



自然语言处理导论

张奇 桂韬 黄萱菁

2023 年 1 月 1 日

数学符号

数与数组

α	标量
α	向量
A	矩阵
\mathbf{A}	张量
I_n	n 行 n 列单位矩阵
v_w	单词 w 的分布式向量表示
e_w	单词 w 的独热向量表示: $[0,0,\dots,1,0,\dots,0]$, w 下标处元素为 1

索引 |

α_i	向量 α 中索引 i 处的元素
α_{-i}	向量 α 中除索引 i 之外的元素
$w_{i:j}$	序列 w 中从第 i 个元素到第 j 个元素组成的片段或子序列
A_{ij}	矩阵 A 中第 i 行、第 j 列处的元素
$A_{i:}$	矩阵 A 中第 i 行
$A_{:j}$	矩阵 A 中第 j 列
A_{ijk}	三维张量 \mathbf{A} 中索引为 (i,j,k) 处元素
$\mathbf{A}_{::i}$	三维张量 \mathbf{A} 中的一个二维切片

集合

\mathbb{A}	集合
\mathbb{R}	实数集合
$0, 1$	含 0 和 1 的二值集合
$0, 1, \dots, n$	含 0 和 n 的正整数的集合
$[a, b]$	a 到 b 的实数闭区间
$(a, b]$	a 到 b 的实数左开右闭区间

线性代数

\mathbf{A}^\top	矩阵 \mathbf{A} 的转置
$\mathbf{A} \odot \mathbf{B}$	矩阵 \mathbf{A} 与矩阵 \mathbf{B} 的 Hardamard 乘积
$\det \mathbf{A}^\top$	矩阵 \mathbf{A} 的行列式
$[\mathbf{x}; \mathbf{y}]$	向量 \mathbf{x} 与 \mathbf{y} 的拼接
$[\mathbf{U}; \mathbf{V}]$	矩阵 \mathbf{A} 与 \mathbf{V} 沿行向量拼接
$\mathbf{x} \cdot \mathbf{y}$ 或 $\mathbf{x}^\top \mathbf{y}$	向量 \mathbf{x} 与 \mathbf{y} 的点积

微积分

$\frac{dy}{dx}$	y 对 x 的导数
$\frac{\partial y}{\partial x}$	y 对 x 的偏导数
$\nabla_{\mathbf{x}} y$	y 对向量 \mathbf{x} 的梯度
$\nabla_{\mathbf{X}} y$	y 对矩阵 \mathbf{X} 的梯度
$\nabla_{\mathbf{X}} y$	y 对张量 \mathbf{X} 的梯度

概率与信息论

$a \perp b$	随机变量 a 与 b 独立
$a \perp b \mid c$	随机变量 a 与 b 关于 c 条件独立
$P(a)$	离散变量概率分布
$p(a)$	连续变量概率分布
$a \sim P$	随机变量 a 服从分布 P
$\mathbb{E}_{x \sim P}[f(x)]$ 或 $\mathbb{E}[f(x)]$	$f(x)$ 在分布 $P(x)$ 下的期望
$\text{Var}(f(x))$	$f(x)$ 在分布 $P(x)$ 下的方差
$\text{Cov}(f(x), g(x))$	$f(x)$ 与 $g(x)$ 在分布 $P(x)$ 下的协方差
$H(f(x))$	随机变量 x 的信息熵
$D_{KL}(P \parallel Q)$	概率分布 P 与 Q 的 KL 散度
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	均值为 $\boldsymbol{\mu}$ 、协方差为 $\boldsymbol{\Sigma}$ 的高斯分布

数据与概率分布

\mathbb{X}	数据集
$\mathbf{x}^{(i)}$	数据集中第 i 个样本（输入）
$\mathbf{y}^{(i)}$ 或 $y^{(i)}$	第 i 个样本 $\mathbf{x}^{(i)}$ 的标签（输出）

函数

$f : \mathcal{A} \longrightarrow \mathcal{B}$	由定义域 \mathcal{A} 到值域 \mathcal{B} 的函数（映射） f
$f \circ g$	f 与 g 的复合函数
$f(\mathbf{x}; \boldsymbol{\theta})$	由参数 $\boldsymbol{\theta}$ 定义的关于 \mathbf{x} 的函数（也可以直接写作 $f(\mathbf{x})$, 省略 $\boldsymbol{\theta}$ ）
$\log x$	x 的自然对数函数
$\sigma(x)$	Sigmoid 函数 $\frac{1}{1 + \exp(-x)}$
$\ \mathbf{x}\ _p$	\mathbf{x} 的 L^p 范数
$\ \mathbf{x}\ $	\mathbf{x} 的 L^2 范数
$\mathbf{1}^{\text{condition}}$	条件指示函数：如果 condition 为真，则值为 1；否则值为 0

本书中常用写法

- 给定词表 \mathbb{V} , 其大小为 $|\mathbb{V}|$
- 序列 $\mathbf{x} = x_1, x_2, \dots, x_n$ 中第 i 个单词 x_i 的词向量 \mathbf{v}_{x_i}
- 损失函数 \mathcal{L} 为负对数似然函数： $\mathcal{L}(\boldsymbol{\theta}) = -\sum_{(x,y)} \log P(y|x_1 \dots x_n)$
- 算法的空间复杂度为 $\mathcal{O}(mn)$

