arXiv:2303.01576v1 [cs.HC] 2 Mar 2023

DeepSeer：通过状态抽象实现交互式循环神经网络解释与调试

Zhijie Wang
阿尔伯塔大学
埃德蒙顿，AB，加拿大
zhijie.wang@ualberta.ca

Yuheng Huang
阿尔伯塔大学
埃德蒙顿，AB，加拿大
yuheng18@ualberta.ca

Da Song
阿尔伯塔大学
埃德蒙顿，AB，加拿大
dsong4@ualberta.ca

Lei Ma
阿尔伯塔大学，加拿大
东京大学，日本
ma.lei@acm.org

Tianyi Zhang
普渡大学
美国印第安纳州西拉斐特
tianyi@purdue.edu



图 1：DeepSeer 是一款交互式工具，通过状态抽象帮助用户理解和调试循环神经网络模型。DeepSeer 通过提供全局解释 和局部解释，形成协同循环，帮助程序员理解循环神经网络模型。程序员可以使用 DeepSeer 探索全局解释，快速理解模型并发现潜在漏洞，然后深入局部解释，以情境化全局解释。程序员也可以根据局部解释调试特定实例，然后通过缩小视野并与全局解释进行比较，验证调试假设。

摘要

鉴于循环神经网络（RNNs）在处理序列数据方面表现优异，因此已被广泛应用于自然语言处理（NLP）任务。然而，由于循环神经网络固有的复杂性以及缺乏透明度，对循环神经网络进行解释和调试颇具挑战。尽管已有许多面向循环神经网络的可解释人工智能（XAI）技术被提出，但其中大多数仅支持局部解释，而非全局解释。本文介绍了 DeepSeer，这是一个交互式系统，通过多个紧密协同的视图，同时提供对循环神经网络行为的全局和局部解释，以帮助用户理解和调试模型。DeepSeer 的核心是一种状态抽象方法，它将循环神经网络模型中语义相似的隐藏状态进行归并，并将模型抽象为有限状态机。用户可以通过检查与每个状态相关联的文本模式来探索全局模型行为，并状态之间的转移。用户还可以通过检查给定输入的状态轨迹和中间预测结果，深入查看单个预测。一项涉及28名参与者的被试间用户研究表明，与一种流行的可解释人工智能技术 LIME 相比，使用 DeepSeer 的参与者对循环神经网络模型行为进行了更深入、更全面的评估，更准确地识别出错误预测的根本原因，并提出了更多可执行计划来提升模型性能。

ACM 分类概念

• **以人为中心的计算** → **交互式系统与工具**；• **计算方法** → **机器学习**；•**软件及其工程** → **软件测试与调试**。

关键词

可解释人工智能、模型调试、循环神经网络、可视化

特此授权用户免费制作本作品的全部或部分数字副本或纸质副本，用于个人或课堂使用，但前提是副本不得出于营利或商业利益而制作或分发，且副本必须保留本声明以及出现在首页的完整引文。本作品中由作者以外的其他方拥有版权的组件必须遵守其版权规定。注明出处的摘要使用是允许的。以其他方式复制、重新出版、发布到服务器或重新分发到列表，则须事先获得具体许可和/或支付费用。请通过 permissions@acm.org 申请许可。*CHI '23，2023年4月23日至28日*，德国汉堡
© 2023年版权归所有者/作者所有。出版权已授权给美国计算机协会。美国计算机协会 ISBN 978-1-4503-9421-5/23/04...$15.00
https://doi.org/10.1145/3544548.3580852

美国计算机协会参考文献格式：
Zhijie Wang、Yuheng Huang、Da Song、Lei Ma 和 Tianyi Zhang。2023年。DeepSeer：通过状态抽象实现交互式循环神经网络解释与调试。载于*2023年美国计算机协会人机交互大会论文集（CHI '23），2023年4月23日至28日*，德国汉堡。美国计算机协会，纽约，美国，20页。
https://doi.org/10.1145/3544548.3580852

警告：使用场景中的毒性检测示例包含一些可能会造成干扰的内容。

1 引言

深度神经网络（DNNs）凭借在现实世界中的复杂任务上出色的表现，正越来越多地应用于实践中，例如自动驾驶、 [49], 虚拟助手 [9], 和疾病诊断 [34]。深度学习系统的快速发展带来了机遇，同时也带来了挑战和担忧。其中一个主要担忧源于深度神经网络的可解释性 [36]。不同于决策逻辑以源代码形式手动编写的传统软件，DNN 模型包含大量通过非线性函数连接的神经元，其权重会从训练数据中自动学习。通过设置断点并检查运行时值，可以轻松检查和分析传统软件的内部状态。然而，DNN 模型的内部状态是高维向量，而不是符号值。仅通过检查这些向量，很难判断 DNN 模型学习了哪些模式，也难以解释模型为何做出特定预测。因此，DNNs 的这种内部复杂性和难以理解性带来了重大的调试挑战，也引发了人们对 DNNs 可信度和可靠性的担忧。

尽管机器学习（ML）、人机交互（HCI）和可视化领域近年来日益关注于提升深度神经网络（DNN）的可解释性，但许多现有技术将深度神经网络模型视为黑盒，并生成特征重要性等模型无关解释，却不揭示深度神经网络模型的内部工作机制 [32, 39, 53, 54]。虽然已有一些技术用于可视化深度神经网络模型中的隐藏状态，但其中许多技术都聚焦于卷积神经网络（CNNs） [8, 48, 57]。在本研究中，我们尤其关注循环神经网络（RNNs）。与其他类型的深度神经网络相比，循环神经网络能够处理长度可变的序列数据，例如文本和音频。循环架构赋予循环神经网络内部记忆，这已被证明有助于学习序列数据中的时间模式。然而，这种架构也给循环神经网络内部状态的可视化带来了挑战。不同于每层数量固定且层数固定的神经元的卷积神经网络，循环神经网络是无界的。此外，与卷积神经网络可视化中常见的分别处理每一层不同，研究循环神经网络单元的动态十分重要，即可视化一系列隐藏状态中所嵌入的时间模式。

本文介绍了 DeepSeer，这是一种交互式系统，可帮助模型开发者理解和调试循环神经网络模型。我们的核心洞见是将循环神经网络模型视为有状态系统。通过对语义相似的隐藏状态进行聚类和抽象，可以将循环神经网络模型表示为有限状态机（FSM），与原始循环神经网络模型相比，它更小，也更易于浏览。此外，与以往工作中直接可视化隐藏状态数值的做法不同， [59], DeepSeer 会将隐藏状态映射为更易于理解的表示——与这些状态相关的常见词语和短语。通过检查状态之间的状态转移，用户可以快速识别模型学到的时间模式。

为了评估 DeepSeer 的整体实用性，我们对 28 名在机器学习和循环神经网络方面经验水平各异的程序员开展了一项被试间用户研究。参与者使用一个预训练循环神经网络模型，完成模型理解任务和调试任务；他们使用的工具是 DeepSeer 或一种广受欢迎的可解释人工智能工具 LIME [53]。我们发现，在模型理解任务中，使用 DeepSeer 的参与者对模型行为给出了更有见地的回答，更准确地指出了模型局限性，并针对如何改进指定模型提出了更有用、更多样的建议。此外，在模型调试任务中，与使用 LIME 的参与者相比，使用 DeepSeer 的参与者更准确地识别出了错误分类的原因。

I总之，本研究做出了以下贡献： 贡献：

- **系统**。 我们设计并开发了一个交互式系统，用于理解和调试循环神经网络的内部行为。我们首先利用状态抽象方法，将语义相似的隐藏状态归并，从而将循环神经网络模型抽象为有限状态机。随后，我们设计并实现了三个紧密协同的视图：状态图视图、模式摘要视图和实例视图，从不同角度可视化并解释循环神经网络模型的内部行为。我们已在 GitHub 上开源该系统 [1]。

- **可视化与交互**。 我们提出了一系列可视化和交互设计，以便从不同粒度理解和调试循环神经网络。具体而言，我们结合了状态图、响应式工具提示、状态轨迹、颜色高亮、筛选、实例匹配和模式摘要，以同时展示模型的全局行为、实例级解释、关键模式和相似实例。

- **评估**。 一项被试间用户研究表明，DeepSeer 有助于机器学习开发者理解模型的整体行为并调试错误分类。

2 背景：循环神经网络

循环神经网络（RNN）是一种深度神经网络，专为处理序列输入（例如文本数据）而设计。本节将简要介绍其基础知识。

如图 2 所示，循环神经网络模型接收序列输入 $\{x_1、x_2、\ldots\ldots、x_T\}$。循环神经网络模型首先初始化其隐藏状态向量 $h_0 \in \mathbb{R}^N$，其中 N 是该隐藏状态向量的维度。在时间步 t，循环神经网络模型接收一个输入 x_t（$1 \le t \le T$），将其内部隐藏状态从上一个时间步 h_{t-1} 更新为新的隐藏状态 h_t。这一过程也可以理解为维护并更新循环神经网络模型的“隐藏记忆”。因此，要理解循环神经网络模型，解读不同隐藏状态 [43, 59]中维护的这类“记忆”至关重要。

为了在时间步 t进行预测，循环神经网络模型会将隐藏状态 h_t 转换为所需的输出格式 y_t。例如，要执行分类任务，通常会将隐藏状态 h_t 输入多层感知机（MLP）网络，将 h_t 映射到 $u_t \in \mathbb{R}^K$，其中 K 为类别数。随后通过“softmax”函数计算概率分布 p_t：

[1] https://github.com/momentum-lab-workspace/DeepSeer

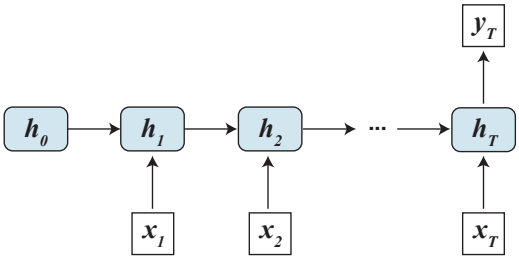


图2：基本循环神经网络（RNN）的工作流程。在每个时间步 t，RNN 接收输入 x_t 以更新其隐藏状态 h_t。通过处理隐藏状态 h_t，可以获得时间步 t 的预测结果（y_t）。

$$p_t = \text{softmax}(u_t)$$
$$p_t^i = \frac{e^{u_t^i}}{\sum_{j=1}^{K} e^{u_t^j}} \quad \text{for } i = 1, \ldots, K \tag{1}$$

通过找到能产生最大概率 p_t^k 的标签 k，进一步计算时间步 t 的预测结果。

请注意，更新隐藏状态 h_t 的过程可以通过不同类型的循环神经网络单元实现，例如 Elman 循环神经网络单元 [18],、长短期记忆网络（LSTM） [23], 和门控循环单元（GRU） [12]。在我们的用户研究环节中，我们使用 GRU；与其他变体相比，GRU 展现出更高的效率。请注意，我们提出的系统只需要访问隐藏状态，而不需要访问循环神经网络单元内部的更新过程。因此，该系统可以推广到不同类型的循环神经网络单元。

3 相关工作

3.1 可解释人工智能

我们的工作与可解释人工智能（XAI）最为相关，因为我们通过将深度神经网络（DNN）模型抽象为有限状态机（即全局解释），并呈现给定实例的状态轨迹（即局部解释），来提升模型可解释性。先前研究表明，支持模型可解释性能够提高用户对系统的接受度和信任度 [17, 22, 30, 55], 改善公平性 [14], 并提升人机团队绩效 [11]。良好的解释应处于可解释域中 [45], ，也就是说，应将抽象概念（例如数值向量）映射到人类能够理解的领域（例如图像、文本）。我们的工作受到这一原则的启发——不同于一些先前工作将隐藏状态值可视化的做法 [59], ，我们将隐藏状态映射回文本语料库中的语言模式。这样，用户便能轻松识别循环神经网络模型从训练数据中学到了哪些模式。

现有的可解释人工智能方法大致可分为两类：模型无关方法和模型感知方法。模型无关方法 [39, 53, 54] 将待解释的模型视为黑盒。LIME [53] 是这一类中广为人知的技术。给定一个输入实例，它会利用该实例附近的训练数据，学习一个更简单且可解释的模型（也称为代理模型），例如线性回归模型，以近似复杂模型。通过呈现代理模型中特征的重要性，LIME 为该实例的预测生成局部解释。然而，这些模型无关方法在生成解释时通常会忽略模型的内部行为。具体而言，对于循环神经网络模型，它们不会考虑不同隐藏状态之间的状态转移。一方面，这可能导致解释的保真度较低 [52]。另一方面，模型开发者等高级用户群体可能会认为，这不足以用于调试模型行为 [59]。为应对这一挑战，DeepSeer 旨在通过一种新颖的有限状态机抽象来探究循环神经网络模型的内部行为。

与模型无关方法不同，模型感知方法试图揭开深度神经网络的黑盒。在各种模型感知方法中，我们的工作与归因方法和影响函数方法最为相关。归因方法 [56, 58, 64]通常利用深度神经网络模型中的激活值或梯度信息，计算输入特征的重要性分数，例如图像中的像素、句子中的词元。举例来说，Karpathy 等人 [27]提出了一种可视化方法，将神经元的激活值映射到句子中的各个字符。这种可视化方法仅适用于单个句子（即局部解释），当句子数量较多时便难以解读。我们的工作与之不同：我们在有限状态图中汇总来自多个句子的、隐藏状态相似的词和短语，同时提供一种深入查看单个句子状态轨迹的方式。影响函数方法 [7, 31, 32] 根据梯度计算单个训练实例的影响，并找出一组对模型预测有重大影响的实例。我们的影响模式和可能的错误模式设计借鉴了这些方法。具体而言，DeepSeer 通过分析训练数据的隐藏状态，总结出通常会显著影响模型预测或可能导致漏洞的短文本模式。

