，而这些反馈通常会发布在 Google Play 和 Apple App Store 等应用商店中。除了发表评论外，这些分发渠道还允许用户点赞或点踩其他用户的评论。用户评论包含各种信息，包括应用程序存在的问题、改进应用程序的建议以及功能请求等。对于应用程序开发者而言，及时关注用户评论（即撰写评论回复）并解决用户的疑虑，对于保持竞争力和留住用户至关重要。尽管先前已有一些研究致力于自动评论回复生成 [12], [14],

978-1-6654-8045-1/22/$31.00 ©2022 IEEE

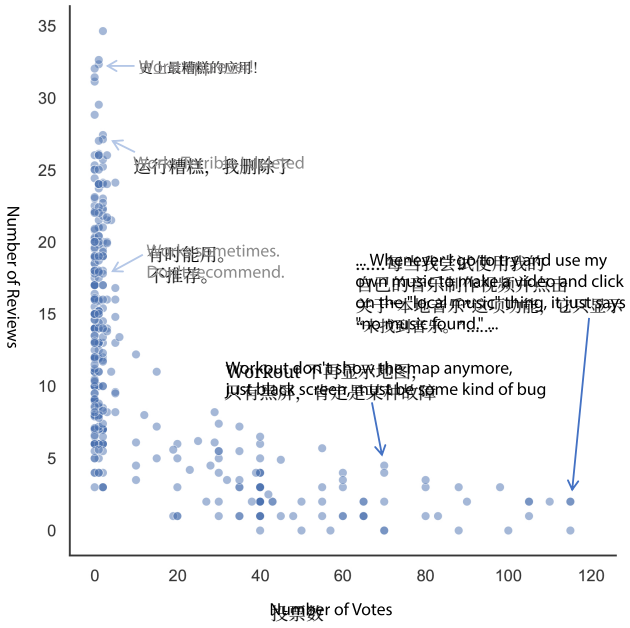

图 1：遵循幂律分布，少数用户评论获得了较高的投票数，而大多数评论几乎无人关注。高投票数的评论有助于应用程序开发者确定功能优先级并修复漏洞，因为这类评论通常会指出具体问题。

但从海量评论中识别关键用户反馈仍然是一项挑战。

用户评论有几个特点，使其难以分析。首先，鉴于用户评论数量庞大、增长迅速且内容繁杂，几乎不可能对其进行人工处理。其次，用户评论通常使用移动电话撰写，且往往包含噪声词语（例如拼写错误、重复内容、非英语词语）。现有研究主要关注评论预处理 [54],评论过滤 [5], 和主题分类 [11]。尽管如此，要从评论中自动识别关键应用问题仍然很困难，因为问题和用户关注点会随时间频繁变化，也因应用程序而异。

在本研究中，我们提出了一种新颖的框架 PPrior，通过自动预测给定评论将获得的投票数（即认为该评论有用的人数），主动对应用问题进行优先级排序，从而使应用开发者能够**主动**对这些应用问题进行优先级排序。PPrior 建立在以下几点认识之上。首先，许多

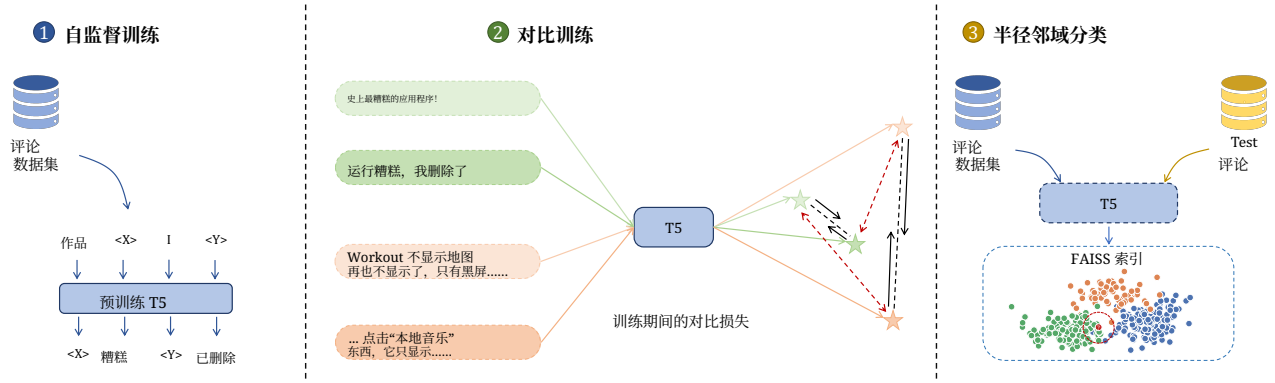

图 2：PPrior 概览 第二阶段的对比训练是所提出框架的关键组成部分 .

应用程序用户可能会遇到相似的问题并分享他们的担忧。其次，修复早期用户遇到的少数软件问题，可能就能为所有用户修复大多数漏洞和问题 [40]。第三，根据最大回报准则，应用开发者应优先处理大多数用户希望看到的功能或漏洞修复。对来自 Google Play 的用户评论大规模数据集子集进行的初步研究也证实了这些认识：少量评论会获得大量投票数，通常表达非常具体且明确的担忧，而许多用户都经历过这些问题。为突出这一现象，图 1 展示了部分用户评论。我们认为，鉴于这些问题会影响大量用户，开发者应优先解决这一小部分问题。图 1 还展示了一些几乎没有或完全没有获得投票数的评论样本。例如，在负面情绪的评论中，“最差的应用程序！”，开发者无法从这类评论中获得太多有用信息，因为它们没有指出应用程序存在的任何具体问题。本研究旨在准确预测一条负面用户评论在特定时间内将获得的投票数，以主动对这些问题进行优先级排序。也就是说，在应用问题导致大量用户感到沮丧并引发疲于应对的局面之前，主动解决这些问题。

我们将投票数预测任务定义为一个分类任务，这看似类似于预测社交媒体帖子的流行度，例如分别预测 Facebook 帖子的点赞数或 Twitter 推文的转发数。然而，事实证明，这些任务本质上有所不同，因为现有的社交媒体帖子流行度预测研究严重依赖用户的社交网络特征 [9], [23], [41] ，而应用商店不具备这些特征；在应用商店中，投票数完全取决于用户评论的内容，以及其他用户是否也有类似的看法，这使得该任务在实践中更具相关性、更有趣且更具挑战性。

我们提出的框架 PPrior 利用预训练语言模型、sentencetransformers [38], 并采用对比学习。图 2 展示了该框架的概览：框架分为三个阶段，并使用预训练 T5 [37] 作为骨干模型。语言模型（如 T5）以自监督方式在海量文本数据上进行训练，并拥有数百万个参数。因此，它们能够生成任何文本数据丰富且准确的表示。第一阶段采用预训练 T5，并进一步进行自监督训练。第二阶段采用对比训练目标。对比学习支持任务无关的训练，并促使模型学习一个通用的嵌入空间，使相似的用户评论彼此接近，而不相似的评论相互远离。投票数为零或较少的用户评论通常不会突出任何具体问题，因此最终会在高维嵌入空间中彼此靠近。此外，通用嵌入空间还能够应对评论中的分布偏移。为了进行高效的相似度搜索，我们使用 Facebook AI 相似性搜索（FAISS） [22] 存储训练样本，并在第三阶段使用半径邻居分类器——当数据不是均匀采样时，相较于经典 K近邻，这是更好的选择 [60] ——来预测给定用户评论的投票数。

