变压器模型在社交机器人检测中的比较分析

Rohan Veit and Michael Lones

Department of Computer Science, Heriot-Watt University, Edinburgh, UK
rv2009@hw.ac.uk, m.lones@hw.ac.uk

摘要 社交媒体已成为当今社会的关键交流媒介。这一认识促使许多方面雇佣人工用户（或机器人）误导他人相信不实之词或采取对他们有利的行为。先进的文本生成工具，如大型语言模型，进一步加剧了这个问题。本文旨在比较基于编码器和解码器转换器的机器检测模型的有效性。开发了管道来评估这些分类器的性能，结果表明基于编码器的分类器表现出更高的准确性和鲁棒性。然而，基于解码器的模型通过任务特定对齐显示出更大的适应能力，暗示在不同应用场景中具有更多的泛化潜力以及更优越的表现。这些发现有助于防止数字环境被操纵的同时保护在线讨论的完整性。

Keywords: 机器人检测，变换器，机器学习

1 介绍

随着互联网的发展，其上的服务也随之演变；曾经无害的平台如社交媒体已从与朋友联系和自我表达的机会转变为一种向大众传播可信度参差不齐的信息的工具，塑造公众信念甚至影响现实世界事件 [2]。像 X 和 Facebook 这样的网站使个人、组织或政府能够以前所未有的范围进行沟通。这一因素巩固了社交媒体作为当今叙事形成支柱的地位，它们对人际交流有着“深远的影响”[18]。

社交媒体的影响尤其体现在其放大叙事的能力上。展示极化和煽动性观点的帖子会受到算法的推广，这些算法更倾向于关注互动而非展示中立信息的有见地的文章。一个这样的例子是 COVID-19 大流行，其间社交媒体助长了误导性信息的传播，导致疫苗拒绝和口罩抵制 [8]。因此，很明显必须采取措施防止虚假信息污染数字空间；识别虚假用户是这一过程的重要组成部分。

一个“机器人”是平台上的一个虚拟用户，它表现得像真实用户一样。当它们首次出现时，由于广播信息的重复性，通常很容易被检测到，这意味

着可以使用基于规则的方法（如内容阈值，比如重复消息限制 [20]）来消除它们。然而，现代机器人已经进化到利用先进的自然语言处理工具，例如大型语言模型，使它们能够更好地融入真实用户中并规避现有的检测措施 [7]。

在这项研究中，将在不同的测试案例下评估多种机器人检测管道，以分析它们的优点、缺点和理想应用。

2 背景

本节将简要提供社交媒体中机器人检测的基础背景以及将在本研究中调查的相关技术。它将涵盖所选案例研究的合理性，并评估最近相关研究及其不足之处，以突出应进行调查的研究空白。这些组成部分将为 section 3 中详细描述的方法奠定基础。

2.1 机器人

在 [12] 中，社交媒体机器人被定义为一种计算机算法，能够生成内容并与用户互动。虽然这本身并不具有恶意（通过现已不存在的账户提醒我推文证明，该账户允许用户在指定时间后自动收到帖子提醒），这种协调数百甚至数千个账户的能力可能会导致诸如垃圾邮件活动 [21]、内容的人工推广 [1]，甚至是通过一种名为 “伪装草根” [15] 的策略操纵政治事件等恶意用途。

2.2 变压器模型

Transformer 模型已成为机器学习任务的前沿技术之一。与传统的前馈网络不同，Transformer 不是按顺序处理每个输入标记，而是使用自注意力来建模标记之间的关系，无需像以前的长短期记忆（LSTM）网络中常见的那种递归层。根据其架构及其生成输出的方式，可以将它们分类为仅编码器、仅解码器和基于编解码器的模型。仅编码器 Transformer 专用于通过双向自注意力理解并表示输入，其中每个标记关注（关联）输入中的所有其他标记。这些模型在需要深入理解输入的任务中表现出色，如文本分类、情感分析和特征提取。一些这样的 Transformer 示例包括 BERT、RoBERTa 和 DistilBERT。

仅使用解码器的变压器模型采用单向自注意力机制，其中每个标记只能关注输入中在其之前的标记。这使得它们非常适合生成任务，在这些任务中，模型必须从给定提示依次预测下一个最可能的单词/标记。然而，通过