我们还建议读者参阅现有的综述和文献回顾 [2, 6, 44]，以进一步了解不同的可解释人工智能方法。

3.2 DNNDebugging、测试与修复

已有多种可解释人工智能技术被用于理解和调试模型错误 [3, 29, 32, 53]。例如，Ribeiro 等人对 27 名参与者开展了一项用户研究，并表明 LIME 生成的解释可用于检测模型学到的虚假相关性。Koh 等人 [32] 证明，影响函数可用于调试领域不匹配。然而，Adebayo 等人 [3] 发现，事后模型解释，尤其是归因方法，有时无法有效检测某些类型的漏洞，例如标签错误和分布外错误。

与此同时，软件工程（SE）领域通过将传统软件工程技术改造应用于深度神经网络模型的调试、测试和修复，开发了多种技术 [40, 41, 62, 63]。DeepRepair [63] 使用基于风格迁移的数据增强方法来修复深度神经网络模型。RNNRepair [62] 识别用于重新训练的有影响力的实例，并在样本级和片段级修正两类错误预测。MODE [41] 提出了一种调试工作流：首先进行模型状态差分分析，然后选择训练实例进行重新训练。LAMP [40] 通过计算输入的重要性来提供数据溯源信息

并通过自动微分实现。这些技术侧重于自动化调试和重新训练流程，不让人类参与其中。我们的工作与这些技术有两点不同。第一，DeepSeer旨在通过将循环神经网络抽象为状态图并识别有影响力的模式，全面理解模型行为，而不局限于诊断模型预测错误。第二，为了帮助模型开发者诊断模型错误，DeepSeer会呈现输入的中间预测结果，并提供用于探查各个词和短语如何影响预测结果的功能，而不是自动定位模型错误的根本原因。

3.3 循环神经网络可视化

人们提出了许多深度神经网络可视化技术，以帮助用户理解和分析深度神经网络模型的内部运作机制。与我们的工作最相关的是专为循环神经网络设计的可视化技术 [27, 43, 59]。Karpathy 等人 [27] 根据激活函数的输出，将循环神经网络在输入句子中关注的字符可视化。Li 等人 [33] 使用基于梯度的显著性分数，而非神经元激活，来衡量输入句子中每个词的重要性，然后将显著性分数以热图形式呈现。这两种可视化都是静态的，每次只能展示单个输入的隐藏状态。Strobelt 等人 [59] 在此基础上扩展，构建了一种名为 LSTMVis 的交互式可视化方法。LSTMVis 在平行坐标图中呈现单个隐藏状态。用户可以交互式地选择输入句子的特定片段，并搜索隐藏状态相似的其他输入。然而，鉴于循环神经网络通常具有数百甚至数千个隐藏状态，逐一可视化隐藏状态会给用户带来显著的认知负担。为解决这一问题，DeepSeer 将相似的隐藏状态聚类为抽象状态，并将循环神经网络模型表示为有限状态机。这大幅减少了用户需要跟踪的状态数量，也使 DeepSeer 能够直接可视化有限状态机，从而呈现整个模型的全局视图，而非单个隐藏状态。我们的工作也与 RNNVis [43] 相关。RNNVis 将相似的隐藏状态聚类为记忆芯片，并以词云形式呈现与每个聚类相关的文本输入。然而，与有限状态机不同，这种设计无法捕捉隐藏状态之间的状态转移，也无法呈现常见词之外更长的语言模式。此外，DeepSeer 还提供了其他功能，以便于检查和调试模型，例如呈现中间预测结果、总结有影响力的模式和错误模式等。

4 设计目标与系统概述

本节首先基于文献综述总结我们系统的设计目标。随后，我们介绍系统概述，讨论系统设计如何支持每项设计目标。

4.1 用户需求与设计目标

为了解循环神经网络开发者的需求，我们对以往文献进行文献综述，这些论文曾针对解释深度学习模型开展形成性研究、对现有工具进行用户研究，或讨论了解释和调试深度学习模型所面临的挑战与机遇。基于文献综述，我们总结出 DeepSeer 的以下设计目标：

G1. 帮助用户理解循环神经网络模型的整体行为。以往研究表明，模型开发者希望从较高层次理解模型学到了什么 [13, 35, 43]。例如，Kaur 等人调查了 197 名机器学习开发者对可解释性工具功能的看法，其中 61% 的回答提到了全局解释的重要性 [28]。具体而言，Ming 等人强调了呈现循环神经网络模型隐藏状态所捕获的语义信息的重要性 [43]。因此，DeepSeer 应帮助模型开发者理解循环神经网络模型的整体行为，尤其是其隐藏状态所学习到的语义信息。

G2. 帮助用户理解模型针对特定感兴趣输入的决策过程。在检查单个预测结果，尤其是错误的预测结果时，模型开发者希望理解模型为何会对该特定输入作出这样的预测 [5, 24, 35]。例如，Hohman 等人通过对九名机器学习开发者开展形成性研究 [24]，发现用户希望了解不同特征如何影响模型的决策。此外，Kahng 等人访谈了十五名 Facebook 开发者，发现对他们而言，理解复杂模型的一种自然方式是检查模型在单个示例上的行为 [26]。大规模调查也证实了局部解释的重要性 [28]——65% 的受访者认为局部解释十分重要。

G3. 帮助并协助用户搜索相似数据。通过与两位自然语言处理开发者开展以用户为中心的设计流程，Liu 等人发现，自然语言处理开发者通常采用一种“探索优先”的方法来发现并调试自然语言处理模型中的错误 [38]。也就是说，开发者倾向于检查并比较相似输入样例的预测结果，以获得洞见。因此，还应提供可追溯性，帮助用户在调试模型预测时轻松探索相似样例 [24]。具体而言，Strobelt 等人强调，为循环神经网络匹配相似样例有助于开发者验证解释假设 [59]。

G4. 帮助用户总结输入数据的共同特征。检查单个数据点可能既繁琐又耗时，从而妨碍洞见的发现。Kahng 等人发现，Facebook 的模型开发者常会整理具有共同特征的数据子集，以了解模型在高层次分类中的行为 [26]。此外，帮助用户识别输入数据的共同特征，对于错误分析尤为有用。Jin 等人发现，机器学习开发者通常需要检查一组错误预测所共有的特征，并验证由这些特征形成的错误模式是否合理 [25]。然而，用户通常需要依据自己的领域知识手动完成这项工作。因此，DeepSeer 应支持用户识别并检查输入数据的共同特征，尤其是预测错误的输入数据。

4.2 系统概述

为了帮助用户从整体上理解循环神经网络模型学到了什么（G1），我们选择将循环神经网络模型呈现为状态图，其中每个节点是一组相似的

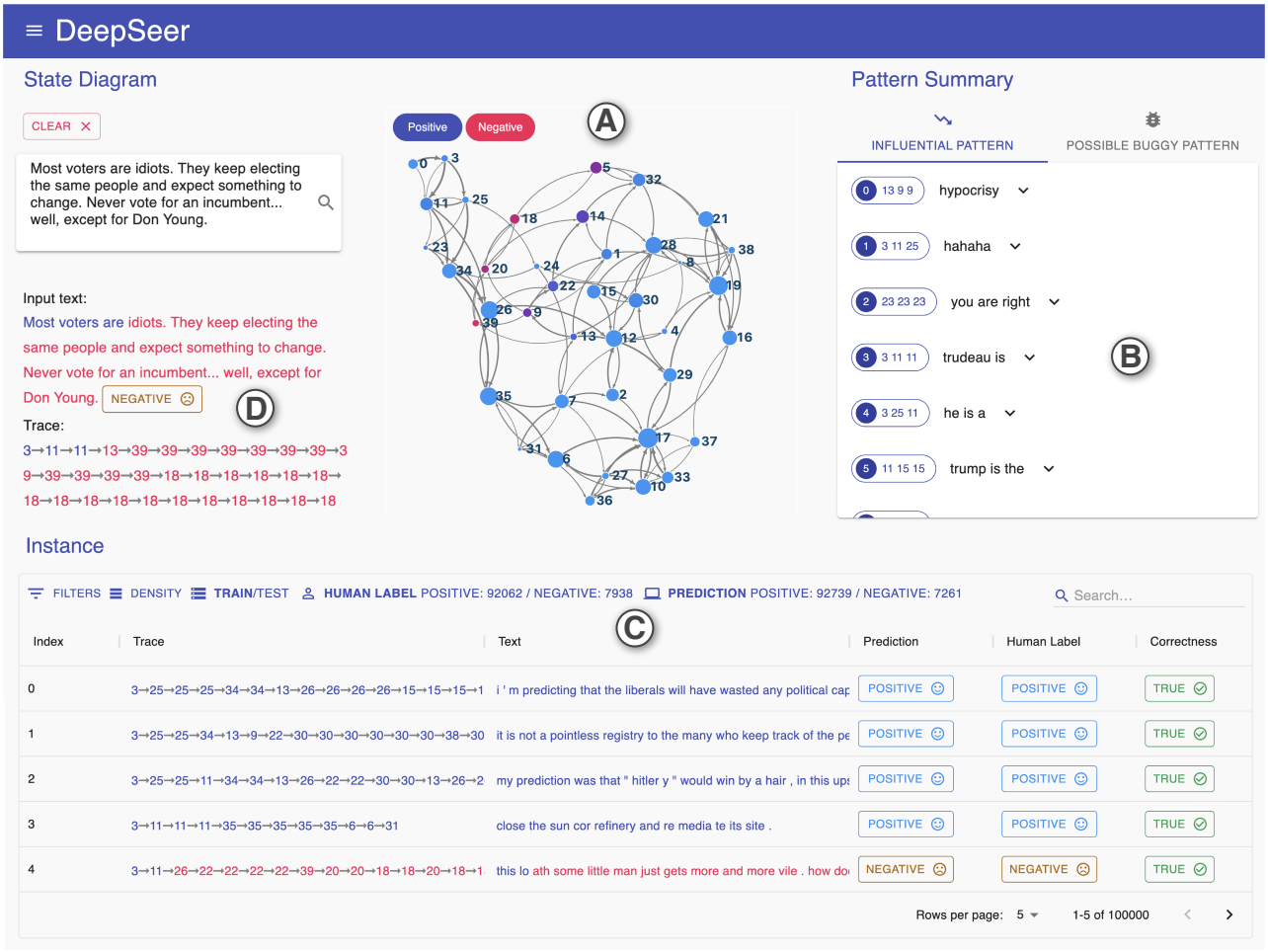


图 3：DeepSeer 是一个用于可视化、理解和调试循环神经网络模型的交互式系统。(A) 状态图视图 **展示循环神经网络模型的抽象状态和状态转移。(B)** 模式摘要视图 **展示循环神经网络模型学到的常见文本模式。(C)** 实例视图 **以交互式数据网格的形式展示原始数据，供用户探索用于训练或测试循环神经网络模型的数据。(D) 用户输入新句子时，会呈现**中间预测结果 。

循环神经网络模型的隐藏状态，如图 3 A◯ 所示。与包含数百或数千个隐藏状态的原始循环神经网络模型相比，经过状态聚类后，状态图规模小得多，因此更便于浏览。此外，DeepSeer 将每个节点与相应隐藏状态记忆的文本模式关联起来，帮助用户理解隐藏状态的语义。另一种设计是直接将隐藏状态的值可视化 [27, 59],；这些值是高维数组，难以解读。相比之下，状态图更便于浏览和检查。

为了帮助用户理解模型在特定输入上的模型决策过程（G2），DeepSeer 将循环神经网络模型读取输入句子中的每个词后所产生的中间模型预测结果可视化（图 3 D◯）。这样，用户就能轻松看出哪个词影响了模型的决策，并对最终结果贡献更大。为支持 G3，DeepSeer 允许用户在实例视图中搜索具有相似文本模式的输入句子（即包含相同关键词或遵循相同正则表达式的句子），或具有相似模型行为模式 的句子（即具有相同状态或遵循相同状态轨迹的句子）（图 3 C◯）。为了帮助用户发现常见模式（G4），DeepSeer 会主动识别对模型预测结果影响较大的高频文本模式，以及错误预测中共有的模式（图 3 B◯）。这些常见模式也可作为全局解释方法的补充（G1）；如果用户觉得状态图令人眼花缭乱，它们还能为调查提供更直接的切入点。

5 设计与实现

5.1 状态抽象

为了从循环神经网络模型生成状态图，我们开发了一种方法，将循环神经网络模型中语义相关的隐藏状态聚类为一个抽象状态。我们的工作受到对有状态循环神经网络进行基于模型分析的研究启发 [15, 16, 47, 51, 61]。这些研究采用各种技术，从有状态循环神经网络中提取可解释的状态转移