为评估 PPrior 的有效性，我们首先抓取了一个数据集，其中包含来自约 10,000 个 Google Play 应用程序的超过 210 万条负面用户评论 [17]。由于用户评论获得的投票数也取决于评论发布后经过的时间长短，因此我们还记录了每条评论在发布一个月后获得的投票数。现有用户评论数据集 [12], [14] 未涵盖大量应用程序。此外，这些数据集没有记录评论在发布固定时间后获得的投票数。我们使用该大型数据集进行了广泛的实验评估，并与采用多种策略来解决类别不平衡问题 [26]–[51]的若干最先进的 Transformer 架构 [4], [10], [37], [50]进行了比较。我们还与用于预测社交媒体帖文受欢迎程度的先进方法 [9], [41]进行了比较。结果表明，在二分类和多分类任务中，PPrior 的马修斯相关系数（MCC） [7] 得分分别至少比所有竞争方法高出 27.97% 和 24.50%。此外，我们的人工研究表明，PPrior 在识别关键应用问题方面的性能与经验丰富的软件开发人员相当。源代码和数据集已公开提供[1]。

[1]https://github.com/MultifacetedNLP/PPrior

总而言之，本文的贡献如下：

- 我们发布了一个大型数据集，其中包含来自 GooglePlay、涵盖约 10,000 个应用程序的超过 210 万条负面用户评论，以及每条评论在一个月内获得的投票数。
- 我们提出了一个新颖的框架，用于预测特定用户评论的投票数，从而实现应用问题主动优先级排序。
- 我们利用大型数据集开展了广泛的实验，并将 PPrior 与最先进的方法进行了比较。自动评测指标和真实用户研究证实了 PPrior 的竞争力及其显著改进。

本文其余部分的组织结构如下：我们在第二节介绍预备知识，在第三节介绍 PPRIOR，随后在第四节介绍实验设置，并在第五节展示评估结果。最后，我们在第六节讨论相关工作，并在第七节总结全文。

II. 预备知识

我们提出的框架 PPrior 建立在自然语言处理（NLP）文献中的若干成熟组件之上。下面将对这些组件进行简要概述。

无监督表示学习与预训练语言模型。无监督（或自监督）潜在表示学习 [48]和预训练语言模型为近年来自然语言处理的成功作出了巨大贡献 [10], [35],[36], [50], ，其中包括促进零样本学习 [47], [49]。由于文本数据十分丰富，无监督学习技术推动了更加稳健的自然语言处理系统的发展。我们需要使用无监督表示学习技术优化所提出的框架，以稳健地处理分布偏移 [45], [46] ，从而准确应对新的、未见过的应用问题。语言模型（例如 GPT-2 [35], BERT [10], T5 [37]）利用海量文本数据，通过无监督技术进行训练。由于模型规模和训练数据集的规模较大，这些模型能够有效捕捉一般语义信息和句法信息。在本研究中，我们采用 T5 这一预训练语言模型作为框架中的骨干模型。我们还利用 Sentence-BERT [39]架构的变体来学习句子级表示。我们没有使用 BERT，而是采用功能更强大的预训练语言模型 T5 来学习句子嵌入。T5模型遵循 transformers [53] 架构，同时包含编码器和解码器堆栈，并在多个自然语言处理基准测试中取得了最先进的结果，包括 SuperGLUE。

T5模型。 受将每项自然语言处理任务都转化为问答任务这一理念的启发， [31], T5 将每项任务都视为一个文本到文本问题。借助文本到文本框架，几乎所有任务都可以使用相同的模型、目标、训练过程和解码过程来处理。T5 针对多任务学习调整了原始的 Transformer架构，因为它同时具有编码器堆栈和解码器堆栈；相比之下，BERT（即只有编码器堆栈，更适合自然语言理解任务）和 GPT（即只有解码器堆栈，更适合自然语言生成任务）则不具备这一结构。Transformer架构主要基于自注意力模块 [6]——这是一种注意力变体，其中序列中的每个元素都由序列中其余元素的加权平均值替代 [2], [18]。T5 使用 SentencePiece [24]将文本编码为 WordPiece词元 [43] ，并调整了受掩码语言建模和 BERT 中词丢弃技术 [3] 启发的去噪与跨度破坏目标，从而训练模型预测输入文本中的缺失跨度（或遭破坏的跨度）。图 2 展示了 T5模型输入和输出的示例。对于示例文本“Works Terrible I deleted”，在用唯一的单个哨兵词元“<X>”和“<Y>”替换被丢弃的跨度后，输入变为“Works <X>I <Y>”。随后，目标是仅预测哨兵词元，所有其他输入词元都被哨兵词元“<X>”和“<Y>”替换。在我们的示例中，输出变为“<X> Terrible <Y> deleted”。与预测目标中的全部内容相比，仅预测哨兵词元的计算成本较低。T5模型在海量清洁爬取语料库（C4）数据集上进行训练，该数据集包含约 750 GB 的数据。C4 是 2019年4月从网络中提取的文本的更清洁版本。在这项工作中，我们在框架中采用 T5 作为骨干模型，并分三个阶段对其进行微调，以捕获能够稳健迁移到测试数据集的通用模式。我们选择使用 T5，主要是因为其多任务学习能力，以及其在分布偏移场景中进行有效学习的鲁棒性。第一阶段，我们执行自监督训练，使模型适应评论数据集；第二阶段，进一步使用对比损失对其进行训练；最后，利用半径邻居分类器对评论做出最终预测。

FAISS 和 KNN。 Faiss库可高效地对稠密向量执行相似度搜索和聚类，并支持多种比较操作，包括 L2距离、点积和余弦相似度。通过使用分层可导航小世界（HNSW） [29]和导航式扩散图（NSG） [13],等可扩展方法在原始向量之上添加索引结构，FAISS 能够有效搜索数十亿个向量。在实现方面，它主要使用 C++ 实现，仅依赖 BLAS。借助 CUDA 支持的 GPU（包括单 GPU 和多 GPU）索引，还可以进一步加快推理速度。由于提供了 Python接口，它与所有深度学习框架兼容。在本文中，我们使用 FAISS 为所提出的 PPrior 框架第三阶段的训练集建立索引。随后，我们使用半径邻居分类器进行预测。半径邻居分类器通过选取给定半径内邻居中最常见的标签来进行预测。由于我们的评论数据集存在类别不平衡，相较于经典 KNN算法，这种方法更为适合。此外，第三阶段不需要任何训练，因此可以根据推理时的数据分布插入其他算法。在实验中，我们还使用加权KNN作为替代方案。

