图像描述在自动化软件文档中的应用：一项实证研究

Kevin Moran†、Ali Yachnes*、George Purnell*、Junayed Mahmud†、Michele Tufano‡、Carlos Bernal Cardenas‡、Denys Poshyvanyk*、Zach H'Doubler*

†乔治·梅森大学，美国弗吉尼亚州，*威廉与玛丽学院，美国弗吉尼亚州，‡微软，美国华盛顿州kpmoran@gmu.edu, ayachnes@email.wm.edu,gwpurnell@email.wm.edu, jmahmud@gmu.edu michele.tufano@microsoft.com, carlosbe@microsoft.com, denys@cs.wm.edu, pzhdoubler@email.wm.edu

arXiv:2301.01224v1 [cs.SE] 3 Jan 2023

摘要——**现有的软件文档自动化技术通常尝试在两种主要信息来源之间建立联系：代码和自然语言。然而，由于更抽象的自然语言与结构更严谨的编程语言之间存在词汇鸿沟，这一推理过程往往十分复杂。图形用户界面（GUI）可能成为弥合这一鸿沟的桥梁，因为图形用户界面能够将有关底层程序功能的重要信息编码为丰富的像素数据表示。本文首次对图形用户界面与软件功能的自然语言描述之间的联系展开全面的实证研究。首先，我们收集、分析并开源了一个大型的功能性图形用户界面描述数据集，其中包含针对热门安卓应用的 10,204 张屏幕截图所撰写的 45,998 条描述。这些描述由人工标注人员提供，并经过多种质量控制机制处理。为深入了解图形用户界面的表征潜力，我们研究了四种神经图像描述模型在以屏幕截图为输入时，预测不同粒度自然语言描述的能力。我们使用常见的机器翻译指标对这些模型进行定量评估，并通过一项大规模用户研究进行定性评估。最后，我们总结经验，并讨论多模态模型在增强未来软件文档自动化技术方面展现出的潜力。**

索引词—**软件文档、图像描述、深度学习**

一、引言与动机

完善的文档通常被视为构建和分发现代软件系统不可或缺的组成部分。事实上，既有研究已阐明文档在开发生命周期中的普遍益处 [1]，[2], [3], [4]，以及技术文档对软件维护与演化的重要性 [5]。然而，尽管文档完善的系统具有诸多价值，现代开发流程和种种限制往往导致人们忽视或放弃一系列文档编写任务 [6], [5], [2], [7]，[8], [1]。这些困难催生了大量关于自动化技术的研究，旨在通过为给定任务生成各类文档，减轻利益相关者的负担。例如，已有方法能够自动生成代码 [9], [10], [11], [12], [13], [14], [15], 、应用程序接口 [16], [17], 、单元测试 [18]，、缺陷报告 [19]，[20], 、发行说明 [21], [22], 和提交信息 [23], [24],等工件的自然语言摘要和文档 [25], [26]。

总的来说，现有自动生成软件文档的技术主要关注对两种基本信息模态之间关系的建模：代码和自然语言（NL）。遗憾的是，由于词汇鸿沟的存在，对这两种信息来源进行推理十分困难；这一鸿沟源于源代码词汇与自然语言描述中使用的、更抽象的词语和短语之间往往存在截然不同的概念关联[27], [28]。近期，Robillard 等人在一份报告中将这一词汇鸿沟视为信息推断问题*et al.* [29], ，并指出其中的关键研究挑战包括：（i）推断未记录的程序属性；（ii）发现潜在抽象和设计依据。这些挑战表明，要克服代码与自然语言之间的语义脱节，可能需要新的知识来源，以编码传统软件或自然语言词汇中通常缺失的独特程序属性。

对于自动化文档而言，一个在很大程度上尚未得到探索的信息来源，是编码在图形用户界面（GUI）中的视觉软件数据。基于图形用户界面的应用程序在现代面向用户的软件中占据主导地位，这从桌面和移动应用的流行程度便可见一斑 [30]。此外，设计精良的图形用户界面能够让具有技术倾向的用户直观地理解高质量应用程序底层的程序功能。因此，直观而言，应用程序的某些功能属性编码在图形用户界面的可视化、基于像素的表示中，使人类的认知过程能够判断该界面所提供的计算任务。这表明，可视化图形用户界面数据中存在一些潜在模式，它们指示了自然使用案例的存在，而这些案例体现了核心功能 [31]。

鉴于图形用户界面在传达程序相关信息方面固有的表征能力，我们提出以下假设，作为本文研究的基础：

图形用户界面传达程序相关信息的表现力，可以有效利用来支持软件文档编写的自动化技术。

尽管现有关于自动化文档的大多数研究都关注代码与自然语言之间的二分关系，但我们认为，图形用户界面中编码的潜在信息可以通过提供

另一种天然反映程序功能的知识来源，帮助弥合现有的语义文档鸿沟。事实上，基于图形用户界面的软件表示有望应对 Robillard 等人 [29]提出的两项挑战。具体而言，图形用户界面有助于推断未记录的程序属性，这些属性天然体现在图形用户界面控件或小部件的设计中（例如，捕捉某项功能，而低质量的代码标识符或注释无法充分体现该功能）。此外，图形用户界面还可作为挖掘抽象概念或设计依据的来源，否则这些内容可能仍晦暗不明（例如，根据用例解释与某个图形用户界面屏幕相关联的一段代码）。克服这些挑战后，我们认为，以图形用户界面为中心的文档将对以下类型的软件文档产生影响：

技术文档：开发者会利用技术文档，例如代码注释或 README 文件，了解软件的功能和接口，以支持工程任务。要准确地自动生成此类文档，是一项具有挑战性的推理任务。然而，研究表明，面向用户的程序中，图形用户界面相关代码所占比例最高可达一半 [32]。这意味着，图形软件数据通过图形用户界面事件处理程序或规定图形用户界面布局的代码（例如 HTML），与基于图形用户界面的软件项目中的很大一部分代码存在某种关联即。因此，如果自动化技术能够有效地从图形用户界面中推断出重要功能，就可以将其与现有技术结合，为开发者提供自动化支持，例如生成注释，或在具备更强功能上下文感知能力的情况下生成代码摘要。正如本文所示，与可视化图形用户界面数据（即屏幕截图）相比，图形用户界面代码/元数据似乎编码了正交的信息，这表明我们或许能够从可视化图形用户界面数据中推断出仅凭图形用户界面代码可能无法推断的文档信息。

用户文档：开发者通常会为用户提供教程或操作指南等文档，以清晰地展示软件的功能。虽然一些经验丰富的用户能够直接从图形用户界面推断出功能，但终端用户的技术水平各不相同，许多人依赖各种形式的终端用户文档 [33]。因此，构建能够自动生成此类文档的技术，可以将开发精力解放出来，用于修复错误等其他关键任务。除了面向用户的常见软件辅助工具外，以 GUI 为中心的程序文档还可以支持全新类别的自动化无障碍功能，而移动应用对此迫切需要 [34]。例如，我们可以设想一种屏幕到功能的引擎，取代常见的文本转语音引擎，帮助行动不便的用户浏览软件，而无需额外的开发工作。

为探究以图形用户界面为核心的自动化软件文档编写的潜力，我们针对这一新兴研究方向最基础的任务开展了首批全面的实证研究之一：给定软件图形用户界面的屏幕截图（或屏幕相关信息），生成自然语言描述。鉴于上述任务是前文讨论的各种潜在应用的基础，我们认为这是探究未来技术可行性的合乎逻辑的第一步。为此，我们收集并分析了一个数据集，旨在理解视觉语义并预测应用程序功能（CLARITY 数据集）。该数据集包含热门安卓应用的 10,204 张屏幕截图及其 45,998 条功能描述，这些应用均可在 Google Play 上获取。我们通过与其他语料库进行交叉熵比较并开展主题建模分析，对该数据集进行了描述性分析，以考察所收集描述的“自然性”和语义主题。为了从该数据集中学习屏幕的功能描述，我们定制、训练并测试了四种用于神经图像描述的深度学习（DL）模型——其中三种从图像数据中学习，另一种从图形用户界面文本元数据中学习——以不同粒度预测软件的功能描述。我们通过测量广泛使用的 BLEU 指标进行定量评估，并通过大规模用户研究进行定性评估。总之，本文的贡献如下：

- 我们收集了 CLARITY 数据集，其中包含来自热门安卓应用的 10,204 张屏幕截图，并附有 45,998 条功能性自然语言描述。这些自然语言描述由人工标注人员提供，经过了多种质量控制机制处理，涵盖屏幕功能的高层和低层描述。虽然已有其他图形用户界面数据集 [35], [36],，但 CLARITY 数据集的不同之处在于，它提供了一组经过详尽标注的屏幕，与 Flickr8K [37]或 MSCOCO [38]类似；
- 我们通过主题建模展示 CLARITY 数据集中存在的内在自然模式。
- 我们使用标准指标和大规模用户研究，对四种针对图像描述定制的深度学习模型进行了全面的定量和定性评估；
- 我们提供了一份在线附录，其中包含模型生成的描述示例和实验数据 [39]。我们的数据集、训练好的模型、代码和评估脚本均已开源，可通过该附录访问。