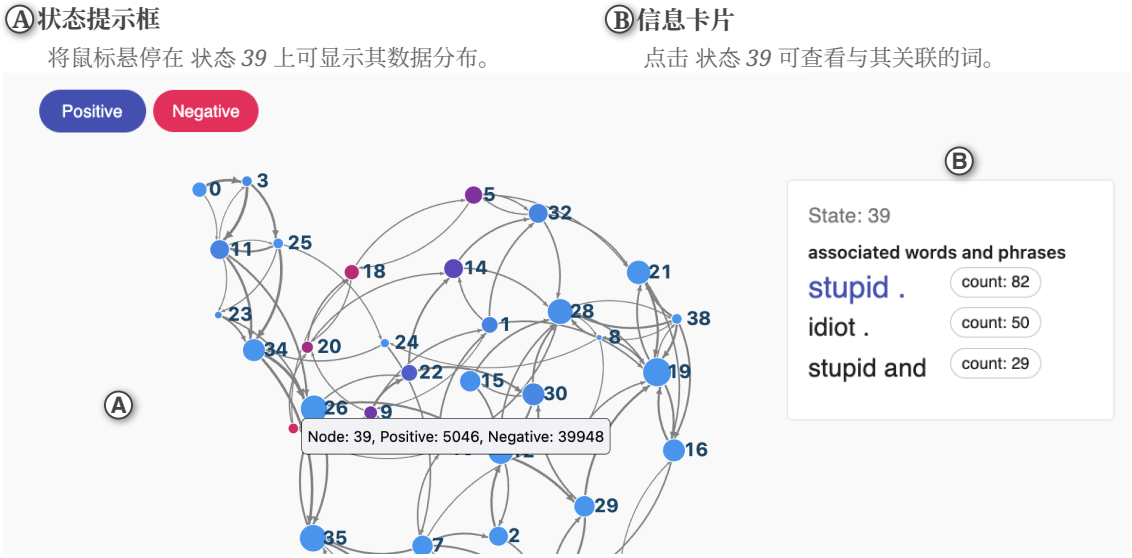


图 4：通过与状态图视图 交互，了解循环神经网络模型的抽象状态。

模型（例如，离散时间马尔可夫链、自动机）。通过将复杂的循环神经网络转换为可解释的状态转移模型，黑盒便成为更加透明的模型，从而提升模型可解释性，也为进一步分析提供了可能。我们选择基于一种最先进的方法 DeepStellar [16], 开展工作，因为已有研究证明它在多种任务中有效，包括对抗检测 [16], 、深度神经网络测试 [16], 和深度神经网络修复 [62]。先前研究还表明，在 97% 的测试数据上，抽象状态能够与原始循环神经网络模型做出相同的预测 [62]。

为了获得训练后循环神经网络的有限状态机表示，我们对状态和状态转移都进行抽象。附录 A 给出了状态抽象算法。下面简要介绍其工作原理。对于训练数据中的每个实例，我们的方法首先在推理过程中记录中间隐藏向量 $\{h_1, h_2, \ldots\ldots, h_l\}$，其中 $h_i (1 \leq i \leq l)$ 是循环神经网络模型的一个具体隐藏状态。l 表示句子中的词元数量。然后，我们的方法对从训练数据中收集的所有具体状态应用主成分分析（PCA）进行降维，再进行抽象。与使用基于区间的方法进行状态抽象的 DeepStellar [16], 不同，我们的方法使用高斯混合模型（GMM） [42] 对相似的具体状态进行聚类。GMM 解决了基于区间的方法的两个关键局限：1）新生成的隐藏向量在测试时可能落在区间之外，导致出现未知状态；2）状态数量会随着 k 维度和 m 区间（m^k）呈指数增长，过多的状态也难以可视化。通过状态抽象，给定输入上的模型预测过程可以表示为一系列抽象状态。我们将这一状态序列称为模型预测的轨迹 。

我们对状态抽象的忠实性进行了定量分析。我们测量了使用场景（第 6 节）和用户研究（第 7.2 节）中的三种不同自然语言处理任务里，抽象模型与原始模型之间的预测一致性。测试数据上的预测一致性分别为 99%、97% 和 85%。这表明，抽象模型能够忠实地表示循环神经网络的行为。实验详情见附录 B。

不同于以往专注于状态抽象技术本身，或使用该技术进行模型测试与修复的工作，我们的工作首次将该技术扩展到交互式模型解释与调试，并提供了更易于使用的用户界面。我们的工作首次将状态抽象融入“人在回路”方法，以丰富的交互机制支持用户理解和调试循环神经网络模型。在以下小节中，我们将介绍基于状态抽象构建的 DeepSeer 交互式功能。

5.2 状态图视图

该状态图视图将上一步从给定循环神经网络中抽象得到的有限状态机可视化。它概览了模型行为，并帮助用户理解其隐藏状态的语义含义。用户可以浏览不同的状态节点，以探索某个状态通常会导向何种预测结果，以及该状态被访问过多少次。具体而言，每个状态都会根据经过该状态的输入实例产生特定预测结果的频率进行颜色编码。在训练期间访问过某个节点的输入句子数量决定了该状态节点的大小。例如，在图 4 Ⓐ中，由于循环神经网络模型可能做出两种预测——正面评论或负面评论，因此所有节点被赋予两种不同的颜色——蓝色表示正面评论，红色表示负面评论。由于红色节点较少，且其大小明显小于蓝色节点，因此可以推断训练数据集中正面评论多于负面评论，并且循环神经网络模型更有可能做出正面预测。两个状态之间边的宽度表示训练期间该状态转移发生的频率。

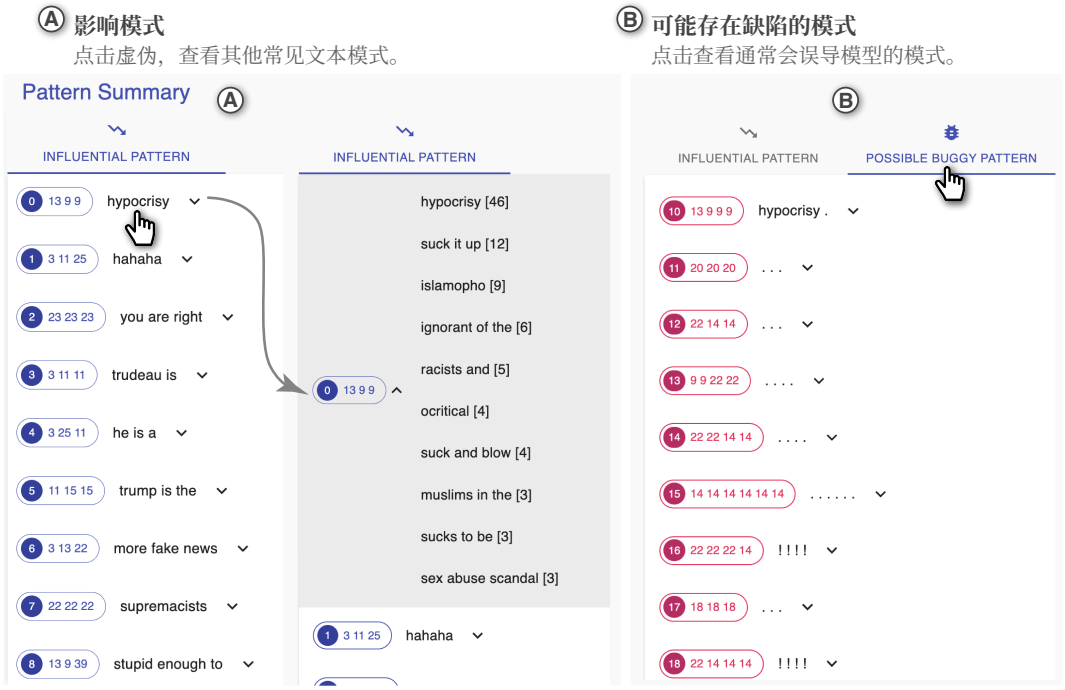


图 5：与 t 交互 he模式摘要视图 在 DeepSeer 中检查循环神经网络模型所学习的文本模式 .

循环神经网络模型在从给定输入句子中读取更多词时，会从一个状态转移到另一个状态。

当用户将鼠标悬停在某个状态上时，系统会显示工具提示，提供有关该状态的更多详细信息，例如，经过该状态并最终被预测为特定结果的训练实例数量（图 4 A◯）。当用户点击某个状态时，会弹出信息卡片（图 4 B◯），显示训练数据中经常与该状态相关联的短语和词语。此功能使用户能够解读隐藏状态所记忆的语义信息。点击某个状态还会更新实例视图，筛除模型预测期间未经过该状态的输入实例。

5.3 模式摘要视图

模式摘要视图 根据训练期间频繁出现的状态转移生成常见模式。具体而言，频繁出现的状态子序列被视为一种模式，并可根据与每个状态相关联的词语或短语，进一步转换为词语序列。DeepSeer 可识别两类模式：有影响力的模式 和可能的错误模式。

影响模式是指会改变模型中间预测结果的模式，例如，在读取给定输入句子中间的某些词后，预测从正面评论变为负面评论。为识别从训练数据中学习到的有影响力的模式，DeepSeer 首先识别每个训练实例的状态轨迹中的转折点（即中间预测发生变化的位置）。然后，DeepSeer 将每个状态轨迹根据转折点划分为子序列。这些子序列按其出现频率排序，并在模式摘要视图中按降序呈现。

可能的错误模式 仅从训练数据中预测错误的数据实例中挖掘。这些模式表明循环神经网络模型未能很好地学习某些情况，因此产生了错误预测。为了识别错误模式，DeepSeer 首先根据预测结果的正确性将训练数据 S 划分为两个子集。我们分别将仅包含正确预测的子集记为 S_c，将仅包含错误预测的子集记为S_f。然后，我们使用 TKS [19]（Top-K序列模式挖掘）从每个子集中挖掘频繁子序列模式。如果某个子序列模式仅出现在错误分类的输入（S_f）中，而未出现在正确分类的输入（S_c）中，则认为该模式可能是错误模式。这些错误模式会根据其频率排序，并按降序呈现在模式摘要视图中。

用户可以点击某个特定模式，查看与该模式相关的高频短语（图 5 A◯）。模式摘要视图使用户能够了解模型学习了哪些模式，以及这些模式将如何影响模型的预测。此外，可能的错误模式 使用户能够识别循环神经网络模型潜在的预测风险。点击某个模式后，实例视图将更新，以筛除不符合所选模式的数据实例。

5.4 实例视图

其中，实例视图（图 6）是一个可滚动的数据网格，其中包含用于训练或测试模型的原始数据。数据网格的行是单个数据实例，列包括：索引、状态轨迹、



图 6：交互 在 DeepSeer 中使用实例视图 探索用于训练/测试循环神经网络模型的数据 1.

文本、预测、人工标签和数据实例的正确性。用户可以按每一列对数据实例进行排序和筛选（图 6 A◯）。用户可以使用TRAIN/TEST 选项卡在训练数据实例和测试数据实例之间切换。人工标签和模型预测结果的分布会在此视图顶部进行汇总和呈现（图 6 B◯）。用户筛选数据实例时，这些分布也会相应更新。用户还可以根据关键词或正则表达式搜索特定的输入数据（图 6 C◯）。匹配的结果将被突出显示，以便更好地进行可视化。对于每个实例，其词和状态会根据中间预测结果进行着色。在实例视图中点击某一行，将更新状态图视图，以呈现所选数据实例的状态转移。

5.5 中间预测结果

前文介绍了用户如何利用不同视图，全面了解模型行为。先前研究表明，允许用户检查并调试单个实例上的模型预测也很重要 [39, 53, 56]。为支持实例级检查和调试，DeepSeer 允许用户在图 7 的文本框中输入一个输入句子。点击放大镜按钮后，DeepSeer 会呈现模型针对给定输入的预测状态轨迹。此外，输入句子中的每个词都会根据中间预测结果着色。例如，在图 7 B◯ 中，“red” 和 “blue” 分别表示负面评论和正面评论。循环神经网络通常会使用读取整个句子后的最终隐藏状态来计算类别概率。在DeepSeer中，句子中每个词读取完毕后的隐藏状态都会输入输出层，以生成中间预测。借助这些中间预测，用户可以检查循环神经网络模型在读取输入句子中更多词语的过程中，预测结果如何随时间变化。模式摘要视图也会更新，仅显示有影响力的模式以及与给定输入句子相关的可能错误模式。借助这些支持，用户可以在调试错误的模型预测时，快速找到可疑的词或短语。

6 使用场景

假设 Alice 是一名模型开发者，她使用 Toxic 数据集 [2]训练了一个用于检测有害内容的循环神经网络模型。该模型会预测一个句子是正面语气还是负面语气。她的模型在训练数据上的准确率达到 95%，但在测试数据上只有 89%。Alice 使用 DeepSeer 来探究为何会出现这种性能漂移。

6.1 循环神经网络模型的可视化与理解

Alice 首先查看状态图视图，借此概览训练好的循环神经网络模型将其视为有限状态机后的情况（图 3 A◯）。在这个状态图中，每个节点代表循环神经网络模型中一组相似的隐藏状态。蓝色节点表示模型访问该状态后[redacted]可能给出 中间预测；而红色节点表示模型更有可能给出 中间预测。Alice 将鼠标悬停在一个名为状[redacted]红色节点上。如图 4 所示，随后弹出的工具提示显示，模型访问该状态后，给出负面中间预测的次数为39，948 次，而给出正面中间预测的次数仅为5，046 次。Alice 点击状态 39 的节点后，右侧会显示一张信息卡片（图 4 A◯），其中列出了与该状态相关的常见词语和短语，例如“愚蠢”、“白痴”和“愚蠢而”。Alice 浏览下方实例视图中的几句话（图 3 C◯），看看这些词语是否出现在仇恨评论中。通过这种方式，