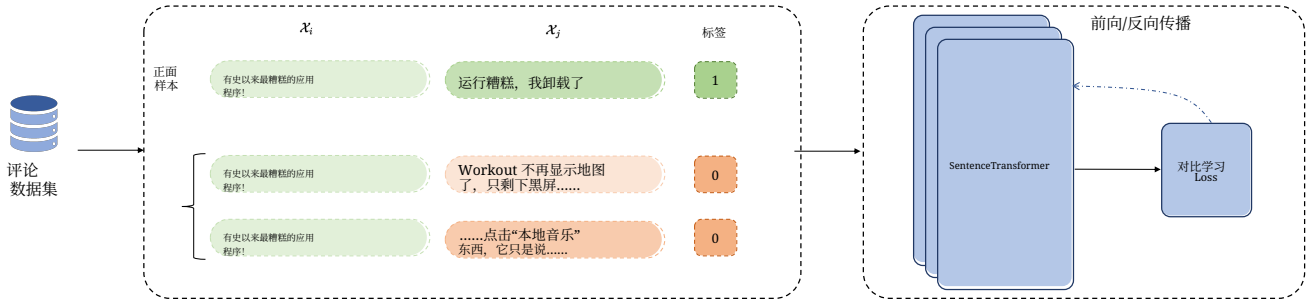

图 3：PPrior框架的第二阶段。对比训练能够学习用户评论的通用表示，使相似评论彼此保持接近。这些稳健且通用的表示让诸如 K近邻之类的简单方法能够轻松识别关键的用户诉求。

III. P主动 P优先级排序 F框架：PPrior

在本文中，我们提出了 PPrior，这是一种通过自动预测给定评论将获得的投票数来主动确定应用问题优先级的新型框架，从而使应用开发者能够主动确定这些应用问题的优先级。用户评论获得的投票数直接反映了认为该评论有用的人数。PPrior 以预训练 T5模型作为骨干模型，并分为三个阶段运行。图 2 展示了该框架的概览。第一阶段加载预训练 T5，并使用原始的去噪和跨度扰动目标，在评论数据集上进一步执行自监督训练。由于 T5 是在 C4数据集上预训练的，而该数据集与应用评论数据存在根本差异，因此这一步克服了数据分布偏移问题。第二阶段采用对比训练目标，以促进任务无关的通用表示学习。由于获得零票或少量投票的用户评论占据数据集的主导地位，因此存在类别不平衡问题。第二阶段的训练还通过生成负样本，并将高投票数样本与低投票数样本配对来应对这一挑战（即让少数类更频繁地呈现给模型）。因此，类别不平衡的影响得以最小化。此外，通用嵌入空间还解决了用户评论中问题频繁变化的问题。框架的第三阶段不需要任何训练，而是使用第二阶段学习到的表示执行推理。该阶段使用 FAISS 将大规模用户评论训练数据存储在可扩展的索引中，并执行高效的相似度搜索，以进行最终预测。该阶段使用半径邻居分类器，已有研究表明，在数据非均匀采样时，该分类器表现更好。由于该阶段不需要任何训练，因此可以根据测试集的数据分布动态替换预测算法。下面我们将进一步介绍 PPrior 的各个阶段。

A. 第一阶段：自监督训练

我们通过使用预训练语言模型的强大能力，在框架的第一阶段初始化时采用预训练 T5模型。我们的实现使用了包含2.2亿个参数的t5-base。该模型在 C4数据集（750 GB 的英语文本数据）上经过 524,288 步预训练，词片段词汇量为 32,000。编码器和解码器均包含 12 个模块，每个模块包含自注意力（具有 12 个注意力头）、编码器/解码器注意力以及一个前馈层。嵌入层具有 768 维。此外，该模型还使用概率为 0.1 的随机失活进行正则化。

第一阶段通过加载预训练的 t5-base，并利用用户评论数据集进行自监督训练，开始 PPrior 的训练，以使模型适应用户评论的分布。该阶段不改变原始的训练目标（T5模型的目标），即随机破坏输入跨度，并尝试预测被掩码的跨度。我们随机抽取并丢弃 15% 的词元。连续被丢弃的词元跨度会被唯一的特殊词元替换，这些特殊词元称为哨兵 ID。我们使用交叉熵损失和最大似然，并采用教师强制 [55]及 AdaFactor 优化器 [44]。训练过程中使用贪心解码，在每个时间步选择概率最高的词元。自监督训练阶段也展示在图 2（左侧）中。第一阶段的输出是经过训练的 T5模型，该模型已适应用户评论。

B. 第二阶段：对比训练

该阶段以经过用户评论数据适配的 T5模型（即第一阶段的输出）为初始化，以处理网络数据（T5模型基于网络数据进行预训练）与可能包含半技术性语言的用户评论数据之间的分布偏移。随后，我们使用对比目标进一步微调 T5模型。图 3 展示了对比训练的概览。为了在第二阶段训练 PPrior，我们利用负采样来捕捉用户评论之间的对应关系。也就是说，将获得相似投票数的用户评论的高维表示彼此拉近，并将所获投票数差距巨大的用户评论的表示相互推远。具体而言，我们为每个正样本对随机选择 K 个负样本。正样本对表示两个获得相似投票数的训练样本，

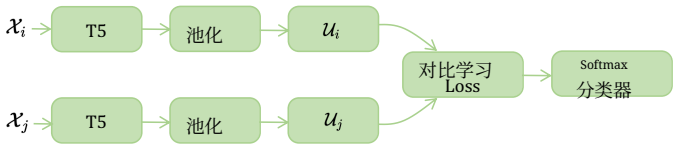

图 4：我们修改后的句子变换器使用 T5模型。

而负样本对包含两个投票数差异很大的样本。图 3 展示了该阶段训练数据的准备方式。我们利用对比损失来指导第二阶段的训练。我们的方法能够在用户评论之间建立具有通用迁移能力的关联，从而处理应用程序中新出现且未见过的问题。此外，负采样使模型能够接触少数类样本（例如，只有极少数用户评论能获得大量投票数），其程度与多数类实例相当。下文将进一步介绍训练过程的详细信息。

句子变换器。 句子变换器也称为 Sentence-BERT，旨在解决使用 BERT [10]和 RoBERTa [27]类架构时句对回归任务所涉及的计算开销问题。句子变换器通过使用孪生网络和三元组损失 [42]应对了这一挑战。在本研究中，我们将句子变换器与 T5 结合，用于基于对比学习的训练。我们修改了 Sentence-BERT 架构，将 BERT 模型替换为我们的 T5模型，以计算句子嵌入。图 4 展示了集成 T5 的改进版 Sentence-BERT 架构。

负采样。 假设 $X = \{\mathcal{X}_1, \mathcal{X}_2, \cdots, \mathcal{X}_n\}$表示用户评论，$y_i \in \mathbb{R}$ 表示第 i条评论 $\mathcal{X}_i$ 获得的投票数。给定两个训练样本$\mathcal{X}_i$和 $\mathcal{X}_j$，如果 $|y_i - y_j| < \lambda$（即正样本对），我们将标签设为 1。对于每个正样本对 $\mathcal{X}_i$ 和 $\mathcal{X}_j$，我们将 $\mathcal{X}_i$（或 $\mathcal{X}_j$，每次替换一个）替换为一个实例 $\mathcal{X}_q$，使得 $|y_i - y_q| \geq \lambda$，并将其标签设为 0（即负样本对）。对于每个正样本对，我们生成 K 个负样本对。也就是说，对于给定的两个样本，如果它们获得的投票数之差的绝对值小于 λ，则将标签设为 1，否则设为 0。在我们的实验中，我们采用 1:4 的负采样，并且在二分类和多分类任务中，λ 分别设为 100 和 4，同时增加以下约束：i）负样本对中的实例不应来自同一类别；ii）正样本对包含来自同一类别的两个实例。