提示工程（修改提供给模型的输入）的过程可以对其进行修改以完成其他任务。示例用例包括文本生成和补全。知名模型包括 OpenAI 的 GPT 系列和谷歌的 Gemini。

编码器-解码器架构通过使用编码器处理输入和解码器生成输出，结合了这两种方法。这使其非常适合需要复杂理解和生成的任务，如翻译或摘要。这种方法的一些示例是 BART[9] 和 T5[14]。

2.3 数据集

由于其庞大的用户基础和范围，X（前身为 Twitter）是机器人检测实验中常用的案例研究。从现在起我们使用“Twitter”这个名字是因为这遵循了之前及近期学术研究中的命名惯例。Twitter 也是受机器人兴起影响最大的社交媒体平台之一，据估计截至 2017 年，有 9–15%的用户为机器人账户[19]。其作为案例研究的重要性还进一步体现在它被用于大规模机器人攻击中，例如为了政治干预 [15]。

TwiBot-22[5] 是一个建立在前辈数据集（Cresci-2015[3], Varol-2017[19], TwiBot-20[6]）之上的数据集。它由 1,000,000 名用户组成，使其几乎是第二大社交媒体机器人数据集 TwiBot-20 的五倍大，后者包含 229,000 名用户。TwiBot-22 还使用了一种分布多样性偏差算法来捕捉广泛的人口特征，同时保持完整的图结构以捕获数据集中用户之间的关系。然后，通过结合 8 个手工标签和 7 个具有竞争力的基于特征的分类器生成用户的标签，这些标签随后使用 Snorkel[16] 进行细化以生成最终的数据标注。这些标签比 TwiBot-20 准确 10.5%。

2.4 Transformer 模型用于检测机器人

各种研究试图使用变换器来区分社交网络中的社交机器人和真实用户。

在 [10] 中，使用了变压器来创建一个稳健且与语言无关的机器人检测系统，采用 BERT 和 RoBERTa 编码用户账户的文本特征。他们使用了 BERT 和 RoBERTa 的多语言基础版本生成输入特征（如用户名、描述和语言指示）的 768 维和 1024 维嵌入向量表示，并将这些与额外的用户元数据拼接，最终由一个名为 Bot-DenseNet 的自定义深度神经网络进行分类，在激活函数中选择了 Scaled Linear Exponential Linear Unit (SELU)，而不是传统的 ReLU 函数。为了解决提供的数据集不平衡的问题，采用了分层抽样。

这种方法后来通过使用各种基于变压器的方法进行了改进。在 [17] 中，预训练语言模型（PLMs）——包括各种 BERT 模型和仅编码器模型 GPT-3——在特定机器人的数据集上进行了微调（TweepFake[4] 和狐狸 8-23），然后输入到一个经典的前馈神经网络中，该网络使用这些 PLMs 的输出来分类用户。这些模型能够通过其动态嵌入能力考虑社交媒体文本中的语境细微差别——包括数字频率、用户提及、标签、URL、表情符号和主题分析——提供了比静态嵌入方法如 GloVe[13] 和 Word2Vec[11] 更大的灵活性和准确性。

仅解码器的变压器也被证明有助于提升机器人分类任务。一项研究详细描述了一种解决方案，其中三个大型语言模型（LLMs）在 1000 个标记样本上进行了“指令微调”。这包括向 LLMs 提供之前解决方案中使用的标准用户元数据和文本数据，以及用户的邻域子集。然后对提供的提示进行调整，以提供两类的上下文示例。该方法比现有的机器人检测措施提高了 9.1%[7]。此外，LLM 显示出对社交网络固有的图结构有合理的理解，并能够建议用户关注列表中的添加和移除操作，使其不易受到机器人检测方法的影响。总之，这表明尽管解码器变压器在机器人检测方面有很多优势，但它们渐进的复杂性也是一把双刃剑，因为这些进步也可能被恶意行为者利用。

先前的工作表明，变换器在机器人检测方面有很大的潜力。然而，对于架构和能力之间的联系理解甚少。特别是，尽管像 GPT-4 这样的强大解码器模型的出现引起了人们使用这些模型进行机器人检测的兴趣，但很少有研究探讨它们与更为成熟的编码器基础模型相比的效果如何。在这项工作中，我们解决了这一问题。