[2]https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification

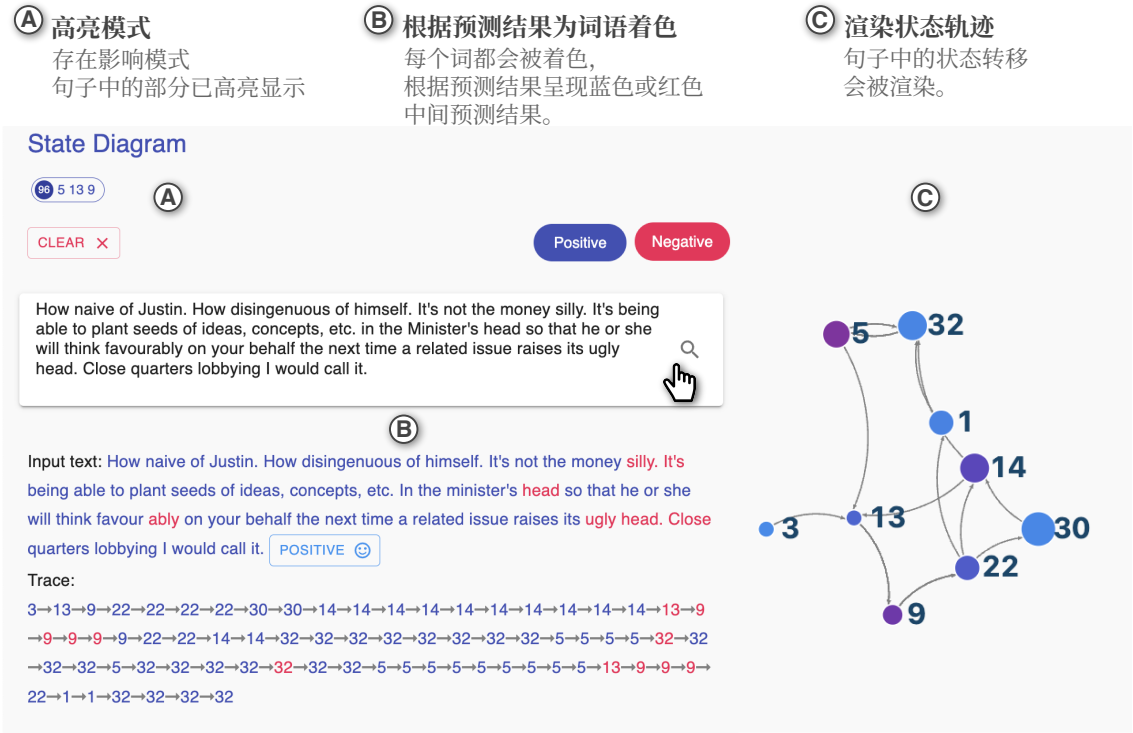


图 7： 中间预测结果**帮助用户在调试时理解模型的预测**。

Alice 确认，她的循环神经网络模型确实从训练数据中学到了一些有意义的模式。

虽然检查与各个状态相关联的文本模式很有帮助，但 Alice 发现逐一检查状态图中的所有状态十分繁琐。因此，她切换到模式摘要视图（图 3 B◯），从另一个角度理解模型。该视图展示了对模型预测有显著影响的文本模式（即有影响力的模式）。Alice 发现了一些有趣的模式，例如“more fake news”和“stupid enough to”。当 Alice 点击一个模式“hypocrisy”（图 5 A◯）时，该模式会展开，显示与相同状态序列 ⑬ → ⑨ → ⑨ 相关联的其他高频模式列表，并按频率排序。[3] 例如，在训练数据中，这个状态序列还记住了“suck it up”（12 个句子）、“ignorant of”（6 个句子）和“suck and blow”（4 个句子）。Alice 点击每个模式时，实例视图也会更新，以筛选出符合所点击模式的训练数据和测试数据。Alice 也很好奇哪些文本模式可能会导致模型预测错误。因此，她切换到可能存在缺陷模式的列表。这些有缺陷模式仅根据误分类句子进行汇总，而非依据整个训练数据集。Alice 看到了一些模式，例如“...”、“!!!”和“???”（图 5 B◯）。看来，她的模型学到了一些标点符号与预测结果之间的虚假相关性，这可能导致了许多错误。为防止模型学习这些虚假相关性，Alice 计划移除这些标点符号，以清理训练数据，从而有望提升模型性能。

6.2 调试 RNNModel

在对模型行为有了较高层次的理解后，Alice 想深入分析数据，探究为什么有些句子会被误分类。因此，她打开实例视图（图 3 C◯），其中以分页表格展示了所有训练数据和测试数据。

Alice 首先注意到训练数据并不均衡。正面句子（92062）的数量明显多于负面句子（7938）。随后，Alice 切换到测试数据，并使用正确性列上的筛选功能找出误分类句子（图 6 A◯）。她将一个误分类句子复制到文本框中，并对其运行实例级模型解释功能（图 7）。句子中的每个词都根据中间预测结果着色，下方还显示了状态轨迹。

Alice 很快注意到，在预测过程中，她的循环神经网络模型将几个词视为负面词语，例如“丑陋的脑袋”。尽管模型识别出了这些侮辱性词语，最终仍将这个句子预测为正面。看来，模型在看到句子中的后续词语后，很快就忘记了这些侮辱性词语。例如，看到“quarters”后，中间预测便从负面变成了正面。

为了验证这一假设，Alice 使用 实例视图中的关键词搜索功能，在训练集中搜索包含“quarters”的句子。Alice 在训练集中找到了 27 个正面句子，而只找到 1 个包含“quarters”的负面句子（图 8）。由于

[3] 单词“hypocrisy”与一连串三个状态相关联，因为它在训练集中被分词为三个词元（hypo-、-cri-、-sy），每个词元都对应一个状态。这种分词机制广泛应用于自然语言处理中，以解决词汇表外问题。



图 8：Alice 在训练数据中搜索“quarters”。

许多包含“quarters”的句子都是正面的，因此模型可能学到了“quarters”与正面情感之间的虚假相关性。Alice 通过查看与词“quarters”相关联的对应状态，进一步证实了这一假设：状态22。Alice 发现，状态22 是一个正面状态。因此，Alice 得出结论，这次错误分类的原因之一是数据不平衡：大多数包含“quarters”的句子都被标记为正面。为了解决这一问题，Alice 认为一种可能的方案是收集更多包含该关键词的数据，以平衡训练集，然后重新训练模型。此外，鉴于模型很快就会忘记一个侮辱性词语，Alice 还计划尝试采用带有注意力机制的长短期记忆（LSTM）架构。与原始RNN相比，这种架构能够让信息在记忆中保留更长时间。

7 用户研究

我们开展了一项组间研究，共有 28 名参与者，以评估 DeepSeer 的有效性和可用性。我们使用 LIME [53], 这一用于解释和调试机器学习模型的知名工具作为比较基线。尽管 LIME 并非专门针对循环神经网络，但它是用于理解和调试模型的广泛使用工具。它在 GitHub 4 上拥有 10.3K 个 stars 和 1.7K 个 forks，其 Python 软件包在 PyPI [5]上的下载次数已达 1,600 万次。因此，我们选择它作为更现实的基线。给定一个模型预测，LIME 可以为输入数据中的元素（例如输入句子中的词）生成带有重要性分数的解释。为了实现公平比较，我们构建了一个与 DeepSeer 类似的 LIME 界面。该界面包含 LIME 提供的现有可视化功能，也包含 DeepSeer 中的实例视图。它不包含状态图视图和模式摘要视图，因为这两者是 DeepSeer 的创新贡献。为了评估 DeepSeer 相较于 LIME 的整体实用性，我们考察了以下研究问题：

- RQ1：与常用的模型解释和调试工具相比，DeepSeer 在多大程度上能够提升用户对循环神经网络模型的理解?
- RQ2：与常用的模型解释和调试工具相比，DeepSeer 在多大程度上能够提高用户识别循环神经网络模型误预测根本原因的准确率?

7.1 参与者

我们通过阿尔伯塔大学计算机科学系和电子与计算机工程系的多个研究生邮件列表招募了 28 名参与者（5 名女性和 23 名男性）。[6] 所有参与者都具备至少基础的机器学习背景。15 名参与者是博士生，其余为硕士生。23 名参与者拥有 2 至 5 年的机器学习经验，3 名参与者拥有 5 年以上经验，2 名参与者拥有约 1 年经验。关于循环神经网络的经验，9 名参与者拥有 2 年以上经验，7 名参与者拥有 1 年经验，12 名参与者拥有不到 1 年经验。参与者还通过一道 7 点李克特量表题目自我报告了他们对开发循环神经网络模型的熟悉程度。中位数为 5，其中 1 表示“我只听说过循环神经网络，但从未使用过”，7 表示“我能够独立构建循环神经网络模型”。25 名参与者表示，他们没有使用过任何深度学习调试工具；3 名参与者表示，他们使用过 Tensorboard [1]。研究通过 Zoom 开展。DeepSeer 和 LIME 均部署为网络应用，参与者可以通过个人电脑访问。

7.2 循环神经网络模型

由于 DeepSeer 专为可视化和调试循环神经网络模型而设计，我们针对两项热门机器学习任务训练了两个循环神经网络模型。每个循环神经网络模型的隐藏状态向量维度均为 256。第一项机器学习任务是预测 Quora 上的问题是否真诚。该任务最初来自 Kaggle（一个热门的在线机器学习与数据科学社区）举办的一项精选竞赛 [50]。在此任务中，我们使用 100,000 个 Quora 问题训练循环神经网络模型，每个问题均标注为真诚或不真诚。该循环神经网络模型的训练准确率为 93.93%，测试准确率为 89.07%。第二项机器学习任务是根据名为 AG's News 的新闻语料库预测新闻文章的主题 [20]。该任务是主题分类研究中一个广为人知的基准 [65]。在此任务中，我们使用 109,886 篇新闻文章训练循环神经网络模型，这些文章被分为四个新闻主题类别，包括“体育”“商业”“世界”和“科学与技术”。该循环神经网络模型的训练准确率为 91.57%，测试准确率为 87.68%。DeepSeer 将每个循环神经网络模型抽象为 40 个状态。此数量是通过经验确定的，以便在准确率与检查状态图所需的认知努力之间取得良好平衡。我们还在附录 C 中对抽象模型进行了忠实度分析。在用户研究过程中，我们随机将两个循环神经网络模型之一分配给参与者，以完成模型理解和调试任务。每项任务对应的 DeepSeer 界面见附录 D。

7.3 协议

我们设计了一项被试间用户研究，在每次研究会话中，用户体验一种条件和一个循环神经网络模型。我们选择被试间设计，而不是被试内设计，是因为体验一种条件大约需要 60 分钟。在被试内设计中体验两种条件需要 120 分钟，时间过长，可能会导致明显的疲劳和挫败感。在每次会话开始时，我们征得参与者同意进行录音。由于这是一项被试间研究，每位参与者只会被分配到一种条件下的一个循环神经网络模型。在每次会话中，参与者只能使用指定的工具：实验条件下使用 DeepSeer，对照条件下使用 LIME [53]。分配的循环神经网络模型和条件在参与者之间进行了平衡。每次研究会话开始时，参与者首先观看所分配工具的 5 分钟教程视频，然后

[4]https://github.com/marcotcr/lime [5]https://pepy.tech/project/lime [6]本研究涉及人体参与，已获大学研究伦理办公室批准。

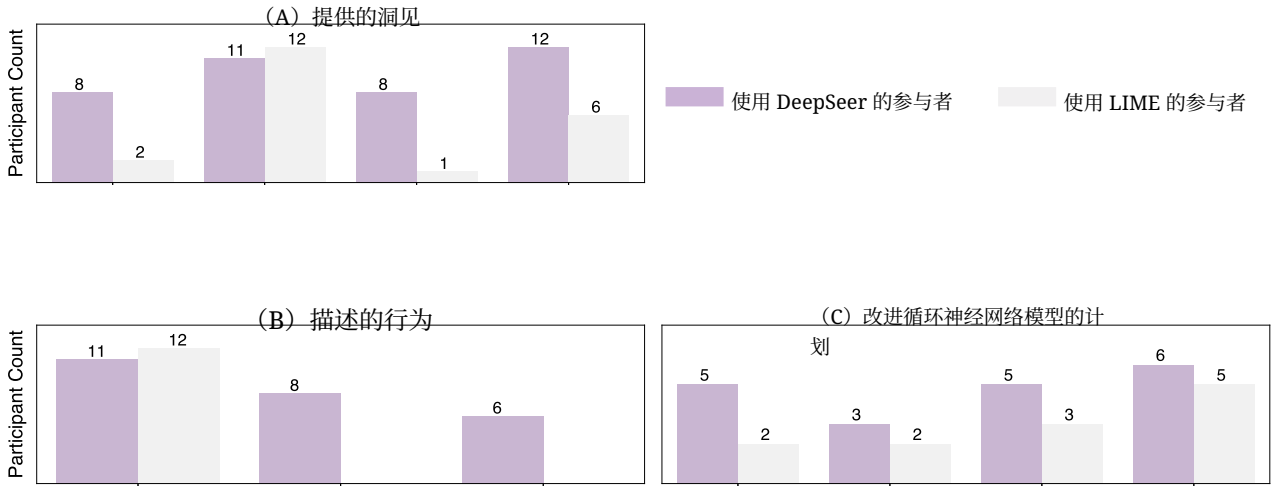


图 9：参与者对模型理解的表现。(A) 提及模型行为不同方面的参与者人数。(B) 以不同方式描述模型行为的参与者人数。(C) 对模型改进提出不同建议的参与者人数。