目 录

14 模型可解释性	1
14.1 可解释性概述	1
14.1.1 可解释的分类	2
14.1.2 解释的评价	3
14.2 解释性分析方法	5
14.2.1 局部分析方法	5
14.2.2 全局分析方法	11
14.3 自然语言处理算法解释分析方法	14
14.3.1 模型解释性分析算法	14
14.3.2 数据解释分析方法	18
14.3.3 可解释评估	20
14.4 延伸阅读	22
14.5 习题	22

14. 模型可解释性

如前所述，目前绝大部分自然语言处理算法都是基于统计机器学习方法，这些数据驱动的算法已经在绝大部分任务上取得了良好的性能。但是，以深度神经网络方法为代表的“黑盒”模型缺乏可解释性。我们不能理解数百亿甚至是数万亿参数中的每个维度的含义，造成了深度学习模型本质上的不可解释性。然而，我们又迫切的需要了解模型是否真正符合人类语言的习惯，机器在语言处理任务中的决策与人类的决策过程有何异同，数据驱动的统计模型与人类语言认知系统的差异等问题。这些问题一方面关系到如何进一步提升自然语言处理算法的处理效果以及鲁棒性，另一方面如果不能很好的解决这些问题，就会给自然语言处理算法在关键业务中地应用带来了极大的风险和挑战。在医疗诊断、金融预测、司法审判等高风险场景中是否能够应用自然语言处理算法，上述问题都是系统成功的关键要素。

本章首先介绍人工智能可解释性基本概念和主要研究内容，在此基础上通用的解释性分析方法，最后介绍可解释自然语言处理中的可解释模型、可解释数据和可解释评估问题。

14.1 可解释性概述

可解释性（Interpretability）问题在统计机器学习模型中广泛存在，在追求更好性能的同时，需要模型更加透明。例如，在智能诊疗问答过程中，为了提供更可靠的服务，模型除了准确寻找患者问题的答案外，同时也应提供机器抽取答案的过程，从而来解释预测行为。杨强教授等人在《可解释人工智能导论》^[1] 中，将可解释人工智能（Explainable Artificial Intelligence, XAI）定义为智能体以一种可解释、可理解、人机互动的方式，与人工智能系统的使用者、受影响者、决策者、开发者等，达成清晰有效的交流沟通以取得人类信任，同时满足各类应用场景对智能体决策机制的监管要求。这对人工智能系统的可解释性提供了更高、更全面的要求。当前，尽管大规模复杂模型已经广泛应用到自然语言处理的各个方向中，也深入影响到了我们生活的方方面面，但是由于其内在决策过程的不可知性，导致在关键业务场景下应用仍然受限，人们无法信任模型的预测结果。如图14.1所示，BERT 在针对该例句的掩盖单词预测任务中虽然给出了合理的预测结果，但是其所依赖的依据并不完全符合人类认知。

从这个例子中也可以看到，虽然目前自然语言处理算法在很多任务上都取得了很好的效果，但

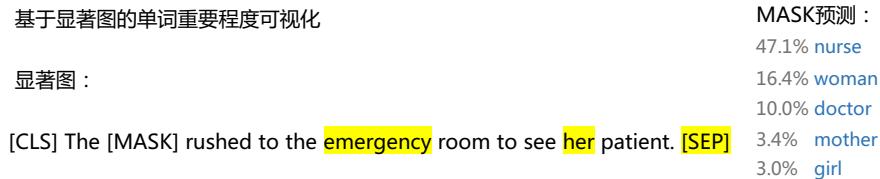


图 14.1 BERT 模型预测依据示例

是仍然需要了解模型的决策依据，这也是优化模型效果并提升对人们对模型信任度的重要方法。由于目前绝大多数自然语言处理算法基于统计机器学习方法，因此算法也离不开数据、模型和评估准则的影响。在数据层面，由于数据样本普遍存在局限和偏见（Bias），仅依赖数据驱动的方法很容易学习到表层模式（Surface Pattern），从不可避免地构建虚假关系（Spurious Relationship）。这些表层模式和关联关系与人们所做决策的依据不同，并不能构建真正的因果关联关系，因此在处理与训练样本不一致的情况时就会产生错误。在模型层面，统计机器学习模型的性能与可解释性之间往往不可兼得。线性模型刻画自变量和因变量之间的线性关系，模型简单易理解，但往往性能欠佳；而深度神经网络模型能刻画自变量和因变量之间复杂的关系，可以拥有更高的性能，但牺牲了可解释性。在评估准则方面，利用标准评测集合使用准确率、精确度等单一指标评价模型效果的方法，虽然推动了自然语言处理的发展，但是缺乏针对模型细粒度和可解释的评价。这些问题都与可解释性息息相关。模型可解释性研究对于发展未来自然语言处理的发展极为重要。

14.1.1 可解释的分类

根据可解释人工智能的定义，可以看到其核心要素是智能体（AI agent）能够有效地“解释”自己，并取得人类使用者的“信任”。解释是信任的基础，随着人工智能系统越来越复杂，功能越来越强大，系统如果要取得人们的信任，就必须要考虑不同用户的应用场景、背景、教育程度等各种因素，提供不同内容与形式的解释。根据系统提供解释的程度以及所面向的受众都可以将人工智能系统的进行分类。遵循《可解释人工智能导论》^[1] 的分类体系，根据系统受众的不同，可解释性可以分为以下几类：