II. 背景

A. 图像与自然语言之间的联系

图像描述任务比分类或标注困难得多，因为一个有效的模型必须能够自动从图像中学习显著特征，并在语义上将这些特征与恰当的自然语言词语和语法对应起来，以描述这些特征。这项在两种截然不同的信息模态之间进行语义对齐的任务，促成了多模态深度学习架构的发展；这类架构将自然语言与基于像素的信息联合嵌入，以预测给定输入图像的恰当描述。这些技术通常在大规模数据集上训练，这些数据集包含带有多个说明文字标注的图像，例如 MSCOCO [38], ，其设计在很大程度上受到编码器 — 解码器神经语言模型的启发，而这类模型传统上应用于机器翻译任务。在本文中，

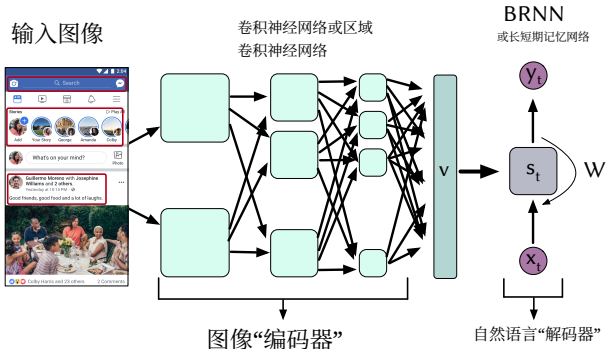


图1：图像描述多模态深度学习架构概览（使用区域卷积神经网络）

我们调整了三种近期的图像描述架构：neuraltalk 2 [40]，、im2txt [41]，和show,attend and tell (SAT) [42] 框架，通过采用定制的预训练和微调流程来预测软件截图的功能性描述。此外，我们还探索了seq2seq神经语言模型。

用于图像描述的深度学习模型建立在编码器-解码器神经语言模型的成功基础之上。im2txt框架将图像描述视为机器翻译问题，其中源“句子”是一幅图像，而目标“译文”是自然语言描述。这类模型的通用架构如图1所示。如图所示，这些架构将编码器循环神经网络替换为卷积神经网络（CNN）；事实证明，卷积神经网络非常擅长学习丰富的图像特征 [43], [44], [45]。谷歌对im2txt 的实现使用长短期记忆（LSTM）循环神经网络 [46] 作为“解码器”模块，在机器翻译任务中也已证明极其有效。neuraltalk2架构的解码器模块由双向循环神经网络（BRNN）[47]构成，而非 LSTM。最后，Show, Attend and Tell（SAT）模型 [42] 采用 LSTM 解码器，并加入注意力机制，通过结合“硬”注意力机制与“软”注意力机制，使模型能够“关注”图像表示中的显著部分。

III. 概述

在本节中，我们将概述本文的数据收集流程以及所进行的各项分析。图2展示了本文的四个主要组成部分。我们研究的首要任务是构建合适的截图-描述对数据集。我们分两部分介绍这一过程：（i）收集屏幕截图（第四节 A），以及（ii）从人工工作者处收集说明文字（第四节 B）。这项数据收集工作的成果是 CLARITY 数据集，其中包含 10,204 张安卓屏幕截图的 45,998 条说明文字。接下来，我们通过实证分析了解说明文字的词汇特性，以更好地了解对其进行建模的难易程度（第五节）。因此，我们比较了使用描述语料库训练的语言模型与其他常用软件工程语料库的交叉熵，并进行了基于 LDA 的

主题分析。接下来，我们将讨论配置和训练三个神经图像描述模型以及一个

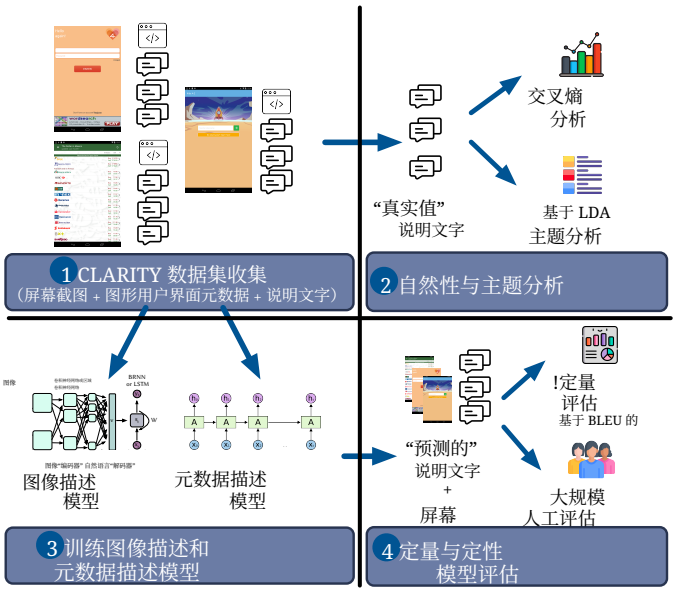


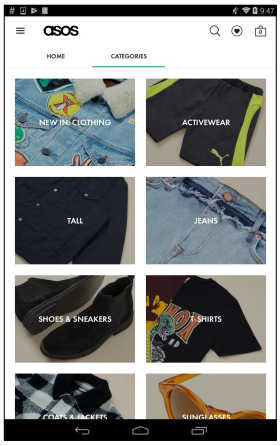
图 2：数据集收集与分析概览

基于序列的模型，以预测软件图形用户界面的功能描述（第六节）。最后，我们将通过自动参考文本评估指标（即 BLEU@n）和大规模人工评估，衡量训练后模型的准确性，从而总结我们的分析。（第七节）

IV. 数据集收集

A. 屏幕与图形用户界面元数据收集

构建 CLARITY 数据集的第一步，是收集一个规模可观且多样化的屏幕截图和 GUI 元数据数据集。我们选择以安卓平台为重点来构建该数据集，主要有三个原因：（i）安卓目前是全球最受欢迎的操作系统 [30]，（ii）Android 应用高度依赖图形用户界面和手势，因此是我们研究的合适对象；（iii）安卓的 screencap和 uiautomator 工具有助于从正在运行的应用程序中提取屏幕截图和 GUI 元数据。值得庆幸的是，相关文献中已有公开的大规模安卓屏幕截图和元数据数据集 [48], [35]。在这项工作中，我们利用了 REDRAW [48], [36] 数据集，其中包含来自 Google Play 商店评分最高的 8,655 个应用程序的近 17,000 张独特屏幕截图。值得注意的是，另一个包含更多屏幕截图的大规模安卓图形用户界面数据集 RICO 也已开放使用 [35]。然而，我们选择使用 REDRAW 数据集，因为它符合我们的一个主要研究目标。也就是说，我们旨在从屏幕截图和 GUI 元数据中学习潜在特征信息。然而，为了使 GUI 元数据与屏幕上显示的内容正确对应，应用程序必须使用 原生 安卓组件。因此，主要使用 Web 技术来显示信息的应用程序，即所谓的混合应用，会使 GUI 元数据变得模糊，并影响我们的研究结果。REDRAW 数据集包含一组经过多个阶段筛选的屏幕截图，以剔除混合应用实例及其他噪声。此外，REDRAW 数据集还包含一组图形用户界面组件图像



高层级描述

该屏幕允许用户查看服装类别

低层级说明文字

左上角图标可让用户访问菜单

右上角图标可让
用户访问购物车

中间的类别列表可让
供用户进行选择

心形图标位于.....的左侧！
购物车左侧的心形图标允许用户
查看收藏夹

图3：CLARITY 数据集中的说明文字示例。

并标注了相应的类型（例如，按钮），我们将在后续研究中使用这些类型（第 VI 节）。这一筛选过程最终得到了总计 17,203 个待标注的候选屏幕。有关筛选过程的完整细节，请参阅 REDRAW 论文 [48]。

B. 功能描述的收集

确定一组合适的屏幕后，我们需要为这些屏幕手动标注功能性说明。这个过程分为两个步骤：(i) 首先，我们开展了一项试标注研究，以制定并验证适用于大规模说明收集的标注方法；(ii) 其次，我们利用亚马逊土耳其机器人的众包工作者平台开展了全面的数据收集研究，收集了超过 1 万个带有功能描述的屏幕。

*1)*说明粒度： 直观而言，图形用户界面会以多个粒度层级编码功能信息。例如，如果询问用户或开发者某个屏幕的高层用途是什么，他们可能会说“此屏幕允许用户浏览服装类别”，如图3所示。这类描述构成了特定屏幕的“高层”功能。然而，单个屏幕很少只实现一种功能，而且可能有多种功能属性共同支撑屏幕的高层功能目的。用户对这类功能属性的描述通常围绕屏幕的交互式组件展开，因为这些组件代表了容易归因于已实现功能的动作实例（例如，用户“正在做某事”）。例如，图3中的屏幕所包含的底层功能包括查看收藏夹、访问购物车，或从列表中选择商品。这类围绕图形用户界面组件展开的“低层”屏幕属性，描述了关键的构成性功能。因此，为了全面呈现每个屏幕的功能视图，我们要求参与者为每个屏幕标注一条“高层”功能性说明，以及最多四条“低层”功能性说明。图3展示了这两类说明，所用的均为 CLARITY 数据集实际收集的说明文字。