输入嵌入。 令 $\mathcal{F}_\theta$ 表示我们修改后的句子变换器模型，并令（$\mathcal{X}_i$，$\mathcal{X}_j$）表示一个训练示例。我们分别使用以下方法获取 $\mathcal{X}_i$ 和 $\mathcal{X}_j$的表示 $\mathcal{U}_i$ 和 $\mathcal{U}_j$：

$$\mathcal{U}_i = \mathcal{F}_\theta(\mathcal{X}_i) \tag{1}$$

$$\mathcal{U}_j = \mathcal{F}_\theta(\mathcal{X}_j) \tag{2}$$

其中 θ 表示模型参数。

目标函数。 我们使用对比目标函数来微调句子级嵌入。对于给定的正训练样本（$\mathcal{X}_i$，$\mathcal{X}_j$）（标签为 1）和负训练样本（$\mathcal{X}_i$，$\mathcal{X}_q$）（标签为 0），对比损失会迫使正样本 $\mathcal{X}_i$ 和 $\mathcal{X}_j$ 在嵌入空间中彼此靠近，同时将负样本 $\mathcal{X}_i$和 $\mathcal{X}_q$推离。使用句子变换器获得正训练样本（$\mathcal{X}_i$，$\mathcal{X}_j$）的评论嵌入，分别为 $\mathcal{U}_i$ 和 $\mathcal{U}_j$，以及负训练样本（$\mathcal{X}_i$，$\mathcal{X}_q$）的评论嵌入，分别为 $\mathcal{U}_i$和 $\mathcal{U}_q$后，训练损失可定义为：

$$\mathcal{L}_{i,j} = -\log \frac{\exp\left(\mathcal{U}_i \cdot \mathcal{U}_j / \tau\right)}{\sum_{q \in Q} \exp\left(\mathcal{U}_i \cdot \mathcal{U}_q / \tau\right)} \tag{3}$$

其中，$\cdot$ 是相似度函数（例如点积），$\tau \in \mathcal{R}^+$ 是表示温度的标量参数，而 Q 是负样本对集合。

对新未见评论的泛化。 所提出的对比目标设计学习用户评论的通用表示，专注于学习评论之间的关联，使高亲和度的评论对在高维流形中相互靠近，而相似度较低的评论对彼此远离。通过这种方式，模型的目标转向学习通用功能，而非学习数据集特定分布及其相应特征。此外，T5的文本到文本框架与对比学习的通用特性支持多任务学习。在此阶段训练得到的T5模型可用于二分类、多分类或回归等一系列任务，而无需重新训练任何组件。

C. 第三阶段：半径邻域分类

所提出框架的第三阶段仅用于推理，不需要任何训练或微调。由于训练集规模庞大，我们使用 FAISS 对用户评论进行索引，以便可扩展且高效地检索相似评论，并利用这些评论进行预测。具体而言，我们将所有训练评论输入经过训练的 T5（即第二阶段的输出），并使用最后隐藏层的输出作为给定用户评论的嵌入向量。第 II 节介绍了 FAISS 的概况，感兴趣的读者可参阅 [22] 了解详情。第三阶段如图 2（右侧部分）所示。为了进行预测，我们使用半径邻域分类（RNC）算法，该算法利用给定半径内的邻居，并选择出现最频繁的标签作为预测结果。在实验评估中，我们针对二分类和多分类任务分别进行了实验。此外，我们还尝试使用加权KNN替代半径邻域分类器。我们将半径设为 2，并将最近邻数量设为 101。由于该阶段仅用于推理，因此可以根据速度需求和数据分布轻松更改索引机制和分类算法。

表 I：数据集统计信息。

	训练集	验证集	Test
评论日期	2021 年 10 月 1 日至 2022 年 1 月 31 日	2022 年 2 月 1 日至 2022 年 2 月 28 日	1 3月22日至3月31日
评论数量	1,176,261	422,267	570,588
每条评论的最小投票数	0	0	0
每条评论的最大投票数	44,178	50,476	47,798
每条评论的平均投票数	13.3	13.65	9.09
应用数量	9,572	8,350	8,369

IV. 实验设置

本节将介绍数据集详情、数据收集与预处理步骤、评估指标以及竞争方法。

A. 数据集

数据收集与预处理。 我们从 Google Play 中涵盖 48 个类别的 9869 个应用程序收集了超过 210 万条负面用户评论（即评分仅为 1 星和 2 星的评论）。这些评论的时间范围为 2021年10月1日至2022年3月31日。此外，我们还跟踪了每条评论在一个月后获得的投票数，因为用户评论获得的投票数也取决于其发布后的时长。例如，如果一条用户评论发布于 2021年11月1日，我们会记录其在 2021年12月1日的投票数。我们新数据集的独特之处在于，许多应用程序未被现有用户评论数据集覆盖 [12], [14],，并且评论并非在固定时间段后记录。此外，对于每条用户评论，我们还收集了其他有用的功能，例如应用名称、应用类别、该应用的评论总数、价格和内容分级等。用户评论中通常存在噪声 [15]。为了最大限度地减少数据集中的噪声，我们遵循了数据过滤方面的最佳实践 [12]。表 I 展示了有关数据集的重要统计信息，以及训练集、验证集和测试集划分的详细信息；图 5 展示了投票数（截断至 100）以及一个月内收到的评论数量的分布。

数据集分析。 基于对部分数据集（500 条用户评论）的人工分析，我们发现，少数评论获得了大量投票数。此外，我们注意到，获得大量投票的评论往往指出具体的应用程序问题，且这些问题可采取行动解决。这表明，许多应用程序用户面临相似（或相同）的应用问题，应遵循最大回报准则优先处理这些问题。另一方面，大量评论没有获得任何投票数。此外，这类评论并未包含任何可采取行动的投诉。图 1 展示了几个用户评论样本。

B. 评估

我们将该问题建模为二分类，同时也将其多分类。在二分类中，我们考察某条用户评论在一个月内获得的投票数是否超过 100 票。同样，在多分类任务中，我们考虑五个类别：0 票、1-5 票、6-25 票、26-100 票和 100+ 票。PPrior 的有效性

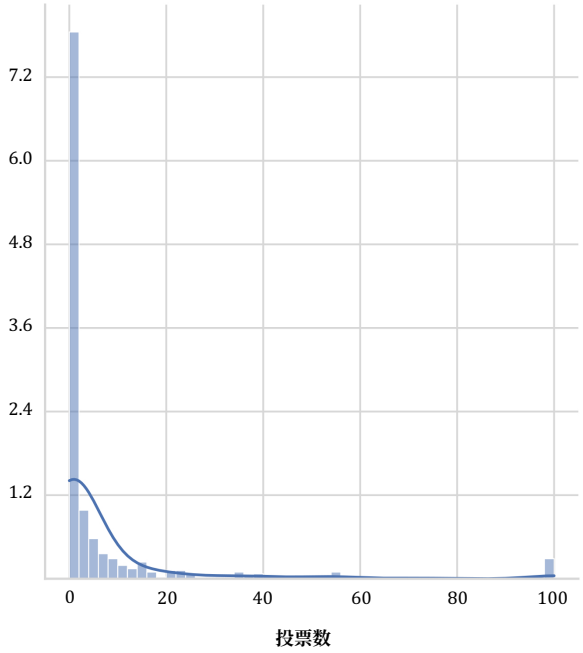

图 5：评论投票数的分布。

通过定量自动指标和主观人工研究进行评估。

自动评测指标。准确率、F1分数和MCC分数是分类任务定量评估的标准指标。在我们的评估中，我们使用宏平均F1分数，这对于类别不平衡任务而言是更好的指标。此外，我们还使用MCC分数，该指标已被证明更加稳健可靠。