- (1) 面向开发者的解释：系统开发人员具有相当的人工智能专业知识，需要依据解释来进一步提升模型性能和鲁棒性，消除偏差，减少模型风险和错误。比如，模型在处理哪些类型的数据时错误率会明显升高？深度模型的每层或者每个维度的具体功能是什么？
- (2) 面向使用者的解释：系统使用者通常不具备人工智能专业知识，更关心的是系统所做出的某个决策的依据是什么。比如，针对疾病诊断系统，医生希望知道系统所给出的判断主要依据是什么？置信度是如何评估的？
- (3) 面向监管者的解释：随着各国对人工智能系统的应用风险预防的加强以及监管立法逐渐加强，人工智能系统要在监管合规条件下运行。比如，模型的训练过程中所使用的数据是否符合隐私保护及数据治理条例^[2]，需要有明确解释及认证。

不同类型的用户所关注的角度不同，但是总来说主要包含透明度（Transparency）、可解构性（Decomposability）、事后解释（Post-hoc Explanation）、可担责性（Accountability）以及适用边界（Applicable border）。

算法透明度仍然是存在一定争议，但是总体上包括包括算法源代码、输入数据、输出结果等在内的算法要素，综合使用算法分析、算法审计等手段合理促成算法透明。2019年我国发布的《新一代人工智能治理原则——发展负责任的人工智能》、2022年欧盟通过的《数字服务法案》、美国国防部发表的《情报部门人工智能伦理框架》等都对算法透明度进行了一定的要求和规范。

算法可解构性是指可以基于该算法本身内部结构提取算法决策机制构建解释，揭示不同特征在算法决策过程中的作用。线性规划、决策树、朴素贝叶斯等算法具有很好的可解构性，可以根据其模型参数和结构清晰地解释其决策过程，甚至可以对其参数进行人工设定。但是，目前能取得很好效果的深度学习算法却无法解释其预测值，其算法可解构性很差。

算法事后解释试图通过可解释替代模型（Surrogate Model）、基于梯度的相关性、沙普利值等方法提供局部或者全局方法，近似解释黑盒算法的决策依据。相比于白盒算法本身所具备的可解构性，事后解释的方法所提供的是针对黑盒模型的近似解释，方法本身也有很多需要进一步研究的内容。

算法适用边界是指算法所适用的领域和范围，目标是以算法的可解释性为基础，确定算法对于特定问题的适用性。算法适用边界的研究可以在一定程度上减少实际应用中，由于无法被数学模型充分地表示的因素所带来的决策错误和应用风险。

算法可担责性在模型解释性的基础上提出了更高的要求，要求模型提供预测结果的正当性。可担责性要求算法能够“谨慎”地做出预测，试图避免目前基于数据驱动的算法由于训练数据、模型结构等原因，所造成的算法偏见和算法不可控等潜在风险。

14.1.2 解释的评价

近年来，针对黑盒机器学习模型，特别是深度神经网络模型的可解释性，研究人员们从多个方面给出了很多方法。对于这些解释方法如何进行评价，也是仍需进一步研究的问题。解释方法的评价可以从以下几个方面开展：

1. 忠实性

忠实性（Fidelity）是指解释方法是否能客观忠实地反映了被解释算法的处理逻辑^{[3][4]}。如果通过解释方法给出某个特征或变量具有重要的作用，而这个特征或变量确实非常重要，那么这个解释方法就具有很好的忠实性。忠实性是解释方法评价中最重要的指标之一。只有具有良好忠实性的解释方法，才能够真正应用于算法解释和评测中。

2. 敏感度

敏感度（Sensitivity）是指解释方法在输入样本或者模型参数发生微小变化时，所提供的结果是否会发生相应的变化^{[5][6]}。通常情况下，希望解释方法对模型参数的敏感度相对较高，而对输入样

本的敏感度可以适当降低。这样的解释方法与模型参数的相关度较高，同时又能够在输入样本有一定噪声的情况下也能够给出可靠的解释。

3. 全面性

全面性（Integrity）是指解释方法所提供的结果是否完全地反映了被解释算法的全部处理逻辑^[7]。如果一个解释方法所提供的结果仅对某一个部分进行了解释，那么这个解释方法就是不全面的。如果一个解释方法能够对整体和各组块都给出解释，那么该算法的全面性就很好。

4. 可读性

可读性（Readability）是指解释方法所给出的解释是否通俗易懂，便于用户理解。解释方法所提出的结果需要提供给各类型角色用于模型效果提升、结果采纳判定等任务。这些都要求解释方法输出的结果简单，让人容易理解。如果解释方法提供的仍然是数亿维度的数值，或者是包含非常多复杂概念和各种关联关系的解释，人们很难理解，也就不能达到解释的要求。