2) 试点数据收集研究：我们开发了一个基于 Java 的初始图像图注收集平台，该平台是一个网络应用程序。作者使用该系统，按照前文所述的图注粒度，为 743 个屏幕进行了人工标注。在本次研究中，我们发现一些屏幕上显示的信息相对较少，即使采用前文讨论的过滤技术，也难以为其标注功能属性。因此，在使用 MTurk 开展大规模图注收集之前，至少有一位作者人工检查了 17,203 个候选屏幕中的每一个，并剔除了功能严重缺失的屏幕。最终得到 16,311 个候选屏幕，用于研究的下一阶段。

3) ***Mechanical Turk*** 数据收集研究： 为了开展大规模数据收集，我们调整了用于收集网页应用程序说明文字的机制，使其能够与 MTurk 众包工作者平台协同工作。为此，我们配置了一项人类智能任务（HIT），向工作者提供一组详细说明，并在数据集中展示一张屏幕截图，同时提供文本输入框，供其填写一条高层级功能性说明以及至多四条低层级功能性说明（设置此上限是为了统一工作者完成 HIT 所需的时间）。这项研究已获得作者所属机构的机构审查委员会批准。

鉴于我们的目标是用自然、地道的英语收集高质量的屏幕功能描述，我们面向主要英语国家的 MTurk 用户招募参与者，要求他们至少完成过 1,000 项 HIT，且 HIT 批准率至少为 90%。我们提供了一套详细的图像说明文字标注指南，通过示例清楚说明高层级说明文字和低层级说明文字的概念，并向用户提供了明确的操作说明和注意事项。完整的说明可参见我们的在线附录 [39]。对于说明文字质量，我们特别提出了三项主要要求：(i) 说明文字应描述屏幕所体现的功能，而非仅仅描述其外观；(ii) 低层级说明文字应包含空间指示 （例如，“屏幕左上角的按钮”）；(iii) 说明文字应使用完整的英语句子，语法也应基本正确。

我们通过从包含 16,311 个候选屏幕的集合中抽取不重复的屏幕，分批发布 HIT 任务，并确保每位用户不会两次收到同一个屏幕。众包任务的工作质量并不总是理想，因此提交说明文字后，需要对其质量进行审核。因此，每个屏幕的说明文字都至少由一位作者依据上述三项质量属性进行检查。如果作者不确定某个屏幕是否符合这些质量属性，则至少由另一位作者进行审核，以达成共识。由于质量问题，共有 2,419 个屏幕被拒绝，并作为新的 HIT 重新发布。总之，2,150 名 MTurk 工人收集了 10,204 个屏幕的 45,998 条图注（涵盖不同粒度），每位参与者对应 ≈5 个屏幕，支付报酬总计超过 2,400 美元。

V. 实证数据集分析

CLARITY 数据集为探索基于图形用户界面的数据与词汇

表 I：基于高层描述学习得到的 LDA 主题 $k = 15$

指定标签	排名前七的词
”颜色选项”	屏幕 显示 应用程序 选项 颜色 书籍 不同
”登录或创建账户”	用户屏幕允许账户登录并创建应用程序
”从列表中选择图像”	用户屏幕允许选择图像列表视图
”按位置搜索地图”	屏幕位置搜索地图，用户查找并显示

表 II：基于低层描述学习的 LDA 主题 $k = 25$

指定标签	排名前七的词
”页面按钮”	页面按钮 顶部、中心、底部、侧面、左侧
”选择日期”	可用日期 选择 一个选项 主题 当前
”相机按钮 ”	视频 图像 照片 图片 底部 相机
“隐私政策横幅”	标题 只是术语 蓝色横幅 隐私 政策

软件数据之间的关系提供了丰富的数据来源。然而，为了更好地理解收集的描述的语义构成，有必要进行研究，以了解：(i) 它们所涵盖的潜在主题，以及 (ii) 它们的自然性，进而了解其可预测性。本节将围绕以下两个研究问题（RQs）对这一现象进行实证分析：

- **研究问题$_1$：** CLARITY 数据集中，高层和低层图注中捕捉到了哪些潜在主题？
- **研究问题$_2$：** CLARITY 数据集中的高层和低层图注有多自然（即多可预测）？

A. 分析方法

1) 研究问题$_1$：探究数据集主题： 为了探究 CLARITY 数据集中的潜在主题，我们针对代表不同粒度功能描述的说明语料库训练了主题模型。具体而言，我们对 CLARITY 数据集中的高层级和低层级分段图注均应用了潜在狄利克雷分配（LDA） [49] 。在我们的分析中，CLARITY 数据集中某一特定屏幕截图的所有说明文字构成一个文档，而给定粒度（即高层级或低层级）下所有屏幕截图的说明文字则构成一个语料库。LDA 有若干可配置的超参数，会影响生成主题的平滑程度。其中包括 k （主题数量）、n （采样算法的迭代次数；本研究使用吉布斯采样 [50], ），以及影响主题分布的 α和 β 。我们将 α 和 β 设为自然语言语料库的标准值，将 n 设为 500；这一数值足以确保模型收敛，并将 k 分别设为 15、25、50 和 75 个主题。

2) 研究问题$_2$：分析图形用户界面描述的自然性：以往研究率先提出了软件自然性 的概念 [51], ，并指出软件即使

与自然语言相比，软件呈现出更多重复模式，因此更具可预测性。近期的研究进一步探讨了这一发现，并证实了某些自然模式的存在 [52]。为说明自然性，这些早期研究基于软件语料库训练了统计n-gram语言模型，并测量了这些模型的“困惑度”（或其经过对数变换的版本，交叉熵）。当模型基于来自同一领域的语料库进行训练时，该指标表示模型对测试语料库中模式的“意外”程度。测得值更低的模型

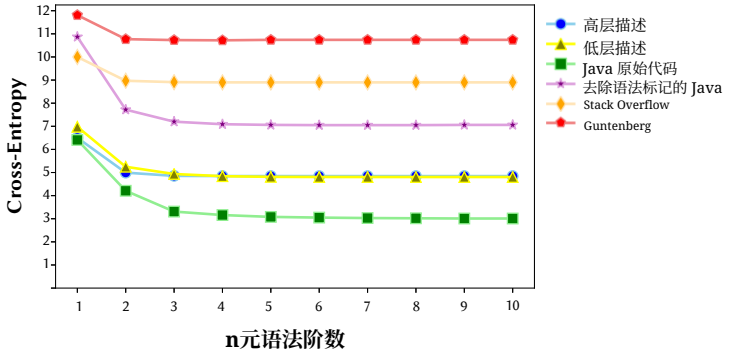


图4：CLARITY 数据集高层和低层图注的交叉熵与其他语料库的比较。

交叉熵代表更高的预测能力，因此也代表更具自然性的底层语料库。

我们遵循这些既有研究的方法，探究 CLARITY 数据集说明文字的自然性。因此，与前一个研究问题的方法类似，我们将收集的描述分为两个语料库，一个用于高层描述，另一个用于低层描述。随后，我们按照十折交叉验证流程，使用 `mitlm` [53] Kneser-Ney 平滑 [54],的实现 [51],训练插值 n 元语法模型；该平滑方法已被证明是最有效的 n 元语法平滑方法。我们报告这些实验中高层和低层语料库的平均交叉熵值，并与先前关于其他自然语言和软件语料库的结果进行比较 [51], [52] 。

B. 分析结果

1) 研究问题$_1$：数据集主题建模结果： 我们在表 I 和表 II 中展示了部分最具代表性的主题结果，并附上为便于阅读而提供的描述性标签；所有结果均收录于附录 [39]。这些主题有助于直观展示 CLARITY 高层和低层说明文字中存在的一些潜在模式。高层描述呈现了若干屏幕级主题，包括在地图上搜索以及调整应用设置。相比之下，低层描述捕捉了描述组件级功能的主题，例如日期选择器、相机按钮和返回按钮。这些结果表明，我们收集的描述文字中存在专属于图形用户界面领域的逻辑主题。

*2)*研究问题$_2$：***Clarity*** 描述的自然性： 我们的自然性分析结果如图 4 所示。该图展示了 CLARITY 数据集中的高层和低层说明文字与其他几个语料库的平均交叉熵对比，数值由 Rahman 等人 [52] 计算得出。具体而言，图中展示了以下语料库的十折平均交叉熵：(i) 古腾堡语料库，其中包含 3,000 多本由一百多位不同作者撰写的英文书籍；(ii) 来自 GitHub 上 134 多个开源项目的 Java 代码；(iii) 不含语法标记（即分隔符、关键字和运算符）的 Java；以及 (iv) 一个 Stack Overflow 语料库，其中仅包含 20 多万篇帖子的英文描述。

如前所述，对于某个特定数据集，交叉熵越低，其自然性就越高。也就是说，交叉熵较低的语料库往往具有更强的潜在模式，因而能够被有效建模和预测。从图4可以看出，CLARITY 的高层和低层说明文字比除原始 Java 代码以外的所有数据集都更自然。值得注意的是，相较而言，有几个因素可能导致 CLARITY 说明文字的交叉熵较低。例如，其他语料库可能规模更大，或包含更多样的人类作者和写作风格。不过，我们提供其他数据集的熵度量，主要是为了对比 CLARITY 数据集与其他常见语料库的可预测性。尽管数据集存在差异，针对 CLARITY 两种说明文字数据集测得的平均熵为 ≈ 5 比特，这表明我们收集的描述具有强烈的语义模式，能够被有效建模以进行预测。此外，我们发现高层和低层说明文字的交叉熵惊人地相似。直观而言，人们可能会认为，由于某些图形用户界面组件（如菜单按钮）的用例反复出现，CLARITY 的低层说明文字会呈现更多常见模式。这表明这两个数据集都倾向于呈现可恰当建模的模式。不过，正如我们在第七节所展示的，图形用户界面相关信息预测说明文字的能力会随粒度而异。