人工研究。 为开展人工研究，我们招募了 10 名研究生，他们每人都至少拥有 4 年应用开发经验。我们按照相应投票数（上限为 100）进行加权，随机选取了 100 条用户评论，并将每条评论分配给三名不同的学生，请他们判断该评论是否指出了任何关键问题。

C. 竞争方法

我们使用大规模数据集开展了广泛的实验评估，并将其与多种最先进的 Transformer架构进行比较 [4], [10], [37], [50] ，这些架构采用了多种策略来应对类别不平衡问题。同样，我们还与最先进的社交媒-

体流行度预测模型 [23], [41]进行了比较。下面，我们将简要概述这些竞争模型。

BERT [10]: BERT遵循Transformer架构，仅使用编码器堆栈，被认为是许多自然语言处理任务中的先进模型。我们使用BERT的多个变体进行比较，例如采用二元交叉熵损失（BCE）和交叉熵损失（CE）的变体。此外，我们还考虑了类别不平衡策略，例如不平衡数据集采样器、Dice损失和焦点损失 [26], [51]。最后，我们还考虑了BERT的另一种变体：获取最后隐藏层的输出，并将其用作用户评论的表示。我们使用加权KNN进行预测，该方法根据邻居与目标样本之间的距离为邻居分配权重。在比较中，我们使用预训练的DistilBert模型。

GPT-2 [4]: GPT-2基于Transformer架构的解码器堆栈。我们在比较中使用distilgpt2。对于GPT-2，我们同样采用了用于BERT的所有策略。

T5 [37]: T5模型采用文本到文本架构（包含编码器和解码器堆栈）。与我们提出的框架相比，这个T5基线不采用对比训练，而是使用上述针对BERT和GPT-2提到的相同训练策略进行训练。为便于比较，我们使用PPrior中作为骨干模型的相同模型t5-base。这个基线也凸显了对比训练在我们提出的框架中的重要性。

MPNet [50]: MPNet结合掩码语言建模（例如BERT [10]）和排列语言建模（例如XLNet [57]）的优势进行预训练。为便于比较，我们也对MPNet采用上述所有训练策略。

EUSBoost [23]: 用于预测转发的最先进方法采用进化欠采样提升策略，并使用多种功能，包括用户网络、推文内容和时间等。

DTP [41]: 用于预测新闻流行度的最先进方法
依赖于初始推文行为和时间
特征 ·

V. 结果

本节将讨论定量结果以及人类研究的发现。此外，我们还将展示消融研究的结果。

自动评测指标。 表II展示了二分类任务的结果，在该任务中，我们考察给定用户评论是否会获得超过100票。每项指标中表现最佳的结果已用粗体标出。可以看到，采用加权KNN推理的PPrior在准确率指标上表现最佳。而采用半径邻域分类策略的PPrior在F1分数和MCC分数上表现最佳。MCC分数和宏平均F1分数都被认为是类别不平衡任务中更稳健的指标。需要强调的是，最佳

表 II：二分类任务的结果。

模型	方法	准确率	F1	MCC
BERT [10]	BCE 损失	0.5934	0.0612	0.1297
	不平衡采样器	0.7873	0.1059	0.1958
	Dice损失	0.8719	0.1451	0.2252
	焦点损失	0.9584	0.0958	0.0855
	KNN	0.5575	0.0592	0.1215
GPT-2 [4]	BCE 损失	0.6702	0.0981	0.1748
	不平衡采样器	0.8887	0.1750	0.2664
	Dice损失	0.9099	0.2230	0.2773
	焦点损失	0.8960	0.1033	0.0914
	KNN	0.5125	0.1125	0.2698
T5 [37]	BCE 损失	0.4705	0.0393	0.1026
	不平衡采样器	0.9405	0.2474	0.2443
	Dice损失	0.8078	0.1347	0.2127
	焦点损失	0.9394	0.0876	0.1022
	KNN	0.5642	0.0580	0.1295
MPNet [50]	BCE 损失	0.5276	0.0745	0.1205
	不平衡采样器	0.7163	0.1372	0.1887
	Dice损失	0.8217	0.1820	0.2525
	焦点损失	0.9511	0.1016	0.1166
	KNN	0.5298	0.1270	0.2446
EUSBoost [23]	欠采样	0.2664	0.0355	0.0661
DTP [41]	时间传播	0.2430	0.0395	0.0690
PPrior	KNN	**0.9798**	0.3201	0.3108
	RNC	0.9784	**0.3465**	**0.3409**

竞争方法中F1分数的最佳方法是采用不平衡数据集采样器的T5，其得分比PPrior低约10个百分点（0.2474对0.3465）。同样，采用Dice损失的GPT-2在竞争方法中取得了最佳MCC分数，但仍比我们提出的框架低6个多百分点。相较于采用Transformer架构的最先进模型，这种性能提升可归因于自监督训练和对比训练等训练策略。PPrior第一阶段中的自监督训练以无监督方式使模型适应数据集的新分布（即用户评论）。同样，第二阶段执行对比训练，该训练被证明能够有效应对类别不平衡挑战，并生成通用且准确的用户评论表示。此外，我们还注意到，对比训练策略能够抵御应用程序中不同问题之间持续的分布偏移，因为测试集中有超过200个应用程序未出现在训练集中。此外，我们还关注了用于社交媒体帖子热度预测的最先进模型ESUBoost和DTP的最差表现。缺少用户网络和时间交互等功能，可能是其表现不佳的原因。

表 III 展示了多分类任务的结果，其中我们将该任务设定为 5 类分类任务。考虑这一任务的主要原因是，开发者往往希望以 1 至 5 的尺度评估某条用户评论的严重程度，而不是进行二元预测。在多分类任务中，我们提出的 PPrior 在准确率、F1分数和 MCC分数这三个定量指标上均明显优于其他方法。具体而言，采用 K近邻的 PPrior 在准确率指标上表现最佳，比最佳竞争方法（即采用不平衡数据集采样器的 GPT-2）高出五个多百分点。同样，在

表 III：多分类任务的结果。

模型	方法	准确率	F1	MCC
BERT [10]	交叉熵损失	0.5377	0.2914	0.2355
	不平衡采样器	0.5543	0.3004	0.2651
	Dice损失	0.5660	0.3425	0.2392
	焦点损失	0.5618	0.3340	0.1212
	KNN	0.5062	0.2989	0.2232
GPT-2 [4]	交叉熵损失	0.4566	0.1985	0.1938
	不平衡采样器	0.6109	0.3512	0.2975
	Dice损失	0.5710	0.3438	0.2514
	焦点损失	0.5557	0.3115	0.1397
	KNN	0.5023	0.2886	0.2775
T5 [37]	交叉熵损失	0.4257	0.1980	0.1792
	不平衡采样器	0.5749	0.3549	0.2810
	Dice损失	0.5248	0.3161	0.2309
	焦点损失	0.5503	0.2963	0.1489
	KNN	0.5094	0.3029	0.2459
MPNet [50]	交叉熵损失	0.3591	0.1521	0.1357
	不平衡采样器	0.4899	0.2726	0.2058
	Dice损失	0.5152	0.2865	0.2285
	焦点损失	0.5928	0.3051	0.1795
	KNN	0.5167	0.3311	0.2507
EUSB [23]	欠采样	0.3121	0.1265	0.0913
DTP [41]	时间传播	0.3189	0.1115	0.1045
PPrior	KNN	**0.6687**	0.4211	0.3442
	RNC	0.6685	**0.4245**	**0.3455**