不同的解释方法所提供结果在上述评价因素上具有不同的权衡。例如模型所有的参数可作为模型行为的一个解释，但这种解释虽然具有很好的忠实性，但是可读性确非常差。又比如，注意力机制可以给出当前单词对句子中每个词的关注程度，可视化编码器在计算单词的最终表示时关注的单词，可以揭示网络如何做出决定。如图 14.2 所示，对于左图和右图中的两个句子，通过对 Transformer 结构中的注意力进行可视化，可以看到“it”可以指代的两个名词，并且各自的关注度反映了它在不同上下文中的选择。这种解释方法具有较好的可读性，但是如果对通过该方法确定的重要词语进行替换，模型预测结果可能并不发生变化，说明该方法的忠实性有所欠缺。

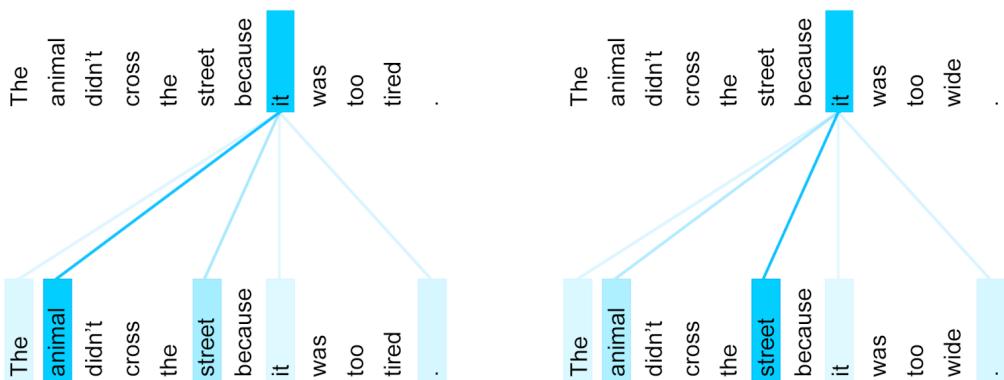


图 14.2 机器翻译任务中，在编码器的 Transformer 层中单词“it”的自注意力分布（八个注意力头之一）。颜色越深，表明注意力分数越高。

本章首先介绍目前可解释机器学习中常用的分析方法，包括局部分析方法与全局分析方法。在

此基础上，从可解释模型、可解释数据以及可解释评估三个方向介绍可解释自然语言处理模型。

14.2 解释性分析方法

根据所关注的视野不同，解释性分析方法可以分为局部解释和全局解释两个类别。局部解释通常是针对单个或一类测试样例，帮助人们判断模型对该样本做出预测背后的原因是否合理、或者挖掘在该样本特征空间的邻域内可能存在的偏差（bias）；全局解释则是针对模型的整体行为，判断模型是否对某些样本存在全局偏差、或者从整体上判断该模型是否可以在现实场景中部署。本节将分别介绍这两类解释性分析方法。

14.2.1 局部分析方法

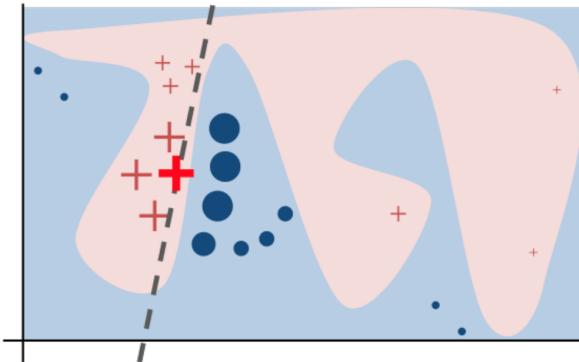
当模型用于为单个样本制定决策时，如何确定对模型对当前样本预测的可信度是非常重要的研究内容。比如，在使用模型进行医疗诊断或者恐怖主义检测时，错误的模型预测结果可能灾难性的后果。在这种情况下，即使模型的整体预测性能为 99.9%，我们也需要尽可能的确定对当前单个样本的预测属于分类正确的 99.9%、还是分类错误的那 0.1%。再比如，利用模型来确定是否给某个申请人批准贷款时，模型只给出拒绝的决策而不提供拒绝的理由，会极大的影响人们的使用体验。而局部分析方法通过对当前样本的预测提供解释，促进模型的使用者和开发者对单个预测的理解，提高人们对当前模型预测的信任。

1. LIME 局部分析算法

LIME (Local Interpretable Model-Agnostic Explanations)^[8] 是一种模型无关的局部分析方法，试图通过学习一个简单的模型来近似原模型在测试样例附近的预测行为，采用一种较为忠实的方式解释分类器或者回归模型的预测。图 14.3 给出了一个二分类问题（蓝色和粉色区域）的示意图。如图所示，尽管模型整体的决策面是非常复杂的，但模型在单个样本（图中粗体红叉点表示）附近的决策面可以使用线性分界面（图中黑色虚线表示）来逼近。因此可以利用简单的线性模型来模拟原始模型在单个样例附近的决策。通过扰动输入样本的特征，来判断哪些特征的存在与否会对模型的决策产生重大影响。例如，删去图片中的某个像素块后模型的性能大幅下降，则说明该像素块是重要的特征。