VI. 从软件图形用户界面生成深度学习功能描述

上一节的分析结果表明，CLARITY 屏幕截图和说明文字数据集中存在潜在模式。本节将详细介绍我们的方法，用于研究不同定制深度学习模型学习这些模式并基于两种图形用户界面表示预测功能描述的能力。

A. Clarity 数据集划分

我们从用户处收集了两种不同粒度的说明文字，以构建 CLARITY 数据集（第四节 B）。在本节的实验中，我们希望探究模型学习高层和低层功能描述的能力。因此，我们将 CLARITY 数据集分为两组，一组仅包含高层级描述，另一组仅包含低层描述。我们还创建了第三个数据集，将高层和低层描述合并，以探究汇集不同粒度的描述是否能提升模型的预测能力。需要注意的是，每条低层图注都被视为一条独立的图注（即每条低层图注都被视为一个独立的数据点），这符合包含多条图注的数据集的惯例 [38]。数据集中的每个屏幕都关联一张屏幕截图和一个图形用户界面元数据文件。为了公平比较不同模型配置的性能，我们创建了统一的训练集、验证集和测试集划分（按照图像/图形用户界面元数据文件的数量，分别占 80%、10%、10%），并将其用于所有模型。用作模型输入的自然语言文本

表 III：研究中使用的图像描述模型配置

模型	标识符	图注配置	模型配置
i m2txt	im2txt-h-imgnet	High	训练于 Inception v3 ImageNet
	im2txt-l-imgnet	Low	
	im2txt-c-imgnet	组合	
	im2txt-h-comp	High	在.....上微调的 Inception v3 组件数据集
	im2txt-l-comp	Low	
	im2txt-c-comp	组合	
	im2txt-h-fs	High	在以下数据集上微调的 Inception v3 全屏数据集
	im2txt-l-fs	Low	
	im2txt-c-fs	组合	
NeuralTalk 2	ntk2-h-imgnet	High	预训练于 ImageNet
	ntk2-l-imgnet	Low	
	ntk2-c-imgnet	组合	
	ntk2-h-ft	High	预训练于 VGGNet ImageNet，采用微调
	ntk2-l-ft	Low	
	ntk2-c-ft	组合	
SAT	高 SAT	High	预训练于 ImageNet
	低 SAT	Low	
	sat-c	组合	

表 IV：模型超参数子集

超参数	im2txt		SAT	序列到序列
批大小	64	16	17	64
嵌入维度	512	512	512	128
解码器循环神经网络大小/单元数	512	512	1024	128
优化器	SGD	SGD	Adam	Adam
初始学习率	2	2	0.001	0.0001
丢弃概率	0.7	0.7	0.3	0.8

根据各模型实现的具体要求进行了预处理 [55], [56], [57]。

B. 图像描述模型配置

我们在 CLARITY 数据集的屏幕截图和说明文字上，对三种神经图像描述模型 im2txt、NeuralTalk2 和 Show, Attend and Tell（SAT）（第二节 A）进行定制、训练和测试。选择探究这三种模型，是因为它们在所采用的卷积神经网络和循环神经网络类型方面有着不同的底层设计决策（第二节 A），这些差异可能会影响它们在我们领域中的表现。需要注意的是，在实验过程中，我们通过调整预训练和微调流程，对这些模型进行了多项定制。然而，鉴于构成这些模型的参数通常数量庞大，即使使用现代硬件，训练耗时也可能相当可观。因此，为了控制实验复杂度，并研究若干能够在合理时间内完成训练的模型配置，我们固定了实验中各模型的超参数值。我们通过随机搜索确定部分参数的最优取值，并为其他参数选用了先前研究报告的最优值。虽然固定了这些模型的超参数，但我们转而在架构层面对图像描述模型的配置进行定制。具体而言，我们研究使用不同数据集和训练流程训练“编码器”卷积神经网络会如何影响模型预测的有效性。此类分析有助于我们更有效地梳理与模型设计决策的利弊相关的总体规律。最终，我们在数个月的机器计算时间内，训练了 15 种以上的不同模型配置（见表 III）。

1) ***im2txt*** 模型配置与训练：对于 im2txt，我们改编了谷歌在 TensorFlow 中开源的模型实现的部

分。鉴于 im2txt 模型需要训练的参数数量极其庞大，即使是相对简单的超参数搜索，在我们的实验中也被证明在计算上难以承受。因此，对于该模型，我们采用了 Vinyals 等人 [41] 在规模相近的数据集上报告的最优参数集。这些超参数值的部分内容见表 IV，完整的配置详情可在附录中找到。谷歌公开提供的 im2txt 模型实现采用 Inception v3 [58] 图像描述架构作为其编码器卷积神经网络。

在以往的工作中，Inception 模型的权重通过在大规模图像分类数据集 ImageNet [59], 上训练来初始化，该数据集包含“常见”的图像类别。然而，鉴于我们将这些模型应用于一个非常具体的领域（预测软件的描述），尚不清楚在更广泛的 ImageNet 数据集上训练的 Inception v3 模型能否捕捉 CLARITY 数据集中的细微语义模式。因此，我们探索了三种不同的模型配置来研究这一现象：一种使用在 ImageNet 上预训练的 Inception v3，另外两种使用在特定领域数据集上微调的 Inception v3。我们使用的第一个特定领域图像数据集是第 IV-A 节所述的 ReDraw 裁剪图像数据集，其中包含超过 19 万张原生安卓图形用户界面组件图像，并按组件类型标注（例如，按钮、文本视图）。我们使用的第二个特定领域图像数据集由 CLARITY 数据集中的完整屏幕截图构成，并按照其 Google Play 类别进行标注。

*2)**NeuralTalk2** 模型配置与训练*：对于 NeuralTalk2，我们采用了 Karpathy等人以 Torch 和 Lua 编写的实现 [56]。鉴于该模型的训练时间更短，我们对其进行了简短的随机超参数搜索，并以最优的 `im2txt` 参数作为起点。搜索得出的最优值见表 IV。作为卷积神经网络解码器，NeuralTalk2采用了在 ImageNet [59] 数据集上预训练的 VGGNet [44] 架构。与我们的 im2txt 配置不同，我们还探究了联合微调NeuralTalk2的卷积神经网络和循环神经网络的效果。因此，我们探索了 NeuralTalk2的两种配置：一种在 CLARITY 数据集上联合微调预训练的 VGGNet，另一种则不进行微调。我们的训练流程与`im2txt` 模型类似，即使用 CLARITY 数据集的高、低和组合图注训练数据训练模型 50 万次迭代，并每隔 2000 次迭代保存一次模型检查点。

3) 展示、关注与讲述模型配置与训练：对于 `SAT` 模型，我们采用了该模型在 TensorFlow [60]中的开源实现。用于训练模型的超参数见表 IV。该实现使用 VGG16 [44] 作为其编码器卷积神经网络。我们在 CLARITY 数据集上，分别使用低、高和组合说明文字训练 `SAT` 模型 500K 次迭代，并每隔 1K 次迭代保存一次检查点。请注意，由于该模型的训练成本高得令人难以承受，我们没有像使用 `neuraltalk2`那样，尝试使用经过微调的 VGGNet。

表 V：元数据图注模型配置

模型	标识符	图注配置	模型配置
序列到序列	seq2seq-h-type	High	训练于图形用户界面组件类型
	seq2seq-l-type	Low	
	seq2seq-c-type	组合	
	seq2seq-h-文本	High	训练于图形用户界面-组件文本
	seq2seq-l-文本	Low	
	seq2seq-c-文本	组合	
	seq2seq-h-tt	High	训练于GUI 组件类型 + Text
	seq2seq-l-tt	Low	
	seq2seq-c-tt	组合	
	seq2seq-h-ttl	High	训练于GUI 组件类型 + 文本 + 位置
	seq2seq-l-ttl	Low	
	seq2seq-c-ttl	组合	

C. 元数据图注模型配置

为了探索在图形用户界面元数据的词汇表示与自然语言功能描述之间进行转换的能力，我们使用谷歌的序列到序列 [57] 框架训练并测试一种编码器-解码器神经语言模型。请注意，近期研究提出了能够利用结构化文本属性 [61], 的新模型，但这类模型的实现通常不可用，因此我们将更先进模型的研究留待未来工作。我们选择采用该模型默认的通用架构和超参数，因为已有研究表明，它们在广泛的机器翻译任务 [62]中均表现有效。具体而言，我们的编码器网络由一个带门控循环单元（GRU）的双向循环神经网络构成，解码器网络则由一个带长短期记忆单元的循环神经网络构成；超参数列于表 IV。