3 方法论

3.1 数据预处理

首先，数据来源于 TwiBot-22 数据集。由于开发环境的计算资源有限，仅提取了前 10%的用户消息以保持一个可操作的数据集大小。从这些数据中，得到了 5 个包含 1,000 用户的训练/测试分区，用于五折交叉验证。

在数据清洗完成后，接下来需要定义将提供给模型进行推理的结构化数据类型。为此，使用了 Python 库 Pydantic 来定义数据模式并进行强大的数据验证。首先，从清理后的 JSON 数据中定义了用户和推文的数据模式。通过这些模式，可以轻松实例化相应的 Python 对象，并进行了严格的类型验证以确保处理好格式错误或缺失的字段。

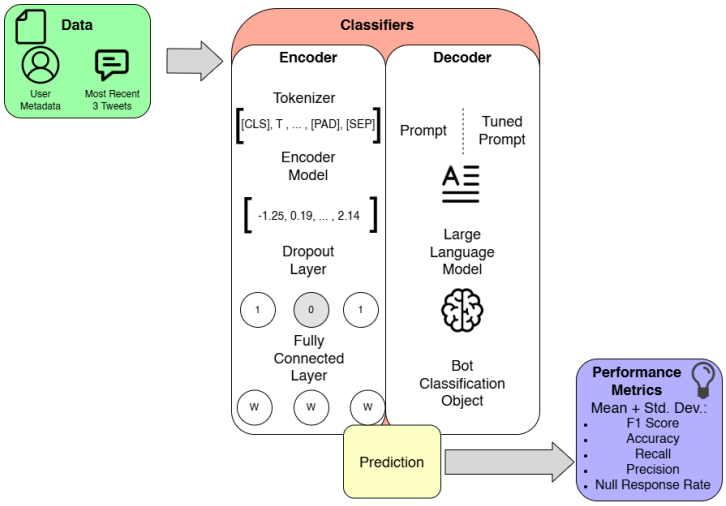

图 1. 方法总结图

3.2 解码器管道创建

数据预处理之后，用于评估解码器模型的管道架构被搭建起来。该管道旨在促进结构化数据处理、模型推理，并最终对输入的用户是否可能是人工账户进行二元预测。

Ollama 被选中用于实现模型推理。Ollama 是一个设计用来在本地或基于云的环境中简化和高效运行大语言模型（LLMs）推理的框架，为用户提供了简单且有效的方法在其自己的硬件上运行模型。这导致了延迟减少以及对外部 API 服务的依赖降低。这意味着希望部署此管道或类似管道的用户可以确信模型的正常运行时间，并对模型类型和配置有更大的控制权。值得注意的是，Ollama 只支持一个精选的大语言模型库，并且由于该管道需要结构化数据作为输入和输出（以方便在现实世界中的应用），这进一步限制了所选大语言模型为那些支持工具使用的情况。用于分析的选定大语言模型包括：LLama3.1、Mistral 和 Qwen2.5。

为了便于在现实世界中的实施，选择让管道以结构化对象的形式提供响应。这意味着使用了 PydanticAI，这是一个基于 Pydantic 提供的数据类基础模型接口的 Python 框架，允许创建强制执行由数据模式定义的严格输入和输出格式的代理。这有助于开发需要可靠输出的应用程序，在使用自动决策的系统中很常见，例如社交机器人的检测。最终的 PydanticAI 代理被定义为接受字符串段落格式（以支持提示工程）作为输入，并将一个定义好的