F1分数方面，采用 RNC 的 PPrior 比最佳竞争方法（采用不平衡数据采样器的 T5）高出 19.61%。如果考察 MCC分数，采用不平衡数据集采样器的 GPT-2 是最佳竞争方法，但其表现比采用 RNC 的 PPrior 低约五个百分点。我们还注意到，在竞争方法中没有明显的优胜者。然而，一种看似非常简单的不平衡数据集采样策略在大多数情况下往往效果更好。我们在第二阶段采用的对比训练策略在某种程度上模拟了不平衡数据集采样，通过负采样使模型更频繁地接触少数类实例，不过对比学习还具有其他优势。由于与二分类任务相同的原因，EUSB 和 DTP 等方法在多分类任务中的表现再次处于最差之列。我们还研究了骰子损失函数和焦点损失函数的影响，这些函数专门用于解决数据集中的类别不平衡问题。我们注意到，这些损失函数几乎总是优于交叉熵损失函数或二元交叉熵损失函数，这表明这些损失函数能够在一定程度上缓解类别不平衡问题。总而言之，从定量评估结果来看，PPrior 在两个任务的所有指标上都明显优于其他方法。

读者可能不会对 PPrior 在多分类任务中略高于 66% 的准确率印象深刻。我们还对此进行了深入研究，并在进一步的错误分析中发现，大多数错误都发生在类别边界附近。例如，分别获得 25 个投票数和 26 个投票数的评论彼此并无差异，但根据数据集中的类别划分边界，它们却属于不同的类别。此外，在我们的实验中，λ的值被设为四，因此不会对投票数差异小于四的评论进行对比。为克服这类情况，我们还尝试了更小的 λ值以解决这一问题，但总体性能有所下降。最后，我们进行了另一项分析，考察预测概率最高的两个类别中是否包含真实类别。在该实验中，PPrior 的准确率达到了 98% 以上。

消融研究。我们特别采用 T5 基线模型作为对比模型，以凸显所提出训练策略带来的改进。我们注意到，在所有指标上，无论是二分类还是多分类任务，PPrior 与 T5 基线模型的所有变体之间始终存在显著的性能差异。此外，我们还为所有 Transformer 模型引入了一个基于 KNN 的变体，以说明 PPrior 的性能提升并非源于现成的 KNN 实现。具体而言，我们使用所有模型最后隐藏层的输出，并进一步使用 KNN 进行推理。基于 T5 的实验，以及在所有对比模型中加入 KNN 组件的实验表明，PPrior 的性能提升主要归功于对比损失和自监督训练。

人工研究。 在人工研究中，对于给定的评论，当三名学生中至少有两名同意该评论指出了一个关键问题时，我们将其标签设为 1，否则设为 0，并将这些标签视为真实标签。在此过程中，我们最终发现 100 条用户评论中有 13 条涉及关键问题。为了比较 PPrior 与真实软件开发人员的表现，我们采用了二分类任务，考察 PPrior 是否能够预测给定评论会获得超过 100 个投票数。如果是，我们就将其视为关键问题。同样地，我们又招募了 10 名具有类似经验的研究生，请他们预测给定的用户评论是否属于关键问题。根据人工研究的实验结果，我们发现 PPrior 的表现可与经验丰富的应用开发者相媲美：在预测关键应用问题时，PPrior 在 100 次预测中仅出错 3 次，而人类仅出错 2 次。

VI. 相关工作

软件特性优先级排序。 近期研究通过利用用户反馈，研究了包括功能改进和漏洞修复在内的软件发布规划。Maalej 等人 [28]讨论了数据驱动需求工程的最新进展和未来方向。Ciurumelea 等人 [8] 开发了一个应用评论分类系统；此外，为了解决评论中提到的问题，他们推荐使用推荐系统来提出代码变更建议。Nayebi 和 Ruhe [32] 利用软件特性属性（如价值和凝聚性）来选择下一次发布规划中的功能。随后，他们提出了非对称发布规划（ARP） [33]，其中他们对非对称特性评估进行建模，并将发布规划表述为协同创建优化规划。最近，Zhang 等人 [58]研究了用户评论，并将软件特性优先级排序表述为一个优化问题，旨在最大化特定用户群体的应用评分。Yang 等人 [56] 使用一种半自动化

方法来确定待修复漏洞的优先级；他们研究了 6 个应用程序，并提取了十四个功能来对评论进行分类。

相比之下，我们的方法在比现有研究大得多的数据集上开展研究，并提出了一种利用用户的喜欢与不喜欢的新颖方式，而这一点被现有工作所忽视。此外，我们的工作还可与许多现有技术互补，例如，将 [8] 中的代码变更推荐与PPRIOR相结合，可以获得更多收益。

社交帖子热度预测。许多研究聚焦于社交媒体平台，以预测帖子的流行度并分析社交行为 [25]。Nesi等人 [34] 开发了一种预测模型，用于预测某条推文是否会被用户转发 [34]。Tsugawa和Kito [52] 利用用户的转发行为，并基于预测模型发现用户关系。Zhang等人 [59]收集了六个月的Twitter动态，以预测其与股市表现（包括道琼斯、纳斯达克和标普500）之间的相关性。Daga等人 [9]预测了Twitter帖子的点赞和转发。类似地，Khoerunnisa等人 [23]利用用户资料、内容和时间来预测转发。Saeed等人 [41] 根据Twitter上的初始推文行为预测新闻流行度。相比之下，PPRIOR专注于应用商店中的用户评论；与Facebook和Twitter等社交网络相比，这些评论不包含用户资料，而本工作弥补了这一差距。此外，我们还预测用户评论的喜爱度，这是现有工作尚未探索的内容。

应用评论分析。 许多现有研究从不同角度分析应用评论，包括应用功能请求 [19], 应用商店分析 [30], 开发者回复生成 [12], [14]等。Martin 等人 [30] 研究了应用商店，得出结论认为应用评论非常重要，并且包含有用的信息。MARA [21]利用应用评论预测用户的功能请求。Gu 和 Kim [19]开发了应用评论解析器，以了解用户对不同应用功能的意见。一项针对 4.5 百万条评论的研究 [20] 观察到了开发者回复应用评论的重要性。同样，一项 Google 研究 [16]强调，开发者回复评论能够带来更高的应用评分。PPRIOR 通过提供一个包含大量评论的数据集，并根据用户反应对应用问题进行优先级排序，扩展了现有研究的成果。