为了研究安卓图形用户界面元数据中不同属性的代表性，我们创建了四种由不同属性组合构成的图形用户界面元数据配置（表 V）。我们选择使用这些属性组合，因为它们代表了（i）最有可能具有取值的属性，以及（ii）涵盖范围广泛的信息类型（例如，显示文本、组件类型和空间信息）。请注意，序列到序列模型在高级描述数据集上的收敛情况并不稳定，因此我们不报告这些结果。与其他模型的训练方式一致，我们将序列到序列模型训练了 500k 次迭代，并每隔 2k 次迭代保存一次检查点。

VII. 深度学习模型评估

为了探究本文开篇提出的核心假设，并评估第 VI 节所述的深度学习模型，我们开展了全面的实证评估，主要目标有两个：(i) 根据广受认可的机器翻译效果指标，对模型的预测能力进行内在评估；(ii) 通过考察并评定模型预测的自然语言功能描述的质量，对模型进行外在评估。本次评估着重关注所研究模型能否有效预测准确、简洁且完整的功能描述。为实现研究目标，我们提出以下研究问题：

- **研究问题$_3$：** 我们的模型预测的自然语言描述有多准确？
- **研究问题$_4$：** 从评估者的角度来看，我们模型预测的自然语言描述有多准确、完整且易于理解？

表 VI：模型的 BLEU 分数评估结果

模型	Capt.	模型类型	B_c	B_1	B_2	B_3	B_4
im2txt	High	im2txt-h-fs	12.4	24.8	12.6	6.7	5.3
	Low	im2txt-l-comp	27.0	45.6	31.8	20.0	10.1
	组合	im2txt-c-comp	30.3	51.7	35.9	22.1	11.6
NeuralTalk2	High	ntk2-h-imgnet	13.3	27.4	13.5	7.3	5.3
	Low	ntk2-l-ft	27.4	47.5	32.8	19.5	9.6
	组合	ntk2-c-ft	30.1	52.1	36.0	21.8	10.8
序列到序列	Low	seq2seq-l-type	18.1	44.6	17.0	7.9	0.24
	组合	seq2seq-c-type	16.9	38.9	14.7	6.0	0.08
SAT	High	高 SAT	17.7	30.1	18.3	12.9	9.8
	Low	低 SAT	35.0	52.5	38.7	28.1	20.7
	组合	sat-c	37.7	56.8	42.0	30.5	22.0
NeuralTalk2	训练于 Flickr8K		34.0	57.9	38.3	24.5	16.0
NeuralTalk2	训练于 MSCOCO		40.7	62.5	45.0	32.1	23.0
im2txt			42.6	66.6	46.1	32.9	24.6
SAT			45.7	71.8	50.4	35.7	25.0

A. 评估方法

1）研究问题$_3$：实证评估模型准确性： 为了评估我们训练的模型生成的说明文字的准确性，我们沿用以往的研究 [40], [41], 报告预测描述在由图像和图形用户界面元数据构成的共享 CLARITY测试集上的 BLEU 分数 [63] 。BLEU 分数是机器翻译研究中使用的一种标准指标，用于衡量预测描述（模型的输出）与参考描述（从 CLARITY 测试集中收集的人类描述）之间的文本相似度。BLEU 分数可根据不同长度子序列的相似度进行测量（即，BLEU$_n$），我们报告 BLEU$_1$ 至 BLEU$_4$, 的分数，以及这些分数的平均综合得分，这也是惯例 [40], [41]。对于图像描述模型，我们采用适配 CLARITY测试集的 BLEU 分数 *coco-caption* 实现。对于所有图像描述模型中的每张测试图像，我们使用宽度为 3 的束搜索，在候选预测中生成三条描述。序列到序列模型也以相同方式进行评估。我们选择使用宽度为 3 的束搜索，因为对模型预测的初步定性检查表明，该宽度在预测准确性和模型置信度之间取得了合理的平衡。对于高层描述，我们将三条候选描述与参考描述进行比较，并计算每个模型的 BLEU$_n$分数总体平均值。对于低层描述和组合描述，我们将预测描述与参考描述逐一配对比较，并计算每种模型配置的 BLEU$_n$ 分数总体平均值。

*2)*研究问题$_4$：人类对预测描述的感知： 为了定性评估所研究模型生成的说明文字，我们开展了一项大规模研究，邀请了另外 220 名从 MTurk 招募的参与者。我们从 CLARITY 测试集随机抽取了 220 个屏幕，并根据各模型与描述层级组合的综合BLEU分数，使用 `im2txt`、`NeuralTalk2` 和 序列到序列的最优配置，为这些屏幕生成高层、低层和组合描述。由于模型训练所需时间有限，本研究未纳入 SAT 说明文字。我们创建了一项 HIT，每位参与者会查看 11 张配有说明文字的屏幕截图。其中两条说明文字是高层和低层参考描述，用作对照；另外 9 条则来自模型预测。

我们将屏幕和描述对安排到各项 HIT 中，并确保：1）每项 HIT 中不会出现两张相同的屏幕截图；2）11 种说明文字（2 条参考描述、9 条模型生成描述）在每项 HIT 中各出现一次。每项 HIT 中这些说明文字的顺序均随机排列，以避免不同 HIT 中相同的说明文字顺序造成偏差。按照这种安排，每个屏幕截图-描述对由 11 名参与者进行评估。查看这些屏幕截图-描述对后，参与者需要回答六个评估问题。其中三个问题（EQ$_1$-EQ$_3$）改编自先前一项评估自动生成的代码摘要质量的研究 [21],, 分别询问准确性、完整性和可理解性。其余三个问题（EQ$_4$-EQ$_6$）为开放式回答，要求参与者说明准确之处、不准确之处以及改进之处。完整的问题集和 HIT 见我们的在线附录 [39]。与 CLARITY 数据集收集类似，每位参与者的回答都由至少一位作者仔细审核；若回答不完整，则予以剔除。我们持续收集回答，直到由不同受访者完成 220 项 HIT。

B. 评估结果

1) 研究问题$_3$ 结果：***BLEU*** 分数评估： 我们在表 VI 中展示了所有训练模型中效果最佳的模型配置和检查点对应的 BLEU分数结果；其他模型配置的结果可在我们的在线附录 [39]中找到，此外还提供了图注示例。蓝色高亮的单元格表示每种图注类型中表现最佳的模型配置。总体而言，我们观察到，`SAT` 在所有图注粒度上都取得了最高的 BLEU 分数。我们推测，这归因于该模型加入了先进的注意力机制，使其能够“聚焦”于不同的图像区域或特征，从而有效处理多种图注粒度。总体而言，序列到序列模型在各种图注类型上的表现都相当差，表明其生成丰富表征的能力较弱。或许最有意思的是，我们发现 i m2txt 框架的最佳模型配置是以特定领域数据集对卷积神经网络进行条件化的配置。具体而言，最佳高层图注模型以完整屏幕截图为条件，而最佳低层图注则以裁剪后的图形用户界面组件屏幕截图为条件。

另一个显著的总体趋势是，与高层描述相比，低层图注模型和组合图注模型往往能取得更高的整体 BLEU 分数。这在一定程度上符合直觉，因为这表明，与反映更抽象功能描述的高层描述相比，低层描述中的可视化图形用户界面与词汇模式之间存在更自然的联系。考察各模型最优配置生成的说明文字后，可以明显看出，`im2txt` 和`SAT` 生成的输出图注更加多样，而`neuraltalk2`生成的图注则较为单一；在许多软件文档任务中，前者可能更有用。

最后，值得探讨的是，我们的模型与在更传统的数据集上训练的相同模型相比，其 BLEU 分数有何差异。

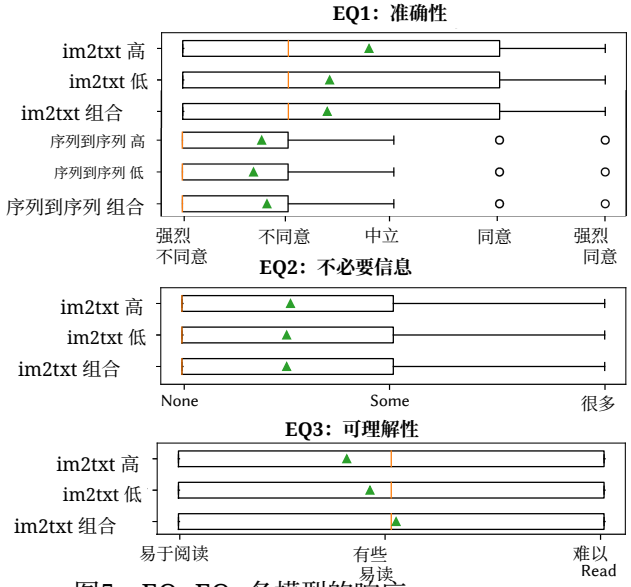


图5：EQ_1-EQ_3 各模型的响应

这些数据集的结果列于表 VI 底部，包括 Flickr8k [37] 和 MSCOCO [38]。鉴于深度学习模型对数据量的需求，以及 MSCOCO 数据集规模大得多（包含 ≈123k 张图像，每张图像有 5 条说明文字），我们并不期待在 CLARITY 数据集上训练的模型能胜过在 MSCOCO 上训练的模型。因此，结果并不意外：在 MSCOCO 数据集上训练的`im2txt`、`neuraltalk2`和`SAT`模型，平均而言，在组合和低层图注上的表现优于在 CLARITY 数据集上训练的相同模型，BLEU 分数高出 ≈ 10 分；在高层描述上的 BLEU 分数则高出 ≈ 27 分。然而，当我们考察 Neuraltalk2 在规模更为接近的 Flickr8K 数据集（包含 ≈ 8K 张图像，每张图像有 5 条说明文字）上的表现时，发现其在 CLARITY 的低层和组合数据集上的表现相当；`SAT` 模型略胜于在 Flickr8K 上训练的 `neuraltalk2` 模型，而在高层描述上的差距稍大。总体而言，这些结果表明，与规模相近的数据集相比，在 CLARITY 数据集上训练的深度学习模型表现相似。