“机器人检测响应”数据类作为输出：该对象包括一个布尔值 ’is_bot’ 指示器以及选定结果的原因，提供模型推理可见性并更好地引导模型产生合理的回应。

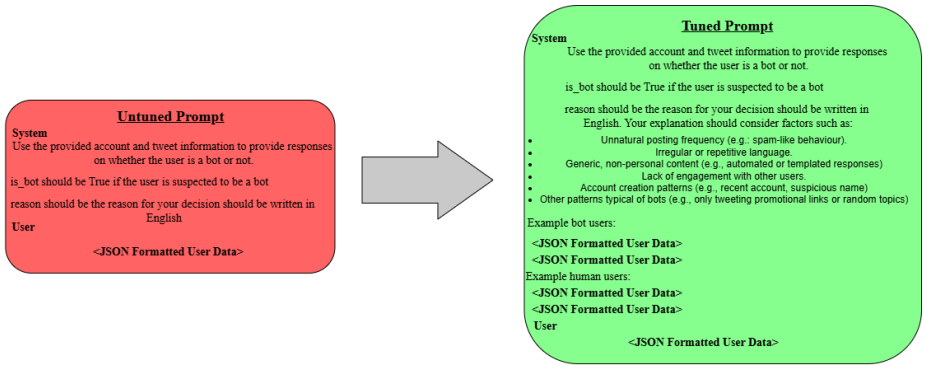

图 2. 未调优和已调优的解码器提示

最后，为了便于比较分析分类器对特定任务对齐的响应能力，采用了提示工程。具体来说，使用了少样本学习来为模型提供示例用户及其标签，以帮助引导模型将这种逻辑外推到未见过的例子上。为此，从五个测试分割之外抽取了四个示例用户，并将其插入到模型提示中的“示例机器人/人类用户”部分。最后，任务规范也被用来帮助指导模型基于特定的用户特征（即少量帖子或近期账号创建）进行决策。

在数据和解码器管道创建之后，编码器管道的开发开始。该管道负责将原始文本数据转换为与编码器模型兼容的格式，其输出可用于分类输入的数据是否可能属于人工用户。

3.3 编码管道创建

编码器管道是使用 HuggingFace 的 Transformers 库和 PyTorch 开发的，这两个广泛采用的框架用于实现机器学习模型。首先——类似于解码器管道的创建——必须选择内部编码器模型。鉴于之前的文献，选定用于分析的结果模型为：DistilBERT、BERT 和 RoBERTa。

选定内部模型后，就开始设计分类器的工作；首先，通过 HuggingFace 的 `Bert` 模型接口实现了架构的编码部分，该接口允许仅通过更改提供的模型名称就能轻松从由 HuggingFace 维护的模型仓库导入模型。然后，使用相

同的接口可以轻松地在提供的模型上进行前向传递，生成代表输入文本用户数据的语义嵌入。这些嵌入随后被送入一个 dropout 层以规范化输出数据并防止模型过拟合，确保编码器模型不会因为训练所需的子集数据而相对于解码器模型具有优势。最后，dropout 层馈送到一个全连接线性层，该层负责将嵌入表示映射到最终的决策形式，即为两个类别（机器人或人类）之一生成置信分数。

数据管道已经实现，可以很容易地导入到编码器实验中。然后，再次使用 HuggingFace 来获取一个分词器（特定于为该实验选择的编码器模型），该分词器将用于创建一个 PyTorch 数据集类，负责处理数据的批处理和混洗。

与解码器管道中的大型语言模型组件预训练不同，编码器管道的神经网络组件需要这一阶段来确定所提供的输入示例中哪些特征对分类是重要的。为了训练编码器分类器，该模型被提供了一个示例以进行前向传播并做出预测。然后，使用 PyTorch 的交叉熵损失准则计算模型预测的损失，并利用 PyTorch 的自动微分系统来进行反向传播。这会根据模型参数计算损失的梯度，然后通过带有权重衰减修复的 Adam 算法优化器更新模型参数。

最后，类似于解码器管道，进行了模型微调。这包括形成一个需要修改的超参数列表，以及它们将要分析的值。为微调选择的超参数是：训练周期、替代优化器和学习率。使用这些，进行了一次详尽的网格搜索，以找到最佳性能的超参数值组合，用于细调比较分析（详见 section 4.4）。

4 结果与观察

本节介绍了 section 3 中概述的实验结果。每个子部分详细说明了解码器和编码器分类器在不同条件下的性能，并通过准确性、精度、召回率、F-1 分数和空响应率来评估其性能。此外，还提供了这些指标的标准偏差以评估模型的变化性。