花 5 分钟熟悉该工具。随后，参与者有 30 分钟使用所分配的工具探索指定的循环神经网络模型，并通过问卷分享他们对模型行为的理解。问卷包括三个主要问题：(1) 你对模型的表现和行为有哪些认识？(2) 你是否发现了循环神经网络模型的任何漏洞或局限性？如果有，你发现了哪些类型的漏洞？(3)你将如何进一步改进该模型？完成模型理解问卷后，参与者有 10 分钟使用所分配的工具调试 5 个错误的模型预测。对于每个错误的预测，他们需要写下输入数据被错误分类的原因，并通过问卷提交答案。在研究会话结束时，参与者需要填写一份调查问卷，分享他们的体验。具体而言，研究后调查包括 NASA任务负荷指数（TLX）问题 [21]，用于测量研究过程中的认知负荷。每位参与者都会获得一张价值 25 美元的亚马逊礼品卡，作为其时间的补偿。

8 用户研究结果

本节介绍被试间用户研究的结果。我们首先展示并分析使用 DeepSeer 和 LIME 的参与者在模型理解和调试任务上的表现差异。随后介绍参与者对 DeepSeer 工具特性及认知负荷的看法。为简洁起见，我们用 P1-P14 表示使用 DeepSeer 的参与者，用 P15-P28 表示使用 LIME 的参与者 [53]。

8.1 RQ1：用户对模型理解的表现

为评估用户对模型的理解表现，两位作者手动评估并编码参与者的回答，并统计参与者分享的关于模型行为的正确洞察数量。具体而言，两位作者召开了 4 次会议，以制定编码手册并解决标注不一致的问题。最终生成了 651 个编码，并将其归入 32 个主题。最终的 Cohen's Kappa 系数为 0.9061。需要注意的是，只有在两位作者均同意的情况下，一条洞察才会被视为正确。

总体而言，使用 DeepSeer 的参与者提供了更多见解（53 条，而使用 LIME 的参与者为 21 条）。每位参与者提供的见解数量平均相差 2.3 条，且这一差异具有统计显著性（Welch t 检验：$p = 0.0003$）。图 9 展示了参与者分享的不同类型见解的细分情况。使用 DeepSeer 的参与者分享了更多关于全局模型行为、模型性能和错误行为的见解。例如，P12 说："模型似乎对输入句子后半部分赋予了很高的权重。"P9 写道："这个模型经常混淆商业和科学类别。"此外，使用 DeepSeer 的参与者在描述模型行为时经常提及文本模式和状态，而使用 LIME 的参与者大多提及具体关键词。P25 说："当数据集规模很大且类别很多时，用 *LIME [很难总结模式]*。"由于 LIME 专为局部解释而设计，因此使用 LIME 的参与者中只有 2 人能够为指定的循环神经网络模型推导出全局解释，这并不令人意外。

与使用 LIME 的参与者相比，使用 DeepSeer 的参与者还提出了更多有用且多样的模型改进建议（图 9 (C)）。请注意，只有当建议与观察到的模型误预测的根本原因相关，并且被机器学习社区认可为有效的模型改进机制时，才会被视为有用。具体而言，5 位使用 DeepSeer 的参与者注意到，读取更多词元后会出现忘记先前词元的错误模式，并建议添加注意力层；而使用 LIME 的参与者中只有 2 位注意到了这一点。P10 写道："对于许多

表1：参与者在模型调试任务中分享的回答平均值。

	Quora		AGNews		总体	
	DeepSeer	LIME	DeepSeer	LIME	DeepSeer	LIME
每位参与者提供的合理解释	4.3	2.7	4.3	1.1	4.3	1.9
每位参与者提及的诱发故障关键词	5.3	2.7	4.6	1.4	4.9	2.1
每位参与者提及的非诱发故障关键词	0.8	3.7	1.0	3.3	0.9	3.5

错误预测而言，模型很可能一开始给出正确预测，但随后就转向错误方向。也许我们可以借助注意力机制之类的方法，缩短时间依赖的长度。" 最后，使用 DeepSeer 的参与者平均耗时 27 分 53 秒（σ = 2 分 44 秒），而使用 LIME 的参与者平均耗时 28 分 19 秒（σ = 3 分 15 秒）。我们未观察到任务完成时间存在显著差异。

8.2 RQ2：模型调试中的用户表现

为了评估用户在模型调试任务中的表现，我们统计了参与者针对五个误分类句子给出的合理解释数量。为了评估参与者答案的正确性，两位作者首先手动检查了循环神经网络的隐藏状态和训练数据，以诊断这五个错误分类。他们的调查结果被用作真实误分类解释。随后，他们检查参与者的解释是否与真实情况一致。

如表1所示，使用 DeepSeer 的参与者针对错误分类给出了更多合理解释。平均而言，使用 DeepSeer 的参与者为5个误分类句子给出了4.3个合理解释，而使用 LIME 的参与者仅给出了1.9个合理解释。2.4的均值差异具有统计显著性（Welch t 检验，$p < 0.0001$）。

此外，我们统计了参与者提及的正确诱发故障关键词数量。使用 DeepSeer 的参与者识别出的正确诱发故障关键词多于使用 LIME 的参与者（均值：4.9 对 2.1）。2.8的均值差异具有统计显著性（Welch t 检验，$p < 0.0001$）。此外，使用 LIME 的参与者更常将关键词误认为诱发故障关键词（均值：3.5 对 0.9）（Welch t 检验，$p < 0.0001$）。这是因为 LIME 会先训练一个代理稀疏线性模型来模拟循环神经网络模型，然后根据该线性模型计算词语重要性。这有时会导致解释不可靠。一些参与者在研究过程中也注意到了这一点。P22 评论道："在某些情况下，我发现这个工具 *[LIME]*生成的解释并不可靠。"

一个有趣的观察是，使用 DeepSeer 的参与者能够识别出超越词语模式的、更复杂的错误模式。例如，P9 回答道："一开始，*'football'*表示模型要预测体育，这正是模型所做的事情。但当*'UK'*出现时，状态就转变为*'world'[*相关状态*]*并停滞在那里。"LIME 用户没有提供此类见解，因为 LIME 会分别处理单个词语，无法捕捉模型决策过程的动态变化。

最后，使用 DeepSeer 的参与者平均用时 9 分 9 秒（σ = 1 分 21 秒）完成了这项任务，而使用 LIME 的参与者平均用时 9 分 25 秒（σ = 0 分 11 秒）。任务完成时间没有显著差异。

8.3 用户感知与认知负荷

我们的研究后调查征集了参与者对 DeepSeer 所有关键特征的反馈。总体而言，参与者认为 DeepSeer 的视觉编码和界面直观、实用且清晰。在 14 名参与者中，13 人自述未来开发和调试循环神经网络模型时愿意使用 DeepSeer，另有 1 人持中立态度。在 7 点李克特量表上，中位数为 6.5（1——完全不想使用，7——如果有的话一定会使用）。以下将结合研究后调查和用户研究录音，报告参与者对 DeepSeer 各项关键特征的定性反馈。

中间预测结果。 使用 DeepSeer 的全部 14 名参与者都认为 DeepSeer 按需提供的中间预测结果很有用。中位数评分为 7 分（满分 7 分）。P9 提到："逐步查看中间预测结果有助于我理解模型为什么会做出错误的预测。例如，这段文本显然与体育有关。然而，模型进入了状态 *20*，而该状态与体育并没有太大关系。" 此外，参与者也喜欢对中间预测结果进行颜色编码。

状态图。 在 14 名参与者中，有 11 名认为 DeepSeer 中的状态图很有用。中位数评分为 6 分。P12 评论道："将循环神经网络模型提取为状态图很不错，我认为在解读 *[*循环神经网络模型*]* 以及更复杂的数据（例如医疗数据）时，这也会很有帮助。"尽管大多数参与者并不觉得状态图令人难以应付，但有 3 名参与者觉得它略显复杂，另有 1 名参与者觉得它非常复杂。14 名参与者中有 7 名认为与状态图进行交互很有用，例如查看状态上的统计分布以及与每个状态相关联的关键词。

模式摘要。14 位参与者中有 9 位认为，在模式摘要视图中查看模式，并根据特定模式筛选数据集很有用（中位数评分：6）。P2 提到："这有助于我们了解模型内部，并通过可能存在错误的模式找到错误。"具体而言，参与者还提到，模式摘要视图有助于调试。P8 说："我可以点击错误模式来查看相关句子。这有助于我找出模型为何通常会错误分类 *[*句子*]* 中具有这些模式的句子。"

实例搜索与筛选。 大多数参与者认为，与实例视图中的每个数据实例交互（中位数评分：7）以及搜索相似实例（中位数评分：7）很有用。P4 提到："我喜欢与每个标签对应的颜色，感觉这对查看示例很有帮助。我也喜欢它清楚地展示了示例的真实类别和预测结果。此外，这项功能还可以

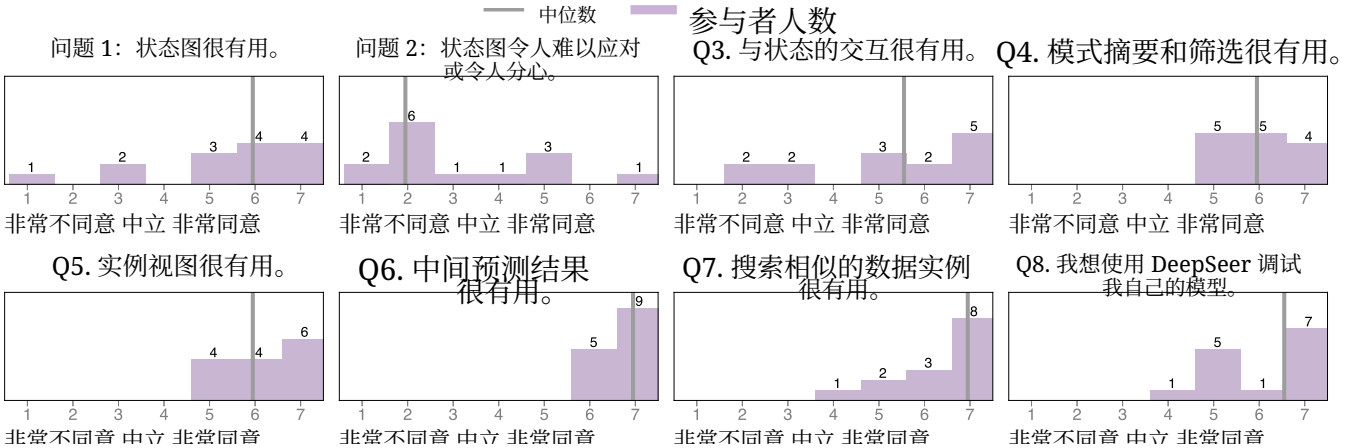


图 10：参与者 对 DeepSeer 工具特性的评分（1 表示“非常不同意”，7 表示“非常同意”）

按正确性、预测和真实标签筛选示例对我也很有帮助。”

局限性与建议。 14 位参与者中有 5 位指出，如果 DeepSeer 能提供有关模型准确性的其他统计信息（例如混淆矩阵），会更好。1 位参与者建议，添加解释的置信度分数有助于他们做出更明智的决策。1 位参与者认为状态图在理解上颇具挑战。P6 表示：“状态图光看起来似乎有点难以理解，对于初次打开此应用的用户而言，可能不会一眼就明白。”

认知负担。 在研究后调查中，参与者通过 NASA TLX 问卷 [21] 对研究的认知负荷进行了评分。图 11 展示了他们对 NASA TLX 五个问题的评分。我们发现，在匆忙程度、表现、努力和挫折感方面，使用 DeepSeer 与使用 LIME 之间没有显著差异（Welch t 检验：$p = 0.7731$、$p = 0.7244$、$p = 0.6916$ 和 $p = 0.5620$）。由于 DeepSeer 展示了更多有关模型行为的信息（例如状态图、模式视图以及按需提供的中间预测结果），使用 DeepSeer 的参与者感受到了更高的心理需求（中位数：5 vs. 4，Welch t 检验：$p = 0.0011$）。

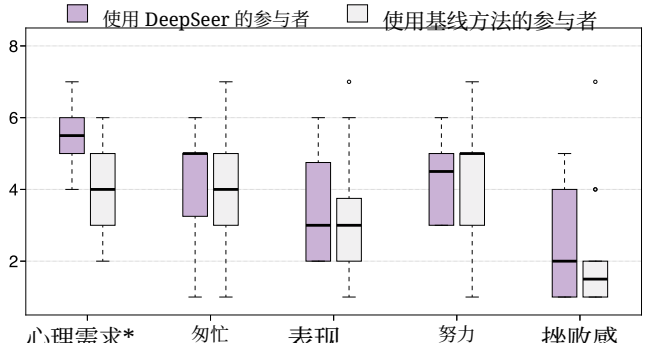


图 11：参与者对认知负荷的评分（1 表示“强烈不同意”，7 表示“强烈同意”；* 表示均值差异具有统计显著性。）

9 讨论

9.1 设计启示

用户研究结果表明，与基准工具 LIME [53] 相比，DeepSeer 能帮助用户更全面地理解指定模型，并在模型调试方面表现得更好。我们认为，这主要归功于 DeepSeer 对解释模型全局和局部行为的交互式支持。尽管已有少数研究讨论了全局和局部解释的重要性 [24, 45],，但我们的工作针对如何在统一界面中支持循环神经网络模型的全局和局部解释，提供了具体见解。