2）研究问题$_4$ 结果：人工评估：图5总结了人工研究期间表现最佳的模型配置在 EQ_1-EQ_3 上的结果，以及序列到序列模型的准确性得分。所有模型配置的完整结果见我们的在线附录。对 EQ_4-EQ_6 的回答因图注类型而异，完整内容已列于附录中。总体而言，im2txt 在准确性方面表现最好，其次分别是 NeuralTalk2 和序列到序列模型。对于 im2txt，尽管参与者的反应不一，但许多受访者确认图注准确无误（例如：“该描述准确地描述了屏幕，事实上这是一个条款与条件页面。”），并提出了类似于参考图像描述的细微改进建议（例如：“可以补充设置涉及的具体内容（即安全性）。”）。如图5所示，im2txt 生成的描述始终被评为可读且包含相关信息。另一个值得关注的现象是

值得注意的是，BLEU 分数所反映的表现与人类的感知似乎并不一致：尽管 NeuralTalk2 在两种模型配置下取得了更高的 BLEU分数，但在 EQ_1-EQ_3，的各项评估中，参与者始终认为 im2txt 生成的说明文字优于其他模型。

VIII. 讨论与经验教训

经验教训 1：图形用户界面的功能描述具有很高的自然性，可以使用深度学习技术进行建模。我们观察到，在低级别和组合数据集上训练的深度学习模型，其表现与在规模相近的通用图像描述数据集（例如，Flickr8K）上训练的模型相似。这表明，图形用户界面屏幕截图可用于扩充自动化文档方法。

第2课：以 GUI 为中心的软件文档模型受益于先在特定领域的 GUI 数据上进行预训练，而非使用通用图像数据集（例如 MSCOCO）对模型的定性分析结果表明，对于`im2txt`，效果最佳的配置是在特定领域的卷积神经网络数据集上训练的。这说明，从通用数据集中学习到的图像特征，与从更贴近软件的数据集中学习到的图像特征，其效用存在明显差异。这表明，未来若要利用深度学习模型生成以 GUI 为中心的程序文档，应着眼于收集大型的与 GUI 相关的数据集，并从中提取特征。

第3课：未来以 GUI 为中心的程序文档自动化方法，很可能会受益于结合屏幕截图与 GUI 元数据所蕴含的正交语义。本文的评估表明，在软件文档任务中，屏幕截图的表征能力似乎更强。然而，鉴于这两种信息模态之间存在显著差异，我们也观察到它们编码了正交的语义模式，结合起来可以更有效地生成文档。我们发现，基于图像的模型生成的某些说明文字受到词汇表有限的影响。例如，某些预测描述与以下内容相似：“屏幕允许用户选择 <未知>”，其中 UNK 标记表示缺失的标记，应映射到某种未观测到的应用属性，例如“专辑封面”或“商店位置”。不过，可以将此类预测与 GUI 元数据中的词汇表结合起来，从而帮助生成更完整、准确的描述。因此，未来一个有前景的研究方向是联合编码屏幕截图和词汇型 GUI 元数据。

第4课：训练图像描述模型以预测特定或多样化的功能内容是一项艰巨的任务。面向图形用户界面文档的实用模型应能够预测特定信息（例如，给定方法处理程序中某个特定按钮的功能），也能预测多样化的功能（即能够生成描述给定屏幕上任意位置功能的文字）。然而，我们在各个模型中观察到的一个现象是，最常见的功能类型（例如，返回按钮、菜单按钮）恰好也是我们的

模型在未见过的屏幕截图中最常预测、且最有把握的功能。这在一定程度上是意料之中的，因为模型在训练期间见过这类功能的样本最多。因此，如何提高预测多样性仍是未来研究需要解决的问题。通过使用规模更大、更多样化且经过专门整理描述的数据集（例如扩展 CLARITY 数据集），可以在一定程度上缓解这一问题。然而，要更有效地预测多样化的特征，可能还需要领域特定模型或模型集成。

第5课：评估以图形用户界面为中心的自动化文档方法的后续研究应纳入人类研究，因为人类对模型的感知可能不同于自动参考文本评估指标。我们的研究中最令人惊讶的结果之一是，人类对深度学习模型生成的说明文字的感知似乎与通常用于评估模型预测准确性的 BLEU 指标之间存在不匹配。这表明，人类感知的某些方面并未被 BLEU 指标以及其他可能的翻译指标有效捕捉。

IX. 局限性与对有效性的威胁

内部效度。对内部效度的威胁是指实验中可能导致观察结果的意外因素。为了构建数据集，我们依赖 MTurk 及其工作人员为每张屏幕截图提取高层和低层描述。需要注意的是，我们并未要求 MTurk 工作人员提供技术软件文档描述，而是要求他们以不同的详细程度描述屏幕功能。为尽量减少低质量说明文字，我们仅向拥有超过 1k HITS、来自英语国家且 HIT 批准率超过 90% 的工作人员发布任务。此外，每个成功完成的 HIT 都至少经过一位作者审核，以确保质量。如果对说明文字质量存在任何疑问，至少会有另一位作者介入解决歧义。因此，由于描述质量低下，我们拒绝了 2,429 个 HIT。

外部效度。对外部效度 的威胁涉及结果的普遍适用性。与任何收集而来的数据集一样，CLARITY 数据集的普遍适用性也存在外部效度威胁。不过，我们使用了来自 Android 领域的多种热门应用程序，从这些应用程序中提取了热门屏幕截图，并由大量且背景多元的 MTurk 工作人员为这些截图撰写说明文字。在数据收集过程中，为了让 MTurk 工作人员能够完成任务，我们每个屏幕仅收集了 4 条低层描述，因为工作人员往往会放弃较长的任务，或表现不佳。这意味着，对于包含许多图形用户界面组件的某些屏幕，部分组件可能缺少自然语言描述。然而，鉴于数据集的规模以及屏幕截图和说明文字的多样性，我们认为这些低层描述具有相当的代表性。

X. 相关工作

用于图像描述和图形用户界面的深度学习。Hossain等人 [64] 近期对用于图像描述的深度学习模型进行了广泛研究，综述了用于评估这些模型的各种架构和数据集。然而，这项综述并未考察任何图像描述模型预测软件功能描述的能力。软件工程社区中，应用深度学习技术处理图形用户界面相关数据的论文数量有限。Chen等人 [65] 设计了一种方法，使用神经机器翻译将安卓屏幕截图转换为图形用户界面骨架。然而，他们的技术只能根据图像预测图形用户界面结构，无法生成描述功能的自然语言文本。近期，Zhang 等人 [66] 创建了一个 iOS 图像描述数据集，用于训练能够为无障碍数据生成说明文字的模型。然而，作者并未公开其数据集，而且他们的目标是处理无障碍数据，这与我们生成描述功能的说明文字这一目标不同。Chen等人 研究了在移动应用中使用深度学习图像描述模型为图形用户界面组件添加标签 [67], ，但这种方法仅旨在为有限范围内的图形用户界面组件预测简短标签，而我们的研究则聚焦于为单个图形用户界面组件和完整屏幕截图生成完整句子形式的功能描述。

基于图形用户界面的移动应用分析。GVT 和 GCat 分析图形用户界面的视觉属性，以检测设计违规和演化变化 [68], [69]。相比之下，我们专注于图像描述技术，旨在为屏幕截图提供功能性程序描述。REMAUI [70], REDRAW [48],和 pix2code [71] 等方法旨在根据应用程序屏幕截图自动生成移动应用代码。相反，我们利用深度学习技术，以基于像素的图像作为输入，生成的是功能描述而非源代码。Chen 等人 [72] 提出了 StoryDroid，可自动生成 Android 应用的可视化故事板，辅助应用设计过程。然而，他们的方法无法从图形用户界面数据中生成应用程序的功能描述。此外，Deka 等人展示了如何通过自编码器 [35]，使用语义搜索浏览 Rico 数据集。UiRef [73]通过考量请求用户输入的图形用户界面组件的语义，来解决安全和隐私问题。此外，Liu 等人 [74] 提出了一种根据语义图形用户界面模式自动分类移动应用图标的方法。Xiao 等人提出了 IconIntent，它结合程序分析与图标分类，以检测涉及隐私的图形用户界面组件[75]。与上述研究不同，我们旨在预测图形用户界面的功能描述，用于软件文档。

XI. 结论

在未来中，我们面实证研究之一，探讨图形用户界面相关信息与程序功能描述之间的联系。我们构建了包含图形用户界面屏幕截图、元数据和自然语言描述的 CLARITY 数据集，在该数据集上训练了深度学习模型，并展示了这些模型弥合程序视觉信息与词汇信息之间语义鸿沟的能力。