给定需要解释的分类器 f ，输入 x 以及预测为某个类别的概率，算法的过程可以大致分为以下几个步骤：首先，引入可解释的特征。需要将 x 转化为对应的特征向量 x' 。若样本本身是结构化数据，输入本身是具有含义的，则只需要采样获取扰动的样本；而对于非结构化数据，则需要先引入可解释的特征。对于图片而言，可以利用超像素（super pixel）的方式做图片分割，将图片切分成若干块，用二分向量来指示某个超像素是否存在，对于文本数据则利用单词是否存在作为二分向量的指示。

其次，获得原始 x 的扰动样本。在预测样本的邻域内随机采样，对于连续型特征，根据正态分布来采样随机数产生扰动样本；对于类别型特征，则根据训练集的分布进行采样，若与测试样

图 14.3 LIME 模型结果示例^[8]

本特征相同则为 1，否则为 0。对特征向量 x' 进行扰动后获得对应的 z 和 z' ，并计算扰动样本 z 与 x 的距离 $D(x, z)$ 。

最后，则是训练解释模型 g ，其优化目标是最小化 g 和 f 在扰动的样本上的预测，即令 g 尽可能地逼近 f 在样本 x 附近的决策行为，同时保持 g 尽可能的简单。此外，考虑到扰动的样本可能和 x 偏离很远，模型 f 对偏离较远的样本的决策面可能不再是线性的，因此 LIME 算法加入了对距离的惩罚，使得模型可以更关注和 x 更近的样本，其目标函数为：

$$\mathcal{L}(f, g, \pi_x) = \sum_{z', z \in Z} \pi_x(z)(f(z) - g(z'))^2 \quad (14.1)$$

其中， g 使用的是 K-Lasso 回归模型， $w = \min_{w_0, w} \left\{ \frac{1}{N} (y - w_0 - Xw)^2 \right\}$ ，且 $\sum_{j=1}^p |w_j| \leq K$ 。 K 为选择的特征数量。 $\pi_x(z)$ 是一个指数核函数，刻画了 z 和 x 的相似程度（在图 14.3 中距离越小、越相似、点越大）， $\pi_x(z) = \exp(-D(x, z)^2/\sigma^2)$ ， σ 为超参。最终根据模型 g 学习到的参数 w 的值的大小就可以知道对应的超像素点或者单词的重要性。

LIME 通过扰动特征根据预测的变化来判断特征的重要性，同时最后可以获取对应特征的重要值，因此是一种非全局忠实，但是局部较为忠实，同时可读性也较好的方法。并且，由于 LIME 是一种模型无关的算法，其适用性也相对广泛。但是，LIME 的使用上也存在一些问题：1) 需要确定邻域的范围，对于不同的邻域，产生的解释可能不同甚至相悖；2) 对结构化数据的扰动特征采样时，可能会忽略特征之间的相关性，导致产生一些不合常理的样本来解释模型；3) 解释模型 g 需要预先设定，不同的解释模型产生的解释也可能不同。而这些缺点也导致了 LIME 方法本身不太稳定的缺点。

2. 显著图局部分析算法

显著图（Saliency Map），也称为热力图，计算了单个输入的各个部分与模型预测结果的相关

性程度，相关性分数越高的部分对模型输出的重要性程度也越高。基于显著图的可视化结果，人们能直观地在视觉上将模型输出归因到输入样本的某些部分。与 LIME 相比，显著图的可解释性不需要新增解释模型学到的参数，它利用原始模型的单个输入与参数通过设计好的公式计算得到。因此，显著图拥有良好的可读性，但间接基于模型参数的获取方式也在一定程度上降低了它的忠实性。

获取显著图的方式主要包含以下类型：

- 基于注意力：将模型中注意力模块的注意力分数，转化为显著图中的相关性分数（本章第14.3.1节将详细介绍相关方法）。优点是方法简单直接，可读性高；缺点则是依赖于注意力模块，并不适用于所有模型，且忠实性难以保证。
- 基于扰动：通过扰动单个输入或神经元后，观察对网络中后续神经元产生的影响^[9]。优点是能够直接观察到某些输入部分对特定神经元或输出的影响；缺点则是计算效率低，因为每次扰动都需要单独的一遍经整个网络的前向传播。
- 基于反向传播：通过一次反向传播，将重要性信号从输出神经元传播至输入神经元。与基于扰动的方法不同，仅需一次传播即可生成显著图的方式保证了该方法的高效快速，但同时使用不同的反向传播方式，生成显著图的方法又存在不同的问题。

因为模型的训练方式大多采用梯度的反向传播，所以基于反向传播的显著图生成方式和其他方法相比，除了高效快速，忠实性也更高。因此许多工作选择对这种方法进行精细设计以获取质量更高的显著图。

基于反向传播的显著图获取方式中，最经典的做法是基于梯度的，首先应用于图像分类任务中。使用模型输出类的预测得分对输入图像各个像素的梯度值表示该输入与对应预测类别的相关性程度^[10]，计算公式如下所示：

$$G_i(x) = \nabla_x F_i(x) \quad (14.2)$$

其中 x 是模型输入的一幅图像， $F_i(x)$ 是模型将 x 归为类别 i 的预测得分， $G_i(x)$ 则表示该模型输入 x 对输出类别 i 的显著图，大小与 x 一致。图14.4给出了使用 ConvNet 神经网网络，通过 ILSVRC-2013 数据集进行训练，针对模型输出得分最高的类别给出的显著图示例。通过对比输入图片和所对应的显著图，我们可以看到模型分类所依赖的信息是否与人的认知一致，从而可以分析和改进模型结构。