在 DeepSeer 中，全局解释主要通过状态图视图和模式摘要视图呈现。抽象后的状态图有助于用户理解隐藏状态及其复杂的状态之间的状态转移，而汇总后的文本模式则帮助用户快速识别模型学习到的、具有影响力或存在缺陷的模式。这些全局解释有助于提升用户的理解，并推动调试过程。尽管全局解释有诸多益处，我们仍发现，参与者需要实例级解释来结合上下文理解模型行为。具体而言，给定某个特定状态或文本模式时，用户研究的参与者常常会好奇它在不同文本中听起来如何。在研究后调查中，他们对中间预测结果功能给予了高度评价。DeepSeer 允许用户根据选定的状态或模式主动筛选实例，从而深入查看局部解释；用户也可以沿着状态图回溯，重新查看模型的全局行为。通过这些方式，全局和局部解释形成协同循环，帮助用户理解和调试模型。

此外，我们发现，用户关注给出的解释是如何从循环神经网络模型的内部决策过程中推导出来的。在使用 LIME 的 14 位参与者中，有 [53], 4 位质疑了 LIME 给出的解释（突出显示的关键词）。例如，P24 在研究后调查中评论道

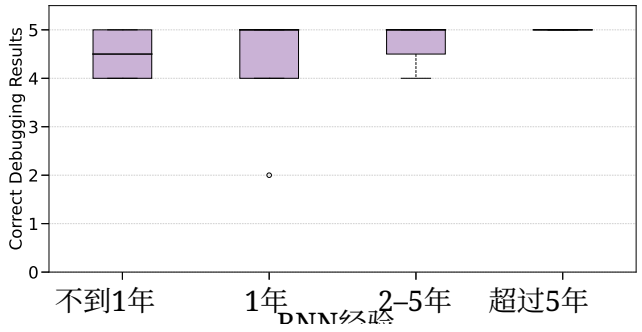


图 12：具有不同 RNN专业水平的 DeepSeer 用户在模型调试方面的表现。

：“我希望 *LIME* 能说明为什么某些词的不真诚或真诚得分较高。”作为一种更直观、可行的解决方案，DeepSeer 不仅传达输入中特定特征与预测结果之间的相关性，还传达模型的内部决策过程。DeepSeer 通过两种方式呈现模型的决策过程。首先，它在状态图视图中呈现模型不同内部状态之间的状态转移。其次，对于单个预测，它会呈现中间预测结果，以及这些结果与模型内部状态之间的对应关系。通过检查这一决策过程，用户能够更好地理解模型如何得出特定预测，并更加信任生成的解释。

作为一种交互式可解释人工智能工具，为用户提供可解释的说明同样十分重要，尤其是对于循环神经网络模型。需要注意的是，已有少数技术尝试将循环神经网络的决策过程可视化 [27, 59]。然而，它们通常只直接将每个隐藏状态的值可视化。例如，LSTMVis [59] 通过平行坐标可视化隐藏状态值的变化。鉴于隐藏状态值本质上是高维空间中的数值，因此很难解释其语义含义。为应对这一挑战，DeepSeer 将隐藏状态与文本语料库中相关联的词和短语相结合，并将它们之间的状态转移可视化为状态图。通过这种方式，内部决策过程对于非专业人士而言变得更易于理解。

9.2 目标用户与用户专业水平

DeepSeer 面向所有需要自行训练和调试循环神经网络模型的开发者。用户可能是经验丰富的机器学习开发者、刚开始学习循环神经网络的普通软件开发者，或是在课程项目中使用循环神经网络的学生。在用户研究中，我们招募了具有不同循环神经网络经验的参与者，其中 4 位参与者的 RNN经验少于 1 年，5 位有 1 年经验，3 位有 2—5 年经验，另有 2 位超过 5 年。进一步分析表明，尽管循环神经网络经验更丰富的参与者表现略好，但差异并不显著（图 12）。这说明 fDeepSeer 的有效性与他们的专业水平并无很强的相关性。

9.3 对不同机器学习任务和模型的泛化

尽管我们的工作仅在情感分析和主题建模任务上评估了 DeepSeer，但我们相信，DeepSeer 也能推广到不同的自然语言处理任务。若要将 DeepSeer 用于其他任务，可以考虑调整抽象状态的颜色映射机制。例如，对于机器翻译任务，可以根据词性标签为每个状态着色。通过检查每个状态的颜色及其关联词语，用户可以判断循环神经网络模型是否正确翻译了句子。

在本研究中，我们聚焦于循环神经网络（RNNs），这是一种用于处理序列数据的代表性模型架构。除循环神经网络外，DeepSeer 也可能用于解释循环神经网络的变体，例如双向长短期记忆网络 [37] 或 Transformer [60]。虽然双向长短期记忆网络的原理与朴素循环神经网络相似，但仍需对状态抽象方法进行一些调整。例如，应考虑收集模型在沿两个方向处理输入文本时的隐藏状态。由于 Transformer 具有置换不变性，它们会同时处理输入句子中的所有词，而不是按序列处理。因此，我们无法再将一个词与一个隐藏状态绑定。不过，可以将每个隐藏层的输出视为一个具体状态。随后，可以在不同隐藏层之间，而不是不同词之间，构建状态转移。

9.4 局限性与未来工作

用户研究设计的一个局限是，对比基线 LIME [53], 旨在生成局部解释，而非全局解释。因此，我们无法直接比较 DeepSeer 与 LIME 在全局解释方面的效果。此外，LIME 并非专为循环神经网络设计。尽管已有一些专门面向循环神经网络的工具，例如 LSTMVis [59] 和 RNNVis [43], ，但由于版本兼容性问题，我们多次尝试后仍无法在自己的循环神经网络模型上运行它们。LSTMVis 和 RNNVis 都是多年前基于过时的深度学习框架（TensorFlow r0.12 和 Torch 7）构建的，因此无法再用于分析由后续版本框架构建的深度学习模型。重新实现这些工具需要付出大量努力。因此，我们认为重新实现它们超出了本工作的范围。另一种方案是通过禁用某些关键特征来创建 DeepSeer 的变体，作为基线；这有助于我们更好地将 DeepSeer 的成功归因于各项特征。未来值得对此进行研究。

由于我们是来自一所R1级大学的研究人员，无法接触到在工作中构建循环神经网络模型的专业开发者和数据科学家。因此，我们招募了有构建循环神经网络模型经验的研究生。与研究生相比，机器学习从业者或许能分享更多有趣的见解。

此外，我们的用户研究仅评估了DeepSeer在情感分析和主题分类任务中的循环神经网络。为了全面考察DeepSeer的实用性，可以考虑利用DeepSeer来理解和调试用于其他任务（如机器翻译）的循环神经网络模型，而不仅限于文本分类。

此外，我们目前的设计仅支持对一个循环神经网络模型进行可视化和分析。借助 DeepSeer 识别出漏洞后，还需要重新训练该循环神经网络模型。因此，一个可能的未来方向是开发用于比较

两个或更多版本的循环神经网络模型 [46] 的工具支持。此外，还可以通过设计支持模型跟踪 [4] 和模型选择 [10] 的工具来改进 DeepSeer。

10 结论

本文介绍了一种名为 DeepSeer 的新型系统，旨在帮助机器学习开发者理解和调试循环神经网络。DeepSeer 使用一种状态抽象方法，将循环神经网络模型中语义相似的隐藏状态归并，并将其抽象为有限状态机。借助 DeepSeer，用户可以探索模型的全局和局部行为，并调试错误预测。我们通过一项针对 28 名模型开发者、涉及两种不同循环神经网络模型的被试间用户研究，展示了 DeepSeer 的实用性和易用性。结果表明，与一种流行的可解释人工智能技术 LIME 相比，DeepSeer 紧密协同的视图帮助开发者更深入地理解循环神经网络模型。此外，使用 DeepSeer 的参与者能够找出更多错误预测的根本原因，并提出更多可执行计划来改进循环神经网络模型。最后，我们讨论了 DeepSeer 带来的设计启示，并提出了若干有前景的未来研究方向。

致谢

我们感谢用户研究中的所有匿名参与者以及匿名审稿人提供的宝贵反馈。本研究部分得到了 Amii RAP 项目、加拿大 CIFAR 人工智能主席计划、加拿大自然科学与工程研究委员会（NSERC 编号：RGPIN-2021-02549、RGPAS-2021-00034、DGECR-2021-00019），以及日本学术振兴会科学研究费补助金（编号：JP20H04168）和 JST-Mirai 计划资助（编号：JPMJMI20B8）的支持。

参考文献

[1] Martín Abadi、Ashish Agarwal、Paul Barham、Eugene Brevdo、Zhifeng Chen、Craig Citro、Greg S. Corrado、Andy Davis、Jeffrey Dean、Matthieu Devin、Sanjay Ghemawat、Ian Goodfellow、Andrew Harp、Geoffrey Irving、Michael Isard、Yangqing Jia、Rafal Jozefowicz、Lukasz Kaiser、Manjunath Kudlur、Josh Levenberg、Dandelion Mané、Rajat Monga、Sherry Moore、Derek Murray、Chris Olah、Mike Schuster、Jonathon Shlens、Benoit Steiner、Ilya Sutskever、Kunal Talwar、Paul Tucker、Vincent Vanhoucke、Vijay Vasudevan、Fernanda Viégas、Oriol Vinyals、Pete Warden、Martin Wattenberg、Martin Wicke、Yuan Yu 和 Xiaoqiang Zheng。2015年。TensorFlow：异构系统上的大规模机器学习。https://www.tensorflow.org/ 软件可从 tensorflow.org 获取。[2] Amina Adadi 和 Mohammed Berrada。2018年。窥探黑箱内部：可解释人工智能（XAI）综述。*IEEE* 访问 6（2018年），52138–52160。[3] Julius Adebayo、Michael Muelly、Ilaria Liccardi 和 Been Kim。2020年。模型解释的调试测试。神经信息处理系统进展 33（2020年），700–712。[4] Saleema Amershi、Max Chickering、Steven M Drucker、Bongshin Lee、Patrice Simard 和 Jina Suh。2015年。Modeltracker：重新设计机器学习性能分析工具。载于第*33*届美国计算机协会计算系统中的人为因素年会论文集。337–346。[5] Saleema Amershi、Dan Weld、Mihaela Vorvoreanu、Adam Fourney、Besmira Nushi、Penny Collisson、Jina Suh、Shamsi Iqbal、Paul N Bennett、Kori Inkpen 等。2019年。人机交互指南。载于*2019*年*CHI*计算系统中的人为因素会议论文集。1–13。[6] Alejandro Barredo Arrieta、Natalia Díaz-Rodríguez、Javier Del Ser、Adrien Bennetot、Siham Tabik、Alberto Barbado、Salvador García、Sergio Gil-López、Daniel Molina、Richard Benjamins 等。2020年。可解释人工智能（XAI）：迈向负责任的人工智能的概念、分类、机遇与挑战。信息融合 58（2020年），82–115。[7] Elnaz Barshan、Marc-Etienne Brunet 和 Gintare Karolina Dziugaite。2020年。Relatif：通过相对影响识别具有解释力的训练样本。载于国际人工智能与统计会议。PMLR，1899–1909。[8] David Bau、Bolei Zhou、Aditya Khosla、Aude Oliva 和 Antonio Torralba。2017年。网络解剖：量化深层视觉表示的可解释性。IEEE计算机视觉与模式识别会议论文集。6541–6549。[9] Tom Bocklisch、Joey Faulkner、Nick Pawlowski 和 Alan Nichol。2017年。Rasa：开源语言理解与对话管理。arXiv预印本 arXiv:1712.05181（2017年）。[10] Markus Bögl、Wolfgang Aigner、Peter Filzmoser、Tim La mmarsch、Silvia Miksch 和 Alexander Rind。2013年。时间序列分析中的模型选择可视分析。IEEE可视化与计算机图形学汇刊 19，12（2013年），2237–2246。[11] Carrie J Cai、Jonas Jongejan 和 Jess Holbrook。2019年。基于示例的解释对机器学习界面的影响。载于智能用户界面国际会议论文集。258–262。[12] Kyunghyun Cho、Bart van Merriënboer、Dzmitry Bahdanau 和 Yoshua Bengio。2014年。神经机器翻译的性质：编码器–解码器方法。载于 SSST-8，即第八届统计机器翻译语法、语义与结构研讨会论文集。103–111。[13] Arun Das 和 Paul Rad。2020年。可解释人工智能（xai）中的机遇与挑战：综述。arXiv预印本 arXiv:2006.11371（2020年）。[14] Jonathan Dodge、Q Vera Liao、Yunfeng Zhang、Rachel KE Bellamy 和 Casey Dugan。2019年。解释模型：解释如何影响公平性判断的实证研究。载于智能用户界面国际会议论文集。275–285。[15] Xiaoning Du、Yi Li、Xiaofei Xie、Lei Ma、Yang Liu 和 Jianjun Zhao。2020年。Marble：有状态深度学习系统的基于模型的鲁棒性分析。载于第35届 IEEE/ACM自动化软件工程国际会议论文集。423–435。[16] Xiaoning Du、Xiaofei Xie、Yi Li、Lei Ma、Yang Liu 和 Jianjun Zhao。2019年。Deepstellar：有状态深度学习系统的基于模型的定量分析。载于2019年第27届 ACM 欧洲软件工程会议与软件工程基础研讨会联合会议论文集。477–487。[17] Mary T Dzindolet、Sc ott A Peterson、Regina A Pomranky、Linda G Pierce 和 Hall P Beck。2003年。信任在自动化依赖中的作用。人机研究国际期刊 58，6（2003年），697–718。[18] Jeffrey L Elman。1990年。发现时间中的结构。认知科学 14，2（1990年），179–211。[19] Philippe Fournier-Viger、Antonio Go mariz、Ted Gueniche、Espérance Mwamikazi 和 Rincy Thomas。2013年。TKS：高效挖掘前k个序列模式。载于高级数据挖掘与应用国际会议。施普林格，109–120。[20] Antonio Gulli。2005年。AG新闻文章语料库。http://groups.di.unipi.it/ ~gulli/AG_corpus_of_news_articles.html [21] Sandra G Hart 和 Lowell E Staveland。1988年。NASA任务负荷指数（任务负荷指数）的开发：实证与理论研究结果。载于心理学进展。第52卷。Elsevier，139–183。[22] Jonathan L Herlocker、Joseph A Konstan 和 John Riedl。2000年。解释协同筛选推荐。载于2000年美国计算机协会计算机支持的协同工作会议论文集。241–250。[23] Sepp Hochreiter 和 Jürgen Schmidhuber。1997年。长短期记忆。神经计算 9，8（1997年），1735–1780。[24] Fred Hohman、Andrew Head、Rich Caruana、Robert DeLine 和 Steven M Drucker。2019年。Gamut：理解数据科学家如何理解机器学习模型的设计探针。载于2019年 CHI计算系统中的人为因素会议论文集。1–13。[25] Zhihua Jin、Yong Wang、Qianwen Wang、Yao Ming、Tengfei Ma 和 Huamin Qu。2022年。Gnnlens：用于图神经网络预测误差诊断的可视分析方法。IEEE可视化与计算机图形学汇刊（2022年）。[26] Minsuk Kahng、Pierre Y Andrews、Aditya Kalro 和 Duen Horng Chau。2017年。Activis：工业规模深度神经网络模型的可视探索。IEEE可视化与计算机图形学汇刊 24，1（2017年），88–97。[27] Andrej Karpathy、Justin Johnson 和 Li Fei-Fei。2015年。循环网络的可视化与理解。arXiv预印本 arXiv:1506.02078（2015年）。[28] Harmanpreet Kaur、Harsha Nori、Samuel Jenkins、Rich Caruana、Hanna Wallach 和 Jennifer Wortman Vaughan。2020年。解读可解释性：理解数据科学家对机器学习可解释性工具的使用。载于2020年 CHI计算系统中的人为因素会议论文集。1–14。[29] Been Kim、Martin Wattenberg、Justin Gilmer、Carrie Cai、James Wexler、Fernanda Viegas 等。2018年。超越特征归因的可解释性：使用概念激活向量（tcav）进行定量测试。载于国际机器学习会议。PMLR，2668–2677。[30] René F Kizilcec。2016年。信息量有多大？透明度对算法界面信任的影响。载于2016年 CHI计算系统中的人为因素会议论文集。2390–2395。[31] Pang Wei Koh、Kai-Siang Ang、Hubert H. K. Teo 和 Percy Liang。2019年。关于使用影响函数衡量群体效应的准确性。载于神经信息处理系统进展32（2019年）。5255–5265。