4.1 实验 1：部分特征集

本次实验通过使用仅包含账户元数据的有限特征集来评估解码器和编码器模型，从而建立了分类器性能的基础线。本实验旨在评估两种模型架构所展现的初始分类能力，并作为进一步实验的参考点。

结果如 table 1 所示，每个架构每项指标的最佳分类器被加粗显示。这表明编码模型在所有指标上都优于解码模型。这可能是由于编码模型中对下

模型	准确性	精度	回忆	F1 分数	空响应率
解码器模型					
Llama3.1:8b	0.505 ± 0.014	0.530 ± 0.014	**0.606** ± 0.016	0.566 ± 0.015	0.051 ± 0.004
Mistral:7b	0.299 ± 0.006	**0.691** ± 0.053	0.177 ± 0.027	0.281 ± 0.036	0.453 ± 0.006
Qwen2.5:7b	**0.569** ± 0.020	0.585 ± 0.020	0.580 ± 0.037	**0.582** ± 0.028	**0.023** ± 0.006
编码器模型					
BERT	0.759 ± 0.017	0.766 ± 0.029	0.749 ± 0.035	0.757 ± 0.016	**0.000** ± 0.000
DistilBERT	0.759 ± 0.015	**0.767** ± 0.022	0.744 ± 0.027	0.755 ± 0.015	**0.000** ± 0.000
RoBERTa	**0.770** ± 0.008	0.757 ± 0.011	**0.796** ± 0.023	**0.776** ± 0.009	**0.000** ± 0.000

表 1. 实验 1 结果（平均值 ± 标准差）

游分类器的显式训练，提供了更好的任务领域对齐。这表明通过利用监督学习，性能可以显著提高。然而，结果也展示了解码模型在没有额外训练的情况下某种程度上的操作能力。

所有模型也展示了较低的标准差，表明训练运行的一致性。然而，解码器模型明显产生了一些空响应，即它们没有为测试集中的所有样本输出类别标签。这表明它们可能不太适合需要二元响应的实际部署。

4.2 实验 2A：丰富的特征集

本次实验旨在评估模型在增强特征集下的性能。本实验的主要目标是评估输入特征的变化如何影响每个模型的性能指标，识别哪些模型可能对特征修改敏感。为了系统地衡量这些效果，table 2 将实验 2A 的结果与实验 1 的结果进行比较，突出显示性能指标的相对增益或损失。

一般来说，当使用完整特征集时，解码器模型的性能下降。平均而言，所有解码器模型的准确率都有所降低，其中最大的降幅出现在 Qwen2.5 中。该模型在召回率和 F1-Score 方面也显示出显著减少。此外，所有解码器模型的空响应率有所增加，表明更完整的特征集影响了分类器的鲁棒性。编码器模型的表现更加稳定，大多数指标的变化相对较小。BERT 保持了其准确率，并在四个记录指标中的三个中得到了改进。RoBERTa 在所有记录的指标中都有所下降。

性能的降低表明，额外的文本数据（以用户最近三条推文的形式）是反生产力的，为确定对标签有贡献的特征的过程增加了噪音。尽管可能令人惊讶，这一观察结果与其他研究是一致的，例如 [7]，其中向用户元数据添加额

模型	准确性 Δ	精度 Δ	回忆 Δ	F1 分数 Δ	空响应率 Δ
解码器模型					
Llama3.1:8b	**-0.040**	**+0.001**	**+0.036**	**+0.016**	+0.081
Mistral:7b	-0.053	-0.072	-0.060	-0.0854	**+0.079**
Qwen2.5:7b	-0.156	-0.010	-0.131	-0.079	+0.228
编码器模型					
BERT	**+0.000**	-0.011	**+0.018**	**+0.004**	**+0.000**
DistilBERT	-0.004	-0.016	+0.020	+0.002	**+0.000**
RoBERTa	-0.015	**-0.008**	-0.028	-0.017	**+0.000**

表 2. 实验 2A 结果（与实验 1 的平均差异）

模型	准确率	精度	回忆	F1 分数	空响应率
Qwen2.5 3b	0.520 ± 0.015	0.594 ± 0.037	0.255 ± 0.018	0.357 ± 0.024	0.040 ± 0.003
Qwen2.5 7b	0.413 ± 0.019	0.573 ± 0.027	**0.448** ± 0.023	**0.503** ± 0.025	0.251 ± 0.017
Qwen2.5 14b	**0.558** ± 0.014	**0.660** ± 0.034	0.246 ± 0.019	0.358 ± 0.024	**0.002** ± 0.001