为了解决直接使用梯度所产生的梯度饱和等问题，许多工作在此基础上提出了改进。其中一类直接修改显著图的计算方式。SmoothGrad^[11] 提出对输入加上随机的高斯噪声，以减少基于梯度的显著图中的视觉噪声，生成更平滑的显著图。具体做法是对特定的输入图像随机采样多个高斯噪声以生成多个模型输入，然后平均生成的显著图，计算方式如下：

$$G_i(x) = \frac{1}{N} \sum_i^N \nabla_{(x+\epsilon)} F_i(x + \epsilon) \quad (14.3)$$

图 14.4 基于梯度的显著图生成方法效果^[10]

其中 N 是随机采样的样本数, $\epsilon \sim \mathcal{N}(0, \sigma^2)$ 表示高斯噪声。尽管该方法可以生成视觉上更清晰的显著图, 但梯度饱和的问题并未解决。

集成梯度 (Integrated Gradients, IG) ^[12] 方法则对输入进行线性插值, 然后将其梯度沿直线进行积分。它将输入从基础值到当前值的梯度积分看作相关性得分, 公式如下:

$$G_i(x) = (x - \tilde{x}) \times \int_{\alpha=0}^1 \frac{\partial F(\tilde{x} + \alpha \times (x - \tilde{x}))}{\partial x} \quad (14.4)$$

其中 \tilde{x} 表示 x 的基础值。因为积分梯度的计算与输入相关, 所以它可以解决当输入到达某些值之后造成的梯度饱和的问题。基于集成梯度的显著图方法也应用于多模态问题回答模型分析^[13]。图14.5给出了一个分析的示例, 红色的单词表示对问题回答有正面贡献, 蓝色的单词表示对问题回答有负面贡献, 灰色的字表示对问题回答基本没有贡献。

图 14.5 基于集成梯度显著图方法在 VQA 任务分析结果示例^[13]

从这个示例中, 可以看到虽然模型对该问题给出了正确的答案, 但是模型所依赖的分类依据是“how”、“are”等这种对于问题语义表达并不重要的词语。相反, “white”、“either”等对语义有重要影响的词语还对本问题的正确回答起到了负面作用。文献 [13] 中还对具有非重要作用的单词进行

了替换，发现这些非重要单词替换之后确实并不影响分类结果。例如，将问句替换为“how spherical are the white bricks on either side of the building”，“how soon are the bricks fading on either side of the building”等句子后，模型所给出的结果依然是“very”。

3. 沙普利值局部分析算法

沙普利值（Shapley Value）是一种来自于联合博弈论的方法，用于根据玩家对总支出的贡献来分配支出^[14]。在机器学习中，则是将不同的特征对最终预测的总贡献分配到各个特征上。沙普利值可以看做边际效益的均值，比如，当 A 单独工作时产生效益 $v(A)$ ，B 加入后则收益变成 $v(A, B)$ ，那么 A 的边际效益则为 $v(A, B) - v(A)$ 。A 的沙普利值就是在所有可能的工作排列组合中边际效益的加权求和。

给定模型 f ，要计算特征 i 对模型的贡献 ϕ_i ， F 为模型 f 中所使用的全部特征集合，计算公式如下：

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)] \quad (14.5)$$

其中， S 是模型中使用的特征的子集， x 是要解释的实例， $f_S(x_S)$ 则表示利用特征子集 S 训练的模型 f_S 对使用相同特征子集的实例 x_S 的预测。因此该算法需要对不同的特征子集都分别训练一个模型，所以只能应用于小数据、小模型。

沙普利值是唯一满足有效性、对称性、冗余性和可加性的归因方法：有效性指的是各个特征贡献值之和等于总贡献；对称性指的是如果两个特征对所有可能的特征集合贡献都相同，那这两个特征的沙普利值相同；冗余性指的是如果一个特征不管加到任意特征集中产生的贡献都为 0，那么它的沙普利值为 0；可加性指的是某个特征的总贡献值是多个特征组合的累计贡献。但由于沙普利值需要计算总特征 F 的所有子集，而当特征的数量增加时，特征集 S 的数量会随之指数增长。因此，文献 [15] 等方法大多使用蒙特卡洛采样的方法来近似计算。

沙普利值通过枚举特征的所有排列组合，来计算特征在所有排列组合上收益的加权平均值。相较于 LIME 通过采样来获取特征的排列组合，采样的数量并不能保证公平性。沙普利值的优点在于可以公平的将贡献分到特征值上，因此沙普利值的忠实力相较于 LIME 更好，最后的输出和 LIME 一样也是所有特征的重要值，因此可读性和 LIME 一样较好。但是沙普利值的缺点在于计算速度很慢，需要一些近似方法来加速计算，这样也会损失一部分公平性。除此之外，和 LIME 不同的一点是，沙普利值需要所有的训练数据来计算每个特征的重要性，这也在一定程度上限制了模型的可用范围，而 LIME 不存在这个问题。