VII. 结论

我们提出了 PPRIOR，这是一种新颖的框架，能够自动确定应用问题的优先级，从而让开发者主动解决关键应用问题。PPRIOR 的提出源于这样一个现实：人们不可能快速处理数量庞大且持续增长的用户评论。借助所提出的 PPRIOR 的精准预测能力，它能够帮助应用开发者自动识别对用户满意度至关重要、需要优先解决的关键应用问题。我们提出的框架依托预训练语言模型和自监督学习策略的强大能力。框架第二阶段的对比训练生成了通用、精准且稳健的用户评论表示，并使 K近邻 等简单方法也能做出可靠的预测。

此外，我们利用 FAISS 库来加速推理。为了开展广泛的实验分析，我们从约 10,000 个 Google Play 应用程序中抓取了一个包含超过 210 万条负面用户评论的大规模数据集。广泛的实验评估表明，在广受认可的定量指标上，PPrior 始终优于多种 Transformer架构，包括 BERT、GPT、T5 和 MPNet。此外，与当前最先进的社交媒体帖子热度预测方法相比，我们的方法具有显著优势。具体而言，在 MCC分数 上，PPrior 至少比所有竞争方法高出 27.97% 和 24.50%。最后但同样重要的是，我们的人工研究表明，PPrior 的性能与资深应用开发者一样出色。

参考文献

[1] 每天在手机上花费的平均时间。 https://www.statista.com/statistics/1045353/mobile-device-daily-usage-time-in-the-us/。访问日期：2022-08-03。[2] Dzmitry Bahdanau、Kyunghyun Cho 和 Yoshua Bengio。通过联合学习对齐与翻译实现神经机器翻译。*arXiv* 预印本：*1409.0473*，2014年。[3] Samuel R Bowman、Luke Vilnis、Oriol Vinyals、Andrew M Dai、Rafal Jozefowicz 和 Samy Bengio。从连续空间生成句子。*arXiv* 预印本 *arXiv:1511.06349*，2015年。[4] Tom Brown、Benjamin Mann、Nick Ryder、Melanie Subbiah、Jared D Kaplan、Prafulla Dhariwal、Arvind Neelakantan、Pranav Shyam、Girish Sastry、Amanda Askell 等。语言模型是少样本学习器。神经信息处理系统进展，33 1901，2020年 1877：–。[5] Ning Chen、Jialiu Lin、Steven CH Hoi、Xiaokui Xiao 和 Boshen Zhang。Ar-miner：从移动应用市场中挖掘面向开发者的信息丰富的评论。载于第*36*届国际软件工程会议论文集，第767–778页，2014年。 [6] Jianpeng Cheng、Li Dong 和 Mirella Lapata。用于机器阅读的长短期记忆网络。*arXiv* 预印本 *arXiv:1601.06733*，2016年。[7] Davide Chicco 和 Giuseppe Jurman。在二分类评估中，马修斯相关系数（MCC）相较于 F1分数和准确率的优势。*BMC Genomics*，21(1)：1–13，2020年。[8] Adelina Ciurumelea、Andreas Schaufelb¨uhl、Sebastiano Panichella 和 Harald C Gall。分析移动应用的评论与代码，以改进发布规划。载于*2017*年*IEEE*第*24*届国际软件分析、演化与再工程会议（*SANER*），第91–102页。IEEE，2017年。 [9] Ishita Daga、Anchal Gupta、Raj Vardhan 和 Partha Mukherjee。利用文本信息检索预测点赞和转发。*Procedia ComputerScience*，168：123–128，2020年。[10] Jacob Devlin、Ming-Wei Chang、Kenton Lee 和 Kristina Toutanova。BERT：面向语言理解的深度双向转换器预训练。*arXiv* 预印本：*1810.04805*，2018年。[11] Andrea Di Sorbo、Sebastiano Panichella、Carol V Alexandru、Junji Shimagaki、Corrado A Visaggio、Gerardo Canfora 和 Harald C Gall。用户会在我的应用中更改什么？总结应用评论以推荐软件变更。载于第*24*届*ACM SIGSOFT*软件工程基础国际研讨会论文集，第499–510页，2016年。 [12] Umar Farooq、A. B. Siddique、Fuad Jamour、Zhijia Zhao 和 Vagelis Hristidis。面向应用的用户评论回复生成。载于*2020*年*IEEE*国际大数据会议（*Big Data*），第699–708页，2020年。[13] Cong Fu、Chao Xiang、Changxu Wang 和 Deng Cai。利用导航式扩散图进行快速近似最近邻搜索。*PVLDB*，12(5)：461–474，2019年。[14] C. Gao、J. Zeng、X. Xia、D. Lo、M. R. Lyu 和 I. King。自动生成应用评论回复。载于第*34*届*IEEE/ACM*国际自动化软件工程会议，第163–175页，2019年。[15] Cuiyun Gao、Wujie Zheng、Yuetang Deng、David Lo、Jichuan Zeng、Michael R Lyu 和 Irwin King。从用户反馈中识别新兴应用问题：微信实践。载于*IEEE/ACM*第*41*届国际软件工程会议：软件工程实践，第279–288页，2019年。