[32] Pang Wei Koh 和 Percy Liang。2017年。通过影响函数理解黑箱预测。载于 国际机器学习会议。PMLR，1885–1894。[33] Jiwei Li、Xinlei Chen、Eduard Hovy 和 Dan Jurafsky。2016年。NLP中的神经模型可视化与理解。载于 计算语言学协会北美分会*2016*年会议论文集：人类语言技术。681–691。[34] Rongjian Li、Wenlu Zhang、Heung-Il Suk、Li Wang、Jiang Li、Dinggang Shen 和 Shuiwang Ji。2014年。基于深度学习的影像数据补全，用于改进脑部疾病诊断。载于 国际医学图像计算与计算机辅助干预会议。施普林格，305–312。[35] Q Vera Liao、Daniel Gruen 和 Sarah Miller。2020年。质疑人工智能：为可解释人工智能用户体验提供设计实践指导。载于 *2020*年*CHI*计算系统中的人为因素会议论文集。1–15。[36] Zachary C Lipton。2018年。模型可解释性的神话：在机器学习中，可解释性这一概念既重要又模糊。*Queue* 16，3（2018年），31–57。[37] Gang Liu 和 Jiabao Guo。2019年。用于文本分类的带注意力机制和卷积层的双向长短期记忆网络。*Neurocomputing* 337（2019年），325–338。[38] Shusen Liu、Zhimin Li、Tao Li、Vivek Srikumar、Valerio Pascucci 和 Peer-Timo Bremer。2018年。Nlize：用于分析和解释自然语言推理模型的扰动驱动的可视化探查工具。*IEEE*可视化与计算机图形学汇刊 25，1（2018年），651–660。[39] Scott M Lundberg 和 Su-In Lee。2017年。解释模型预测的统一方法。载于 神经信息处理系统进展 *30*，I. Guyon、U. V. Luxburg、S. Bengio、H. Wallach、R. Fergus、S. Vishwanathan 和 R. Garnett（编）。Curran Associates, Inc.，4765–4774。http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf [40] Shiqing Ma、Yousra Aafer、Zhaogui Xu、Wen-Chuan Lee、Juan Zhai、Yingqi Liu 和 Xiangyu Zhang。2017年。LAMP：通过导数计算为基于图的机器学习算法提供数据溯源。载于 *2017*年第*11*届786 797届软件工程基础联合会议。–。[41] Shiqing Ma、Yingqi Liu、Wen-Chuan Lee、Xiangyu Zhang 和 Ananth Grama。2018年。MODE：通过状态差分分析和输入选择自动进行神经网络模型调试。载于 *2018*年第*26*届 *ACM* 欧洲软件工程会议与软件工程基础研讨会联合会议论文集。175–186。[42] Geoffrey J McLachlan 和 Kaye E Basford。1988年。混合模型：推理与聚类应用。第38卷。M. Dekker，纽约。[43] Yao Ming、Shaozu Cao、Ruixiang Zhang、Zhen Li、Yuanzhe Chen、Yangqiu Song 和 Huamin Qu。2017年。理解循环神经网络的隐藏记忆。载于 *2017*年 *IEEE*可视分析科学与技术会议（*VAST*）。IEEE，13–24。[44] Christoph Molnar。2020年。可解释机器学习。Lulu.com。[45] Grégoire Montavon、Wojciech Samek 和 Klaus-Robert Müller。2018年。解释和理解深度神经网络的方法。数字信号处理 73（2018年），1–15。[46] Sugeerth Murugesan、Sana Malik、Fan Du、Eunyee Koh 和 Tuan Manh Lai。2019年。Deepcompare：深度学习模型性能的可视化与交互式比较。*IEEE*计算机图形学与应用 39，5（2019年），47–59。[47] Takamasa Okudono、Masaki Waga、Taro Sekiyama 和 Ichiro Hasuo。2020年。通过在状态空间上的回归从循环神经网络中提取加权自动机。载于*AAAI*人工智能会议论文集，第34卷。5306–5314。

[48] Chris Olah、Arvind Satyanarayan、Ian Johnson、Shan Carter、Ludwig Schubert、Katherine Ye 和 Alexander Mordvintsev。2018年。可解释性的构成要素。*Distill* 3, 3（2018年），e10。[49] Zi Peng、Jinqiu Yang、Tse-Hsun Chen 和 Lei Ma。2020年。初探机器学习模型在复杂自动驾驶系统中的集成：以阿波罗为例。载于第*28*届美国计算机协会欧洲软件工程会议与软件工程基础研讨会联合会议论文集。1240–1250。[50] Quora。2019年。Quora非真诚问题分类。https://www.kaggle.com/ c/quora-insincere-questions-classification/data [51] Guillaume Rabusseau、Tianyu Li 和 Doina Precup。2019年。通过谱学习连接加权自动机与循环神经网络。载于第*22*届人工智能与统计国际会议。PMLR，1630–1639。[52] Amir Hossein Akhavan Rahnama 和 Henrik Boström。2019年。LIME框架中的数据偏移与标签偏移研究。*arXiv*预印本 *arXiv:1910.14421*（2019年）。[53] Marco Tulio Ribeiro、Sameer Singh 和 Carlos Guestrin。2016年。"我为什么应该信任你？"解释任意分类器的预测。载于第*22*届美国计算机协会*SIGKDD*国际知识发现与数据挖掘会议论文集。1135–1144。[54] Marco Tulio Ribeiro、Sameer Singh 和 Carlos Guestrin。2018年。Anchors：高精度的模型无关解释。载于*AAAI*人工智能会议论文集，第32卷。[55] Tobias Schneider、Joana Hois、Alischa Rosenstein、Sabiha Ghellal、Dimitra Theofanou-Fülbier 和 Ansgar RS Gerlicher。2021年。ExplAIn Yourself! 自动驾驶中的积极用户体验透明化。载于*2021*年*CHI*计算系统中的人为因素会议论文集。1–12。[56] Ramprasaath R Selvaraju、Michael Cogswell、Abhishek Das、Ramakrishna Vedantam、Devi Pariah 和 Dhruv Batra。2017年。Grad-CAM：通过基于梯度的定位，从深度网络中获得可视化解释。载于*IEEE*国际计算机视觉会议论文集。618–626。[57] Karen Simonyan、Andrea Vedaldi 和 Andrew Zisserman。2013年。深入卷积网络内部：图像分类模型与显著性图的可视化。*arXiv*预印本 *arXiv:1312.6034*（2013年）。[58] Daniel Smilkov、Nikhil Thorat、Been Kim、Fernanda Viégas 和 Martin Wattenberg。2017年。SmoothGrad：通过添加噪声去除噪声。*arXiv*预印本 *arXiv:1706.03825*（2017年）。[59] Hendrik Strobelt、Sebastian Gehrmann、Hanspeter Pfister 和 Alexander M Rush。2017年。LSTMVis：用于可视化分析循环神经网络中隐藏状态动态的工具。*IEEE*可视化与计算机图形学汇刊24, 1（2017年），667–676。[60] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Łukasz Kaiser 和 Illia Polosukhin。2017年。注意力机制就够了。神经信息处理系统进展30（2017年）。[61] Gail Weiss、Yoav Goldberg 和 Eran Yahav。2018年。使用查询和反例从循环神经网络中提取自动机。载于国际机器学习会议。PMLR，5247–5256。[62] Xiaofei Xie、Wenbo Guo、Lei Ma、Wei Le、Jian Wang、Lingjun Zhou、Yang Liu 和 Xinyu Xing。2021年。RNNrepair：通过基于模型的分析自动修复循环神经网络。载于国际机器学习会议。PMLR，11383–11392。[63] Bing Yu、Hua Qi、Qing Guo、Felix Juefei-Xu、Xiaofei Xie、Lei Ma 和 Jianjun Zhao。2022年。DeepRepair：真实世界运行环境中深度神经网络的风格引导修复。*IEEE*可靠性汇刊71, 4（2022 1401 1416 10 1109 2021 3096332），–。[64] https://doi.org/。/TR.。Matthew D Zeiler 和 Rob Fergus。卷积网络的可视化与理解。载于欧洲计算机视觉会议。施普林格，818–833。[65] Xiang Zhang、Junbo Zhao 和 Yann LeCun。2015年。用于文本分类的字符级卷积网络。神经信息处理系统进展28（2015年）。

状态抽象

本节介绍循环神经网络模型状态抽象的技术细节。其核心思想是利用训练数据提取循环神经网络模型的所有可能隐藏状态，然后将相似的隐藏状态分组，以构建有限状态机（FSM）。算法 A1 展示了循环神经网络模型状态抽象的流程。

算法 A1： 循环神经网络模型的状态抽象。

输入： 训练好的循环神经网络模型 R、训练数据 X_{train}、主成分分析维度 k、状态数量 n

输出： 主成分分析模型 P，高斯混合模型 G，一个抽象模型 $A = \{P, G\}$

```
H ← {};
遍历 x ∈ X_train 执行
    H ← 记录_隐藏_状态(R, x);
    H.append(H);
结束
P = 主成分分析(H, k);
G = 高斯混合模型(P(H), n);
返回 A;
```

算法 A1 有两个输入：训练好的循环神经网络模型 R 和训练数据 X_{train}，以及两个参数：主成分分析（PCA）维度 p 和抽象状态的数量n。给定一个训练好的循环神经网络模型，我们首先遍历所有训练数据 X_{train}，收集该循环神经网络模型的所有可能隐藏状态 H（第 1 至 5 行）。第 3 行记录循环神经网络模型在处理特定输入实例 x时的所有隐藏状态 H。假设一个输入实例 x有 l个词，那么循环神经网络模型按顺序处理这些 l词时，将产生 l个不同的隐藏状态。例如，给定句子“I love Machine Learning”，循环神经网络模型将依次处理四个词：“I”、“love”、“machine”和“learning”。因此，循环神经网络模型处理该句子时将产生并记录四个不同的隐藏状态向量。

使用训练数据记录所有隐藏状态后，我们创建一个主成分分析模型 k，将这些隐藏状态的维度降至 p（第 6 行）。同时，我们也得到一个主成分分析模型 P。现在，我们将 $|P(H)|$ 降维后的隐藏状态抽象为 n 个抽象状态。不同于采用基于网格的方法的先前工作 [16]，我们使用高斯混合模型（高斯混合模型 [42]）G对这些降维后的隐藏状态进行聚类（第 7 行）。执行第 7 行后，我们得到一个高斯混合模型 G。至此，我们构建了抽象模型As，其中包含一个主成分分析模型 P 和一个高斯混合模型 G。请注意，主成分分析和高斯混合模型均使用 scikit-learn 实现，除“n_components”外，其他参数均采用默认值。

在我们的使用场景和用户研究中，我们将主成分分析的维度 k 设为 20，状态数量 n 设为 40。我们进一步表明，与原始循环神经网络模型相比，采用此设置的抽象模型能够给出一致的预测，详见附录 B。