4. 神经元激活局部分析算法

激活最大化（Activation Maximization）目标是获得一个可以最大化某些神经元激活值的输入。在正常的神经网络训练过程中，通过反复调整网络的权重，从而最小化神经网络在训练集上的损失。而激活最大化是在神经网络训练完成之后，在固定神经网络参数的条件下，通过基于梯度的

方法优化输入，使某一个神经元的激活值最大^[16]。

假设一个已经训练好的分类器 θ ，可以将输入 x 映射到一个多类别的概率分布上。激活最大化方法的目标就是寻找一个能够最大化该分类器网络第 l 层、第 i 个神经元激活值的输入 x^* ，可以形式化表示为：

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmax}} \{a_i^l(\theta, \mathbf{x})\} \quad (14.6)$$

其中， a_i^l 是一个单独的神经元的激活值，但也可以扩展成一组神经元的激活值，也就是说希望找到一个输入 x^* ，其可以最大化一组神经元的激活值。通过激活最大化获得的输入 x^* 被认为是针对神经网络中的某一小部分神经元的解释。通过分析 x^* ，可以得知这一小部分神经元对什么输入内容更为敏感。为了简化表示，本节以下部分采用使用 $a(\cdot)$ 代替 $a_i^l(\cdot)$ 。

最大化激活是一个非凸优化问题，可以通过基于梯度的方法找到一个局部最优点。在优化过程中，神经网络模型的参数是已知的，可以通过梯度上升更新输入：

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \gamma \frac{\partial a(\theta, \mathbf{x}_t)}{\partial \mathbf{x}_t} \quad (14.7)$$

其中， \mathbf{x}_0 是一个随机初始化的起始输入，通过不断的进行迭代，期望在输入空间中找到一个能够使目标神经元激活值最大的输入 x^* ， γ 是根据经验选择的学习率。这个过程中神经网络的参数是固定的。对输入的优化过程通常在到达一个合适的阈值或者一定的步数后停止。图14.6给出了基于激活最大化方法，使用 ConvNet 神经网网络，通过 ILSVRC-2013 数据集进行训练后，在分类层“washing machine”、“computer keyboard”以及“kit fox”所对应的神经元所对应的激活最大化输入。

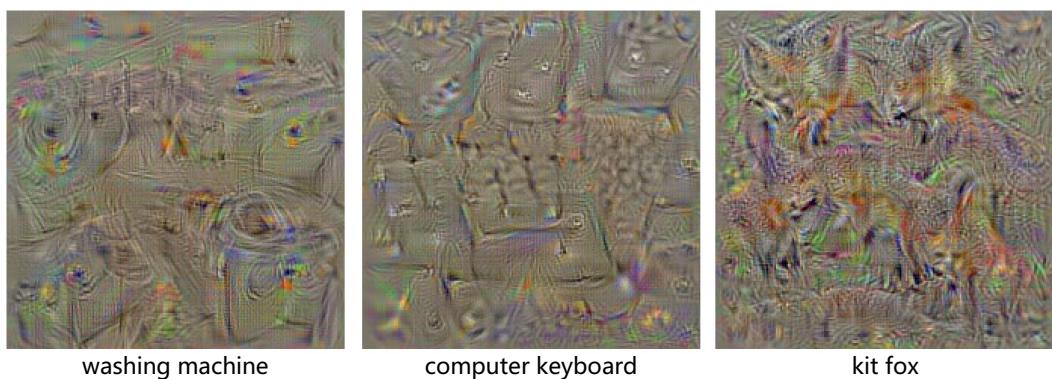


图 14.6 基于激活最大化方法示例^[17]

直接从完全随机的输入出发，在没有任何约束的情况下通过最大化激活值的方法进行优化，得到的结果往往难以理解。如图14.6所示，虽然在相关输入中能够看到一定的类别图像特征，但是还

是很难让人理解。因此，可以对搜索空间进行一定程度的限制。例如，可以从一张真实的图片出发，以使得得到的结果和真实的图像或者训练集中的图片比较相像，进而有较强的可解释性。也可以在目标函数中加入自然图片的一些先验特征来限制搜索范围，改善最大化激活图像的可识别性^{[18][19]}。比如，为了增强激活最大化图像的光滑程度，可以定义一个函数 R 计算图片的总变差 (Total Variation)。然后，沿着同时满足最大化神经元激活和最小化总变差损失两个条件的梯度方向更新。

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \gamma_1 \frac{\partial a(\theta, \mathbf{x}_t)}{\partial \mathbf{x}_t} - \gamma_2 \frac{\partial R(\mathbf{x}_t)}{\partial \mathbf{x}_t} \quad (14.8)$$

根据先验函数 R 的选择不同，最大激活图像会有不同的特点。如图14.7所示，使用 Jitter 函数作为正则化项与不使用正则化项之间还是存在一定的区别，引入正则化项可以更好的进行解释和理解^[18]。

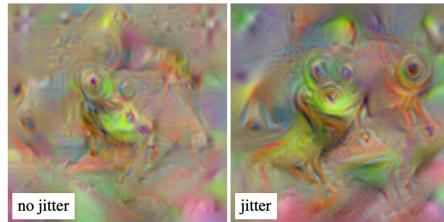


图 14.7 使用和不使用 Jitter 正则项得到的激活最大化图像对比示例^[18]