[16] Kobi Glick。让回复和改进用户评论变得更加容易。https://android-developers.googleblog.com/2019/05/whats-new-in-play.html，2020年。访问日期：2020-04-28。 [17] Google。Google Play 商店。https://play.google.com/store/apps，2022年。访问日期：2022-08-03。[18] Alex Graves。利用循环神经网络生成序列。arXiv 预印本 arXiv:1308.0850，2013年。 [19] Xiaodong Gu 和 Sunghun Kim。“用户喜爱应用程序的哪些部分？”（t）。载于2015年第30届IEEE/ACM国际自动化软件工程会议，第760–770页。IEEE，2015年。 [20] Safwat Hassan、Chakkrit Tantithamthavorn、Cor-Paul Bezemer 和 Ahmed E Hassan。研究 Google Play 商店中免费应用程序的用户与开发者之间的对话。实证软件工程，23(3)：1275–1312，2018年。 [21] Claudia Iacob 和 Rachel Harrison。从在线评论中检索并分析移动应用功能请求。载于2013年第10届软件仓库挖掘工作会议，第41–44页。IEEE，2013年。 [22] Jeff Johnson、Matthijs Douze 和 Herv´e J´egou。利用图形处理器进行十亿规模的相似度搜索。IEEE 大数据汇刊，7(3)：535–547，2019年。 [23] Ghina Khoerunnisa、Widi Astuti 等。使用 EUSBoost，基于用户、内容和时间特征预测转发。Jurnal RESTI（信息系统工程与技术杂志），6(3)：442–447，2022年。 [24] Taku Kudo 和 John Richardson。SentencePiece：用于神经文本处理的简单且与语言无关的子词分词器和去分词器。arXiv 预印本 arXiv:1808.06226，2018年。[25] Kyumin Lee、Jalal Mahmud、Jilin Chen、Michelle Zhou 和 Jeffrey Nichols。谁会转发这条推文？检测 Twitter 上转发信息的陌生人。ACM 智能系统与技术汇刊（TIST），6(3)：1–25，2015年。 [26] Tsung-Yi Lin、Priya Goyal、Ross Girshick、Kaiming He 和 Piotr Doll´ar。用于密集目标检测的焦点损失。载于 IEEE 国际计算机视觉会议论文集，第2980–2988页，2017年。 [27] Yinhan Liu、Myle Ott、Naman Goyal、Jingfei Du、Mandar Joshi、Danqi Chen、Omer Levy、Mike Lewis、Luke Zettlemoyer 和 Veselin Stoyanov。RoBERTa：一种稳健优化的 BERT 预训练方法。arXiv 预印本 arXiv:1907.11692，2019年。 [28] Walid Maalej、Maleknaz Nayebi、Timo Johann 和 Guenther Ruhe。迈向数据驱动的需求工程。IEEE 软件，33(1)。 [29] Yu A Malkov 和 Dmitry A Yashunin。利用分层可导航小世界图进行高效且稳健的近似最近邻搜索。IEEE 模式分析与机器智能汇刊，42(4)：824–836，2018年。[30] William Martin、Federica Sarro、Yue Jia、Yuanyuan Zhang 和 Mark Harman。软件工程应用商店分析综述。IEEE 软件工程汇刊，43(9)：817–847，2016年。[31] Bryan McCann、Nitish Shirish Keskar、Caiming Xiong 和 Richard Socher。自然语言十项全能：作为问答的多任务学习。arXiv 预印本 arXiv:1806.08730，2018年。 [32] Maleknaz Nayebi 和 Guenther Ruhe。面向超级移动应用程序的优化功能。载于2017年IEEE第25届国际需求工程会议（RE），第388–393页。IEEE，2017年。 [33] Maleknaz Nayebi 和 Guenther Ruhe。非对称发布规划：在满意度与不满意度之间进行权衡。IEEE 软件工程汇刊，45(9)：839–857，2018年。 [34] Paolo Nesi、Gianni Pantaleo、Irene Paoli 和 Imad Zaza。评估推文的转发倾向：用于转发的预测模型。多媒体工具与应用，77(20)：26371–26396，2018年。[35] Alec Radford、Jeffrey Wu、Rewon Child、David Luan、Dario Amodei、Ilya Sutskever 等。语言模型是无监督多任务学习器。OpenAI 博客，1(8)：9，2019年。[36] Colin Raffel、Noam Shazeer、Adam Roberts、Katherine Lee、Sharan Narang、Michael Matena、Yanqi Zhou、Wei Li 和 Peter J Liu。探索统一文本到文本转换器的迁移学习极限。[37] Colin Raffel、Noam Shazeer、Adam Roberts、Katherine Lee、Sharan Narang、Michael Matena、Yanqi Zhou、Wei Li、Peter J Liu 等。探索统一文本到文本转换器的迁移学习极限。J. Mach. Learn. Res.，21(140)：1–67，2020年。 [38] Nils Reimers 和 Iryna Gurevych。Sentence-BERT：使用孪生 BERT 网络的句子嵌入。arXiv 预印本 arXiv:1908.10084，2019年。 [39] Nils Reimers 和 Iryna Gurevych。Sentence-BERT：使用孪生 BERT 网络的句子嵌入。载于2019年自然语言处理实证方法会议论文集。计算语言学协会，2019年11月。

[40]Paula Rooney。80-20法则适用于漏洞，而不仅仅是功能。http://www.crn.com/news/security/18821726/microsofts-ceo-80-20-rule-applies-to-bugs-not-just-features.htm。 [41] Ramsha Saeed、Haider Abbas、Sara Asif、Saddaf Rubab、Malik M Khan、Naima Iltaf 和 Shynar Mussiraliyeva。使用深度时间传播模式预测早期新闻流行度的框架。*Expert Systems with Applications*，195:116496，2022。[42] Florian Schroff、Dmitry Kalenichenko 和 James Philbin。FaceNet：用于人脸识别和聚类的统一嵌入。在*IEEE*计算机视觉与模式识别会议论文集中，第815–823页，2015年。[43] Rico Sennrich、Barry Haddow 和 Alexandra Birch。使用子词单元进行稀有词的神经机器翻译。*arXiv*预印本 *arXiv:1508.07909*，2015年。[44] Noam Shazeer 和 Mitchell Stern。Adafactor：具有次线性内存开销的自适应学习率。在国际机器学习会议中，第4596–4604页。PMLR，2018年。[45] AB Siddique、Fuad Jamour 和 Vagelis Hristidis。语言增强且上下文感知的零样本槽位填充。在*2021*年网络会议论文集中，第3279–3290页，2021年。[46] AB Siddique、Fuad Jamour、Luxun Xu 和 Vagelis Hristidis。通过常识知识实现广义零样本意图检测。在第*44*届国际*ACM SIGIR*信息检索研究与发展会议论文集中，第1925–1929页，2021年。[47] AB Siddique、MH Maqbool、Kshitija Taywade 和 Hassan Foroosh。通过零样本可泛化奖励函数个性化面向任务的对话系统。在第*31*届*ACM*国际信息与知识管理会议论文集中，第1787–1797页，2022年。[48] AB Siddique、Samet Oymak 和 Vagelis Hristidis。通过深度强化学习进行无监督释义改写。在第*26*届*ACM SIGKDD*知识发现与数据挖掘国际会议论文集中，第1800–1809页，2020年。[49] Muhammad Abu Bakar Siddique。面向开放域自然语言处理的无监督学习与零样本学习。加利福尼亚大学河滨分校，2021年。[50] Kaitao Song、Xu Tan、Tao Qin、Jianfeng Lu 和 Tie-Yan Liu。MPNet：面向语言理解的掩码与置换预训练。神经信息处理系统进展，33:16857–16867。[51] Carole H Sudre、Wenqi Li、Tom Vercauteren、Sebastien Ourselin 和 M Jorge Cardoso。高度不平衡分割的广义Dice重叠作为深度学习损失函数。在医学图像分析与临床决策支持多模态学习中的深度学习中，第240–248页。Springer，2017年。[52] Sho Tsugawa 和 Kosuke Kito。转发作为社交媒体用户关系的预测因子。*PloS one*，12(1):e0170279，2017年。[53] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Łukasz Kaiser 和 Illia Polosukhin。Attention Is All You Need。在神经信息处理系统进展中，第5998–6008页，2017年。[54] Phong Minh Vu、Tam The Nguyen、Hung Viet Pham 和 Tung Thanh Nguyen。移动应用评论中的用户意见挖掘：一种基于关键词的方法。ASE ’15，第749–459页。IEEE Press，2015年。[55] Ronald J Williams 和 David Zipser。持续运行的全循环神经网络的学习算法。*Neural computation*，1(2):270–280，1989年。[56] Cheng Yang、Lingang Wu、Chunyang Yu 和 Yuliang Zhou。移动应用评论中的短语级用户请求挖掘方法：概念、框架与操作。*Information*，12(5):177，2021年。[57] Zhilin Yang、Zihang Dai、Yiming Yang、Jaime Carbonell、Russ R Salakhutdinov 和 Quoc V Le。XLNet：用于语言理解的广义自回归预训练。神经信息处理系统进展，32，2019年。[58] Jianzhang Zhang、Yinglin Wang 和 Tian Xie。基于在线用户评论挖掘的软件功能改进优先级排序。*Information and Software Technology*，108:30–34，2019年。[59] Xue Zhang、Hauke Fuehres 和 Peter A Gloor。通过Twitter预测股票市场指标：“我希望事情没有我担心的那么糟”。*Procedia- Social and Behavioral Sciences*，26:55–62，2011年。[60] Yujin Zhu、Zhe Wang 和 Daqi Gao。不平衡问题的引力固定半径最近邻。*Knowledge-Based Systems*，90:224–238，2015年。