表 3. 实验 3 结果（平均值 ± 标准差）

外文本降低了 Mistral、Llama2 和 ChatGPT 的性能。然而，在我们的结果中，编码器模型显示出较低的敏感性。

4.3 实验 2B：模型大小比较

本次实验通过重复实验 2A，使用了 30 亿、70 亿和 140 亿参数的 Qwen2.5 变体来研究模型规模的作用，其中 70 亿参数版本在实验 1 中是表现最强的编码器模型。结果如 table 3 所示。这表明较大的 140 亿参数模型相较于 70 亿参数模型，在准确率和空响应率上有所提升。然而，其性能仍低于使用了减少特征集时的 70 亿参数模型的表现，这表明更多的参数并不能使其从额外提供的信息中获益。

此外，较大的模型的召回率和 F1 分数有所降低。通过检查具有 70 亿参数的模型的混淆矩阵，这是因为中间模型似乎比 30 亿和 140 亿参数变体更倾向于预测 'bot'，它有 40%的时间预测为 'bot'，而 Qwen2.5:14b 仅选择 'bot' 的比例为 18%。这可能是由于具有 70 亿参数的模型是 Qwen2.5 的 "默认" 版本，暗示它可能经过了进一步的优化。

模型	准确性	精度	回忆	F1 分数	空响应率
Qwen2.5:14b	0.655 ± 0.022	0.670 ± 0.023	0.617 ± 0.025	0.642 ± 0.023	0.001 ± 0.001
RoBERTa	**0.774** ± 0.004	**0.763** ± 0.012	**0.794** ± 0.026	**0.778** ± 0.019	**0.000** ± 0.000

表 4. 实验 4 结果（平均值 ± 标准差）

4.4 实验 3：调整后的分类器

本实验的目的是比较不同架构对特定任务调优的响应。选定的编码器和解码器模型是在实验 2 中表现最好的那些。为了确保公平比较，两个模型都使用了来自实验 2 的同一特征集。为了调整编码器分类器，采用了超参数微调，而对于解码器分类器，则采用了提示工程。

结果如下所示 table 4。正如所展示的，解码器分类器对调优更为敏感，其准确性从实验 2B 中提高了 9.7%，表明提示工程在此情境下是有益的。不过，编码器分类器的表现仍然优于它，尽管与实验 2A 相比性能仅略微提升。无论如何，它在所有实验中均达到了最高的平均准确率、精度和 F1 分数。另一个值得注意的现象是，在调优后，空响应率降至所有实验中的最低点，表明任务特定的调优也有助于增强模型的鲁棒性。需要指出的是，这种调优方法不需要修改模型参数，这展示了解码器模型在提供任务特定调优可访问性方面的提升。

5 限制

由于数据的可用性，这项工作仅在一个数据集 TwiBot-22 上进行了评估。这可能会限制结论的普遍性。此外，虽然对精简和增强的功能集进行了分析，但 TwiBot-22 捕获的所有信息（如用户账户关系）并未被利用。最后，由于财务和时间成本的限制，仅评估了每种架构中的三个开源模型，并且模型大小的比较仅在实验环境约束范围内进行。

6 结论

本研究对解码器和编码器基础的社会机器人检测模型的有效性进行了深入分析，采用了多种类型的模型、规模及特征集。我们的结果显示，基于编码器的架构在所有评估指标上始终优于基于解码器的模型，表明对于可以实施监督学习的机器人检测任务，它们可能提供更优性能。基于解码器的方

法显示出不太令人满意的结果，但因底层模型易于替换而具有较大的改进潜力。基于解码器的分类器还表现出对通过提示工程进行特定任务调优方法的高度响应性，展示了其低成本改进的潜力。

在未来的工作中，我们计划：评估更大、更复杂的模型；评估一套更多样化的编码器（可能超出 BERT 家族，如 T5）和解码器模型；并且开发一个更全面的功能集以捕捉用户社交网络的更多特征。

References

[1] F. Benevenuto, T. Rodrigues, A. Veloso, J. Almeida, M. Gonçalves, and V. Almeida. Practical detection of spammers and content promoters in online video sharing systems. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, 42:688–701, 11 2011.