致谢

本研究得到美国国家科学基金会 CCF-2007246 和 CCF-1955853 项目资助。本文所表达的任何观点、发现和结论均为作者所有，并不一定反映资助方的意见。

参考文献

[1] J.-C. Chen 和 S.-J. Huang，“软件开发问题因素对软件可维护性的影响：一项实证分析”，《系统与软件杂志》，第 82 卷，第 981–992 页，2009 年 6 月。[2] M. Kajko-Mattsson，“纠正性维护中的文档实践调查”，《实证软件工程》，第 10 卷，第 31–55 页，2005 年 1 月。[3] B. Dagenais 和 M. P. Robillard，“创建和演进开发者文档：理解开源贡献者的决策”，载于第十八届 ACM SIGSOFT软件工程基础国际研讨会论文集，FSE ’10（美国纽约州纽约市），第 127–136 页，ACM，2010年。[4] G. Garousi、V. Garousi-Yusifoglu、G. Ruhe、J. Zhi、M. Moussavi 和 B. Smith，“技术软件文档的使用情况与实用性：一项工业案例研究”，《信息与软件技术》，第 57 卷，第 664–682 页，2015年。[5] A. Forward 和 T. C. Lethbridge，“软件文档、工具和技术的相关性：一项调查”，载于 2002 年 ACM 文档工程研讨会论文集，DocEng ’02（美国纽约州纽约市），第 26–33 页，ACM，2002年。[6] “敏捷宣言 http://agilemanifesto.org。”[7] J. Zhi、V. Garousi-Yusiflu、B. Sun、G. Garousi、S. Shahnewaz 和 G. Ruhe，“软件开发文档的成本、效益与质量”，《系统与软件杂志》，2015年 1 月。[8] B. Fluri、M. Wursch 和 H. C. Gall，“代码和注释会共同演进吗？源代码与注释变更之间的关系”，载于第 14 届逆向工程工作会议论文集，WCRE ’07（美国华盛顿州华盛顿市），第 70–79 页，2007年。[9] L. Moreno、A. Marcus、L. Pollock 和 K. Vijay-Shanker，“JSummarizer：Java类自然语言摘要的自动生成器”，载于 2013 年第 21 届程序理解国际会议，ICPC’13，第 230–232 页，2013 年 5 月。[10] L. Moreno、J. Aponte、G. Sridhara、A. Marcus、L. Pollock 和 K. Vijay-Shanker，“Java类自然语言摘要的自动生成”，载于 2013 年第 21 届程序理解国际会议，ICPC’13，第 23–32 页，2013 年 5 月。[11] P. C. Rigby 和 M. P. Robillard，“在非正式文档中发现核心代码元素”，载于 2013 年国际软件工程会议论文集，ICSE’13（美国加利福尼亚州旧金山），第 832–841 页，2013年。[12] A. T. T. Ying 和 M. P. Robillard，“代码示例摘要的选择与呈现实践”，载于第 22 届 ACM SIGSOFT软件工程基础国际研讨会论文集，FSE’14（中国香港），第 460–471 页，2014年。[13] A. T. T. Ying 和 M. P. Robillard，“代码片段摘要”，载于 2013 年第 9 届软件工程基础联合会议论文集，FSE’13，第 655–658 页，2013年。[14] I. K. Ratol 和 M. P. Robillard，“检测脆弱注释”，载于 2017 年第 32 届 IEEE/ACM自动化软件工程国际会议，ASE’17，第 112–122 页，2017 年 10 月。[15] M. J. Howard、S. Gupta、L. Pollock 和 K. Vijay-Shanker，“从注释—代码映射中自动挖掘基于软件的语义相似词”，载于 2013 年第 10 届软件仓库挖掘工作会议，MSR’13，第 377–386 页，2013 年 5 月。[16] C. Treude 和 M. P. Robillard，“利用 Stack Overflow 的见解增强 API文档”，载于第 38 届国际软件工程会议论文集，ICSE ’16（美国纽约州纽约市），第 392–403 页，2016年。[17] G. Petrosyan、M. P. Robillard 和 R. De Mori，“使用文本分类发现解释 API类型的信息”，载于第 37 届国际软件工程会议论文集——第 1 卷，ICSE ’15（意大利佛罗伦萨），第 869–879 页，2015年。[18] B. Li、C. Vendome、M. Linares-V´asquez、D. Poshyvanyk 和 N. A. Kraft，“自动记录单元测试用例”，载于 2016 年 IEEE国际软件测试、验证与确认会议，ICST’16，第 341–352 页，2016 年 4 月。[19] L. Moreno、W. Bandara、S. Haiduc 和 A. Marcus，“缺陷报告词汇与源代码之间的关系”，载于 2013 年 IEEE软件维护国际会议，ICSM’13，第 452–455 页，2013 年 9 月。[20] K. Moran、M. Linares-V´asquez、C. Bernal-C´ardenas 和 D. Poshyvanyk，“为安卓应用自动补全缺陷报告”，载于 2015 年第 10 届软件工程基础联合会议论文集，FSE’15（意大利贝加莫），第 673–686 页，2015年。

[21] L. Moreno、G. Bavota、M. D. Penta、R. Oliveto、A. Marcus 和 G. Canfora，《Arena：一种自动生成发行说明的方法》，IEEE软件工程汇刊，第43卷，第106–127页，2017年2月。[22] L. Moreno、G. Bavota、M. Di Penta、R. Oliveto、A. Marcus 和 G. Canfora，《自动生成发行说明》，载于第22届ACM SIGSOFT软件工程基础国际研讨会论文集，FSE’14，第484–495页，2014年。[23] S. Jiang、A. Armaly 和 C. McMillan，《使用神经机器翻译根据差异自动生成提交信息》，载于2017年第32届IEEE/ACM自动化软件工程国际会议，ASE’17，第135–146页，2017年10月。[24] L. F. Cort´es-Coy、M. Linares-V´asquez、J. Aponte 和 D. Poshyvanyk，《通过摘要概述源代码变更自动生成提交信息》，载于2014年第14届IEEE国际源代码分析与操作工作会议，SCAM’14，2014年9月。[25] P. Chatterjee、B. Gause、H. Hedinger 和 L. Pollock，《从研究论文中提取代码片段及其描述》，载于第14届软件仓库挖掘国际会议论文集，MSR ’17，（阿根廷布宜诺斯艾利斯），第91–101页，2017年。[26] M. Linares-V´asquez、B. Li、C. Vendome 和 D. Poshyvanyk，《记录以数据库为中心的应用程序中的数据库用法和模式约束》，载于第25届软件测试与分析国际研讨会论文集，ISSTA 2016，（美国纽约州纽约市），第270–281页，2016年。[27] T. J. Biggerstaff、B. G. Mitbander 和 D. E. Webster，《程序理解与概念分配问题》，Commun. ACM，第37卷，第72–82页，1994年5月。[28] J. Guo、J. Cheng 和 J. Cleland-Huang，《使用深度学习技术增强语义的软件可追溯性》，载于第39届国际软件工程会议论文集，ICSE ’17，（美国新泽西州皮斯卡塔韦），第3–14页，2017年。[29] M. P. Robillard、A. Marcus、C. Treude、G. Bavota、O. Chaparro、N. Ernst、M. A. Gerosa、M. Godfrey、M. Lanza、M. Linares-V´asquez、G. C. Murphy、L. Moreno、D. Shepherd 和 E. Wong，《按需开发者文档》，载于2017年IEEE国际软件维护与演化会议，ICSME’17，第479–483页，2017年9月。[30] 《安卓赶超Windows，成为最受欢迎的操作系统 - https://www.cnet.co
m/news/android-most-popular-os-beats-windows-statcounter/。》[31] K. Moran、C. Bernal-C´ardenas、M. Linares-V´asquez 和 D. Poshy- vanyk，《克服语言二分法：迈向有效的移动应用开发程序理解》，载于第26届IEEE国际程序理解会议，2018年5月。即将发表。[32] A. M. Memon，《图形用户界面测试综合框架》。博士论文，2001年。导师：Mary Lou Soffa 和 Martha Pollock；委员会成员：Rajiv Gupta教授（亚利桑那大学）、Adele E. Howe教授（科罗拉多州立大学）、Lori Pollock教授（特拉华大学）。[33] A. Solem，《设计真正会被使用的计算机文档：理解计算机用户态度》，载于第4届系统文档年度国际会议论文集，SIGDOC ’85，（美国纽约州纽约市），第55–56页，计算机协会，1986年。[34] A. Alshayban、I. Ahmed 和 S. Malek，《安卓应用中的可访问性问题：现状、观点及改进方向》，载于第42届国际软件工程会议论文集，ICSE 2020，（美国纽约州纽约市），即将发表，ACM，2020年。[35] B. Deka、Z. Huang、C. Franzen、J. Hibschman、D. Afergan、Y. Li、J. Nichols 和 R. Kumar，《Rico：用于构建数据驱动设计应用的移动应用数据集》，载于第30届用户界面软件与技术年度研讨会论文集，UIST ’17，2017年。[36] K. Moran、C. Bernal-Cardenas、M. Curcio、R. Bonett 和 D. Poshy- vanyk，《ReDraw数据集：一组安卓屏幕截图、GUI元数据及GUI组件标注图像》，2019年1月。由美国国家科学基金会资助（拨款号CCF-1815186）。[37] M. Hodosh、P. Young 和 J. Hockenmaier，《将图像描述构建为排序任务：数据、模型和评估指标》，J. Artif. Int. Res.，第47卷，第1期，第853–899页，2013年。[38] T. Y. Lin、M. Maire、S. Belongie、J. Hays、P. Perona、D. Ramanan、P. Doll´ar 和 C. L. Zitnick，《Microsoft COCO：上下文中的常见物体》，载于《计算机科学讲义》（包括《人工智能讲义》和《生物信息学讲义》子系列），第8693卷（LNCS），第740–755页，2014年。