B 状态抽象的忠实度 ION

在本节中，我们展示抽象模型（即有限状态机）在三个不同任务中能够像原始循环神经网络模型一样做出一致的预测，其中一个任务来自使用场景（第 6 节），另外两个来自用户研究（第 7.2 节）。我们衡量有限状态机与原始循环神经网络之间的预测一致性。假设数据集为 $X = \{x_1, x_2, ..., x_N\}$，抽象模型为 F，循环神经网络模型为 R。预测一致性可通过公式 2 计算。

$$prediction_consistency = \frac{\sum_{i=1}^{N} \mathcal{F}(x_i) == R(x_i)}{N} \quad (2)$$

表 B1 展示了两个模型在各项任务的训练数据和测试数据上的预测一致性。可以看出，对于二分类模型（Toxic 和 Quora），与原始循环神经网络模型相比，抽象模型在训练数据和测试数据上都能给出高度一致的预测（一致性分别为 99 % 和 97%）。对于多分类模型（AGNews），抽象模型仍能给出非常一致的预测（一致性为 86%）。这些结果证明了我们的状态抽象技术具有较高的忠实度。

C 抽象状态的数量

在用户研究中，我们将抽象状态的数量设为 40。这个数值是通过经验确定的，旨在让相对于原始循环神经网络的预测一致性与检查状态图所需的认知努力之间达到良好平衡。为进一步说明这一设置不会影响抽象模型的忠实度，我们在图 C1 中报告了所有三个模型的抽象模型预测一致性随状态数量变化的情况。

如我们所见，状态数量越少，预测一致性越低。然而，当状态数量大于 40 时，预测一致性基本保持不变。因此，在整个动机示例和用户研究中，我们都选择使用 40 个状态。

D 机器学习任务

D.1 机器学习任务 1：情感分析（Quora 数据集）

任务描述：

在这项任务中，参与者获得了一个使用 Quora 数据集训练的循环神经网络模型。

Quora 数据集收集自 quora.com，其中每条文本都被标记为“真诚”或“不真诚”。不真诚的问题被定义为意在表达某种观点，而非寻求有帮助的答案的问题。

参与者首先使用该工具来理解模型。他们被要求使用该工具探索模型在训练数据和测试数据上的行为与表现。探索之后，参与者需要分享他们的发现，例如：他们是否获得了什么洞见？是否发现了什么漏洞？他们会如何改进这个模型？

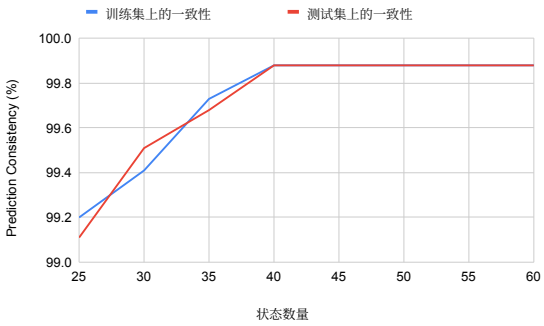
随后，参与者获得了 5 个误分类句子。参与者总共有 10 分钟完成以下任务：对于每个句子，参与者需要借助 DeepSeer 找出该句子被误分类的原因。

D.2 机器学习任务 2：主题分类（AGNews 数据集）

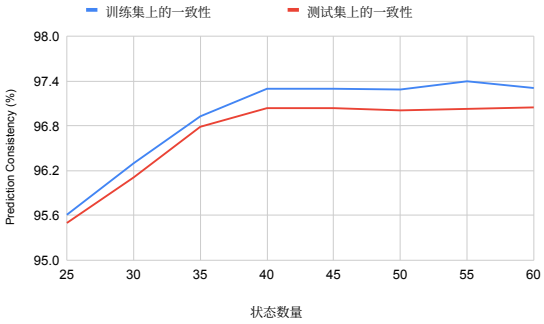
任务描述：

表 B1：抽象模型预测与原始循环神经网络预测之间的定量比较。

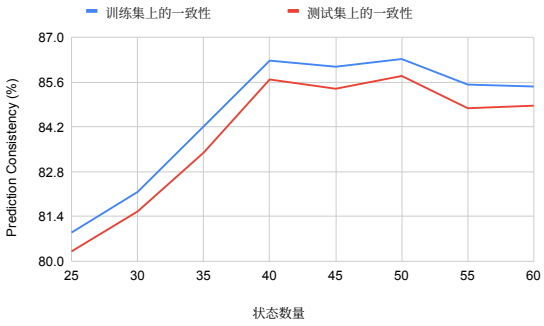
	训练集上的预测一致性	测试集上的预测一致性
Toxic	99.88%	99.88%
Quora	97.30%	97.04%
AGNews	86.28%	85.59%



(a) Toxic 数据集上原始循环神经网络模型与抽象模型之间的预测一致性。



（b）Quora 数据集上原始循环神经网络模型与抽象模型之间的预测一致性。



（c）AGNews 数据集上原始循环神经网络模型与抽象模型之间的预测一致性。

图 C1：三个循环神经网络模型的抽象模型的预测一致性与状态数量的关系。

在此任务中，参与者获得了一个在 AGNews 数据集上训练的循环神经网络模型。

AGNews 数据集由新闻文章组成。该循环神经网络模型将每条文本分类到不同主题中，包括世界、体育、商业和科学/技术。

参与者首先使用该工具了解模型。他们被要求使用该工具探索模型在训练数据和测试数据上的行为及表现。探索结束后，参与者需要分享他们的发现，例如：他们是否获得了任何洞见？是否发现了任何漏洞？他们会如何改进该模型？

随后，参与者获得了 5 条误分类句子。参与者总共有 10 分钟完成以下任务：对于每条句子，参与者需要借助 DeepSeer 找出该句子被误分类的原因。

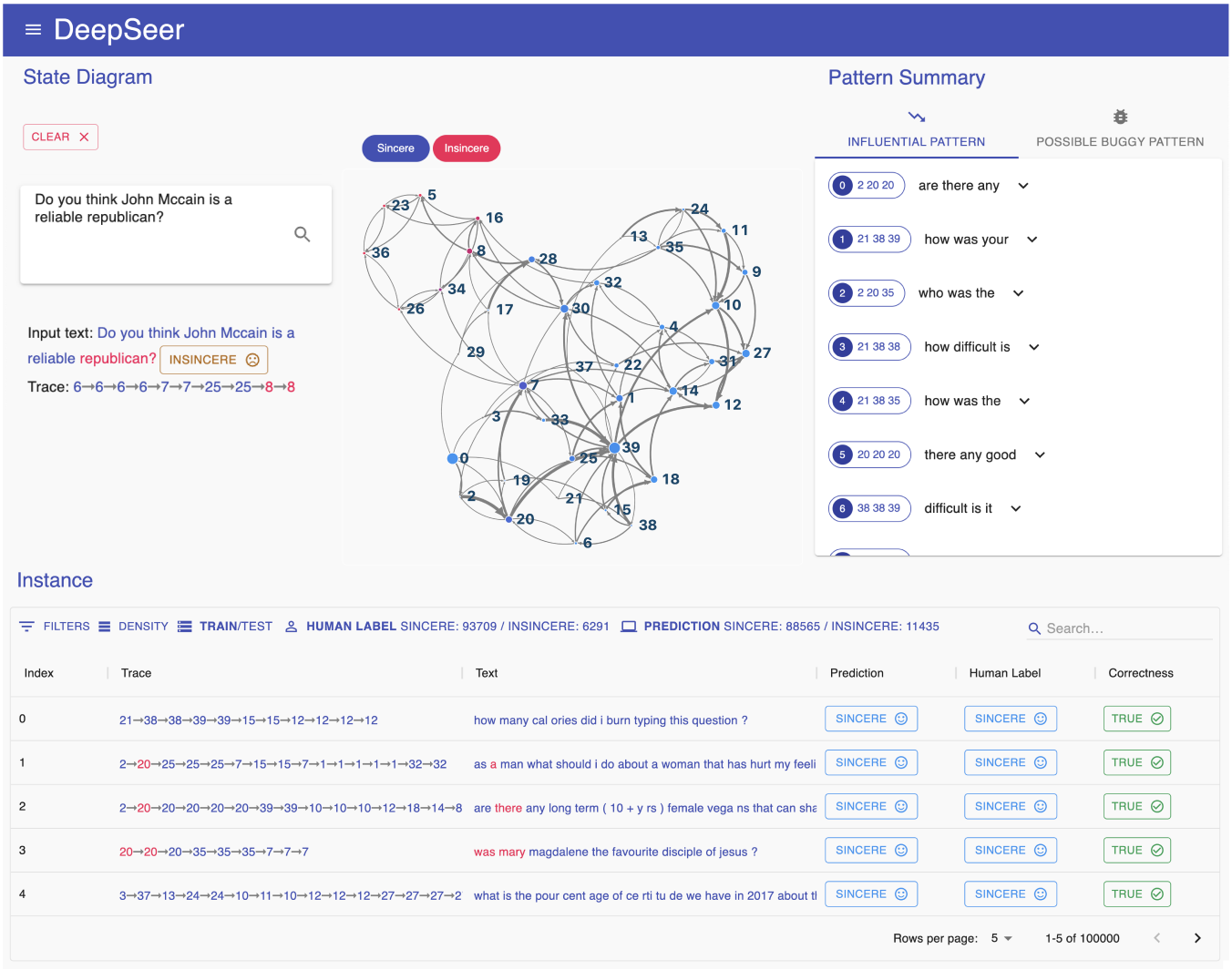


图 D2：DeepSeer 用于机器学习任务 1（Quora 数据集）的界面。

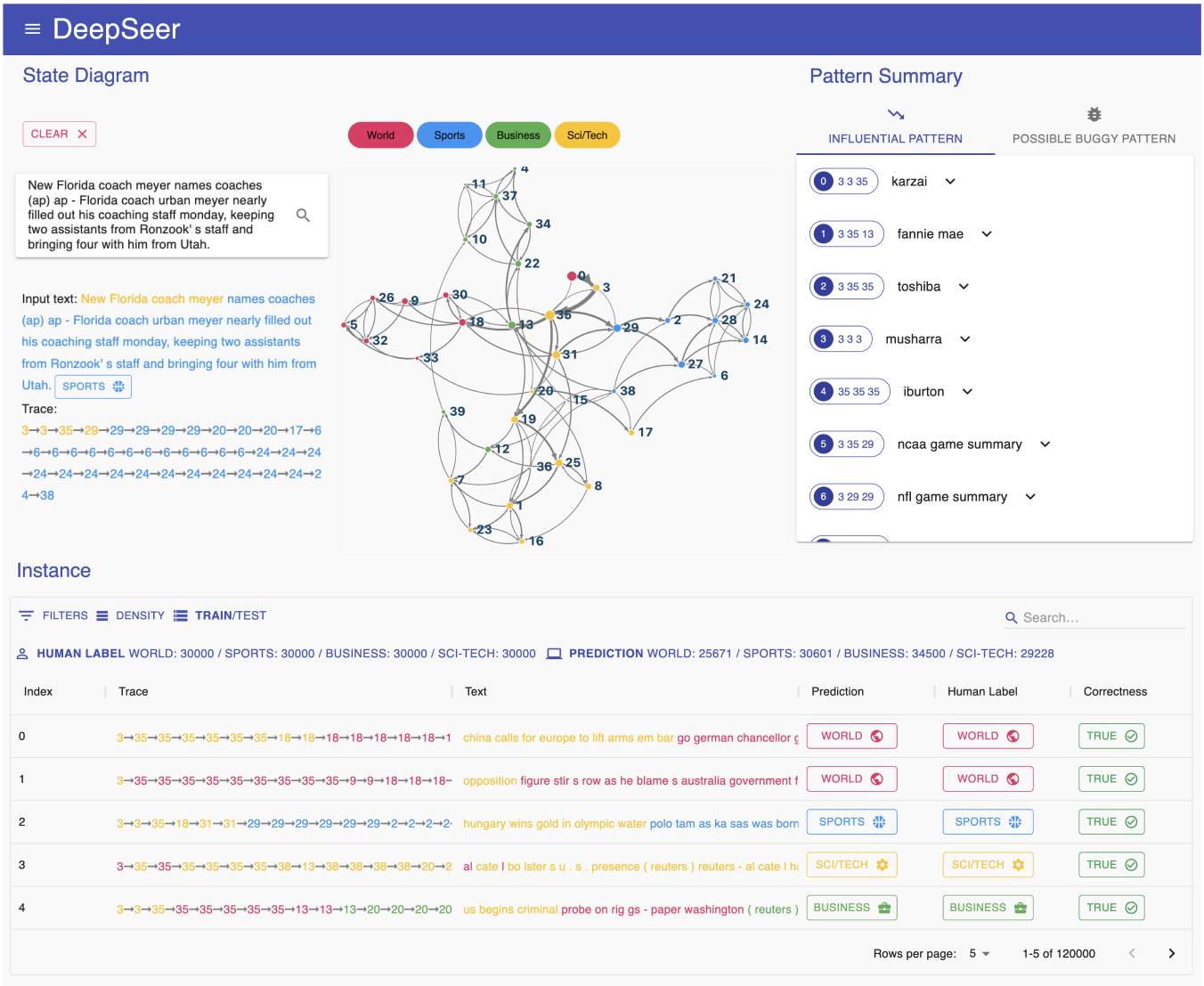


图 D3：用于机器学习任务 2（AGNews 数据集）的 DeepSeer 界面。