[2] A. Bovet and H. A. Makse. Influence of fake news in twitter during the 2016 us presidential election. *Nature Communications*, 10(1):7, 2019.

[3] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi. Fame for sale: Efficient detection of fake twitter followers. *Decision Support Systems*, 80:56–71, 2015.

[4] T. Fagni, F. Falchi, M. Gambini, A. Martella, and M. Tesconi. Tweepfake: About detecting deepfake tweets. *PLOS ONE*, 16(5):e0251415, May 2021.

[5] S. Feng, Z. Tan, H. Wan, N. Wang, Z. Chen, B. Zhang, Q. Zheng, W. Zhang, Z. Lei, S. Yang, X. Feng, Q. Zhang, H. Wang, Y. Liu, Y. Bai, H. Wang, Z. Cai, Y. Wang, L. Zheng, Z. Ma, J. Li, and M. Luo. Twibot-22: Towards graph-based twitter bot detection, 2023.

[6] S. Feng, H. Wan, N. Wang, J. Li, and M. Luo. Twibot-20: A comprehensive twitter bot detection benchmark. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM ' 21, page 4485 – 4494. ACM, Oct. 2021.

[7] S. Feng, H. Wan, N. Wang, Z. Tan, M. Luo, and Y. Tsvetkov. What does the bot say? opportunities and risks of large language models in social media bot detection, 2024.

[8] M. M. Ferreira Caceres, J. P. Sosa, J. A. Lawrence, C. Sestacovschi, A. Tidd-Johnson, M. H. U. Rasool, V. K. Gadamidi, S. Ozair, K. Pandav, C. Cuevas-Lou, M. Parrish, I. Rodriguez, and J. P. Fernandez. The impact of misinformation on the covid-19 pandemic. *AIMS public health.*, 9(2):262–277, 2022.

[9] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, 2019.

[10] D. Martín-Gutiérrez, G. Hernández-Peñaloza, A. B. Hernández, A. Lozano-Diez, and F. Álvarez. A deep learning approach for robust detection of bots in twitter using transformers. *IEEE Access*, 9:54591–54601, 2021.

[11] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space, 2013.

[12] M. Orabi, D. Mouheb, Z. Al Aghbari, and I. Kamel. Detection of bots in social media: A systematic review. *Information Processing & Management*, 57(4):102250, 2020.

[13] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[14] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023.

[15] J. Ratkiewicz, M. Conover, M. Meiss, B. Goncalves, A. Flammini, and F. Menczer. Detecting and tracking political abuse in social media. *Proceedings of the International AAAI Conference on Web and Social Media*, 5(1):297–304, Aug. 2021.

[16] A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, and C. Ré. Snorkel: rapid training data creation with weak supervision. *Proceedings of the VLDB Endowment*, 11(3):269 – 282, Nov. 2017.

[17] A. Sallah, E. Arbi Abdellaoui Alaoui, S. Agoujil, M. Ahmad Wani, M. Hammad, Y. Maleh, and A. A. Abd El-Latif. Fine-tuned under-

standing: Enhancing social bot detection with transformer-based classification. *IEEE Access*, 12:118250–118269, 2024.

[18] K. R. Subramanian. Influence of social media in interpersonal communication. *International journal of scientific progress and research*, 38(2):70–75, 2017.

[19] O. Varol, E. Ferrara, C. A. Davis, F. Menczer, and A. Flammini. Online human-bot interactions: Detection, estimation, and characterization, 2017.

[20] A. H. Wang. Don't follow me: Spam detection in twitter. In *2010 International Conference on Security and Cryptography (SECRYPT)*, pages 1–10, 2010.

[21] X. Zhang, S. Zhu, and W. Liang. Detecting spam and promoting campaigns in the twitter social network. In *Proceedings of the 2012 IEEE 12th International Conference on Data Mining*, ICDM '12, page 1194 – 1199, USA, 2012. IEEE Computer Society.