[39] 《在线附录 http://sagelab.io/Clarity/。》[40] A. Karpathy 和 L. Fei-Fei，《用于生成图像描述的深度视觉语义对齐》，IEEE Trans. Pattern Anal. Mach. Intell.，第39卷，第664–676页，2017年4月。[41] O. Vinyals、A. Toshev、S. Bengio 和 D. Erhan，《展示与讲述：从2015年MSCOCO图像描述挑战赛中汲取的经验》，IEEE Trans. Pattern Anal. Mach. Intell.，第39卷，第652–663页，2017年4月。[42] K. Xu、J. Ba、R. Kiros、K. Cho、A. C. Courville、R. Salakhutdinov、R. S. Zemel 和 Y. Bengio，《展示、关注与讲述：结合视觉注意力的神经图像描述生成》，载于国际机器学习会议，2015年。[43] A. Krizhevsky、I. Sutskever 和 G. E. Hinton，《使用深度卷积神经网络进行ImageNet分类》，载于《神经信息处理系统进展25》（F. Pereira、C. J. C. Burges、L. Bottou 和 K. Q. Weinberger 编），第1097–1105页，2012年。[44] K. Simonyan 和 A. Zisserman，《用于大规模图像识别的超深卷积网络》，国际学习表征会议，第abs/1409.1556卷，2014年。[45] C. Szegedy、W. Liu、Y. Jia、P. Sermanet、S. Reed、D. Anguelov、D. Erhan、V. Vanhoucke 和 A. Rabinovich，《通过卷积不断深入》，载于计算机视觉与模式识别会议，2015年。[46] S. Hochreiter 和 J. Schmidhuber，《长短期记忆》，Neural Comput.，第9卷，第8期，第1735–1780页，1997年。[47] M. Schuster 和 K. Paliwal，《双向循环神经网络》，Trans. Sig. Proc.，第45卷，第2673–2681页，1997年11月。[48] K. P. Moran、C. Bernal-C´ardenas、M. Curcio、R. Bonett 和 D. Poshy- vanyk，《基于机器学习的移动应用图形用户界面原型设计》，IEEE软件工程汇刊，第1 1页，2018年 – 。[49] D. M. Blei 、 A. Y. Ng 和 M. I. Jordan，《潜在狄利克雷分配》，J. Mach. Learn. Res.，第3卷，第993–1022页，2003年3月。[50] I. Porteous、D. Newman、A. Ihler、A. Asuncion、P. Smyth 和 M. Welling，《用于潜在狄利克雷分配的快速塌缩吉布斯采样》，载于第14届ACM SIGKDD知识发现与数据挖掘国际会议论文集，KDD '08，（美国纽约州纽约市），第569–577页，2008年。[51] A. Hindle、E. T. Barr、Z. Su、M. Gabel 和 P. Devanbu，《论软件的自然性》，载于国际软件工程会议，ICSE'12，第837–847页，2012年。[52] M. Rahman、D. Palani 和 P. Rigby，《重新探讨自然软件》，载于第41届国际软件工程会议伴随会议论文集，ICSE '19，（加拿大魁北克省蒙特利尔），即将发表，2019年。[53] 《MIT语言建模工具包 https://github.com/mitlm/mitlm。》[54] P. Koehn，《统计机器翻译》，第1版，2010年。[55] 《谷歌的im2txt实现 https://github.com/tensorflow/models/ tree/master/research/im2txt。》[56] 《Neuraltalk2实现 https://github.com/karpathy/neuraltalk2。》[57] 《谷歌序列到序列框架 https://github.com/google/seq2seq。》[58] C. Szegedy、V. Vanhoucke、S. Ioffe、J. Shlens 和 Z. Wojna，《重新思考用于计算机视觉的Inception架构》，载于2016年IEEE计算机视觉与模式识别会议，2016年6月。[59] O. Russakovsky、J. Deng、H. Su、J. Krause、S. Satheesh、S. Ma、Z. Huang、A. Karpathy、A. Khosla、M. Bernstein、A. C. Berg 和 L. Fei-Fei，《ImageNet大规模视觉识别挑战赛》，国际计算机视觉期刊，第115卷，252，2015年3，211第。– 。[60] 《展示、关注与讲述实现 https://github.com/coldmanck/ show-attend-and-tell。》[61] L. Sha、L. Mou、T. Liu、P. Poupart、S. Li、B. Chang 和 Z. Sui，《基于顺序规划的神经文本生成：从结构化数据出发》，The 第三十一届*AAAI*人工智能大会（*AAAI'17*），第 abs/1709.00155 卷，2017年。[62] D. Britz、A. Goldie、T. Luong 和 Q. Le，《神经机器翻译架构的广泛探索》，*ArXiv* 预印本，2017年3月。[63] K. Papineni、S. Roukos、T. Ward 和 W.-J. Zhu，《Bleu：一种机器翻译自动评估方法》，载于计算语言学协会第四十届年会论文集，ACL '02，（美国宾夕法尼亚州斯特劳兹堡），第 311–318 页，2002年。[64] M. Z. Hossain、F. Sohel、M. F. Shiratuddin 和 H. Laga，《用于图像描述的深度学习综合调查》，*ACM* 计算机调查，第 51 卷，第 118:1–118:36 页，2019年2月。[65] S. Chen、L. Fan、C. Chen、T. Su、W. Li、Y. Liu 和 L. Xu，《Storydroid：为安卓应用自动生成故事板》，载于第四十一届国际软件工程会议论文集，ICSE '19，（美国新泽西州皮斯卡塔韦），第 596–607 页，IEEE出版社，2019年。[66] X. Zhang、L. de Greef、A. Swearngin、S. White、K. Murray、L. Yu、Q. Shan、J. Nichols、J. Wu、C. Fleizach、A. Everitt 和 J. P. Bigham，屏幕识别：从像素为移动应用创建无障碍元数据。美国纽约州纽约：计算机协会，2021年。[67] J. Chen、C. Chen、Z. Xing、X. Xu、L. Zhu、G. Li 和 J. Wang，《让应用无盲区：利用深度学习预测移动图形用户界面组件的自然语言标签》，载于第四十二届国际软件工程会议论文集，ICSE 2020，（美国纽约州纽约），页码待定，ACM，2020年。[68] K. Moran、C. Watson、J. Hoskins、G. Purnell 和 D. Poshyvanyk，《检测并总结不断演进的移动应用中的图形用户界面变更》，载于第三十三届*ACM/IEEE*自动化软件工程国际会议论文集，ASE 2018，（美国纽约州纽约），第 543–553 页，2018年。[69] K. Moran、B. Li、C. Bernal-C´ardenas、D. Jelf 和 D. Poshyvanyk，《自动报告移动应用中的图形用户界面设计违规》，载于第四十届国际软件工程会议伴随会议论文集，ICSE '18，（瑞典哥德堡），页码待定，2018年。[70] T. A. Nguyen 和 C. Csallner，《使用 REMAUI 对移动应用用户界面进行逆向工程》，载于*2015*年第*30*届*IEEE/ACM*自动化软件工程国际会议论文集，ASE'15，（美国华盛顿特区），第 248–259 页，2015年。[71] T. Beltramelli，《Pix2code：从图形用户界面截图生成代码》，*CoRR*，第 abs/1705.07962 卷，2017年。[72] Q. Chen 和 M. Zhou，《用于源代码检索与摘要生成的神经框架》，载于第三十三届*ACM/IEEE*自动化软件工程国际会议论文集，ASE 2018，（美国纽约州纽约），第 826–831 页，ACM，2018年。[73] B. Andow、A. Acharya、D. Li、W. Enck、K. Singh 和 T. Xie，《UiRef：分析安卓应用中的敏感用户输入》，载于第十届*ACM*无线与移动网络安全和隐私会议论文集，WiSec'17，34，2017 23（美国纽约州纽约），第 – 页。[74] T. F. Liu、M. Craft、J. Situ、E. Yumer、R. Mech 和 R. Kumar，《学习移动应用的设计语义》，载于第三十一届*ACM*用户界面软件与技术年度研讨会，UIST '18，（美国纽约州纽约），第 569–579 页，2018年。[75] X. Xiao、X. Wang、Z. Cao、H. Wang 和 P. Gao，《Iconintent：基于图标分类自动识别安卓应用中的敏感用户界面控件》，载于第四十一届国际软件工程会议伴随会议论文集，ICSE '19，（加拿大魁北克省蒙特利尔），页码待定，2019年。