14.2.2 全局分析方法

局部分析方法可以帮助我们理解模型的单个预测。但是，除了单个预测之外，在模型真正应用到真实场景中前还需要对模型进行整体的评估。全局分析方法则可以从全局角度上提供对模型的解释。对拥有大规模训练集合的模型而言，通过局部分析方法逐个检查模型对训练和测试数据预测是否合理，在时间和成本上通常是不可接受的。全局分析方法可以通过建议检查特定样例，大大缩小需要检查的数据范围。

1. SP-LIME 全局分析法

SP-LIME^[8] 是基于 LIME 的一种全局分析方法。LIME 是在局部找到对当前预测影响较大的特征，从而提供单个预测的解释，而 SP-LIME 则是通过选取一组具有代表性且多元化的实例来表示模型的整体行为。SP-LIME 将这个选取样例的问题转化为次模优化 (Submodular Optimization) 问题。次模 (Submodular) 是经济学上边际效益递减的形式化描述，即往集合 A 中增加一个元素的增益要小于等于往 A 的子集中增加一个元素的增益。因此，可以借鉴次模的想法不断的往集合中添加增益最大的元素，来寻找最具代表性的集合。

SP-LIME 首先需要根据 LIME 算法得到 n 样本对应的特征的重要性，从而得到得到一个 $n \times d$

的重要性矩阵 \mathbf{W} , 其中 n 表示样本的数量, d 表示特征的数量。如图 14.8 所示, 重要性矩阵 \mathbf{W} , \mathbf{W}_{ij} 表明第 i 个样本的第 j 个特征的重要性。图中将 \mathbf{W} 简化为二元矩阵。为了选取有代表性且多元化的样本, 需要选择在覆盖尽可能多特征且彼此重合小的样本, 例如, 第 2 个样本和第 3 个样本具有相同的特征值, 则只需要选取一个样本。对重要性矩阵 \mathbf{W} 的每一列求和得到解释空间中不同特征的全局重要性 $I_j = \sqrt{\sum_{i=1}^n \mathbf{W}_{ij}}$ 。在图 14.8 中, 特征 f2 覆盖的样本数最多, 因此 I_2 最大。

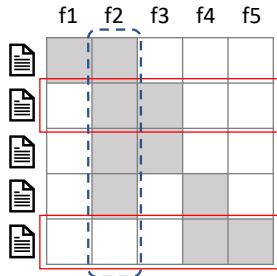


图 14.8 SP-LIME 中使用的 $n \times d$ 的矩阵^[8]

选取样例的标准是尽可能的覆盖所有的特征, 因此可以用贪心的方法来选择样例。首先初始化样例选择集合 V 为空集, 然后不断地往 V 中添加使集合 V 的覆盖率提高最大的样本点。集合的覆盖率定义为:

$$c(V, \mathbf{W}, I) = \sum_{j=1}^d \mathbf{1}_{[\exists i \in V : \mathbf{W}_{ij} > 0]} I_j \quad (14.9)$$

即集合 V 覆盖的特征的个数。当集合 V 的大小达到预设的挑选数量则停止添加。上述算法迭代地增加有最高边际覆盖增益的样本 i , 并以常数 $1 - \frac{1}{e}$ 的速度近似到最优。集合 V 中的样本就是所选取具有代表性的实例。

2. 模型蒸馏全局分析法

现在神经网络模型通常通过引入大量参数提升模型预测性能, 在提升性能的同时, 参数数量的提升也为模型的行分析和解释带来了很大的挑战。如果能将大模型学到的知识通过某种方式转移到一个相对简单的、更可解释的小模型中, 就可以认为小模型可以在一定程度上反应大模型的决策过程, 通过对小模型进行解释性分析, 进而得到在大模型上全局解释。

模型蒸馏是就一种将大模型的知识迁移到小模型中的常见技术。虽然大模型往往拥有非常大量的可学习参数, 需要更大的存储空间和更长的推理时间。但是, 与此同时很多参数并没有得到充分的利用。如果能将大模型中的知识迁移到到小模型中, 那么可以约束参数数量对解释性的影响, 同时最大限度的保留大模型的性能优势。这也是蒸馏一词的来源, 意味着去除大模型中的“杂质”。在知识蒸馏中, 大模型被形象地称为教师网络, 小模型被称为学生网络, 学生网络被要求去拟合教师网络的输出。

Bastani 等人在文献 [20] 中提出使用高度结构化的、更容易进行解释性分析的决策树来近似一个黑盒模型，将得到的决策树作为黑盒模型的全局解释代理。该方法使用坐标轴对齐 (Axis-aligned) 的决策树，树中的非叶子结点都包含一个坐标轴对齐条件 $C = (x_i < t)$ ，其中 $i \in [d] = 1, \dots, d$, $t \in \mathbf{R}$, d 是输入空间 \mathcal{X} 的维度，记条件 C 的可行集为 $F(C) = \{x \in X | x \text{ 满足 } C\}$ 。决策树 T 是一个二叉树，其中一个内部节点 $N = (N_L, N_R, C)$ 拥有左节点 N_L 和右节点 N_R ，以及一个条件 $C = (x_i < t)$ 。叶子节点 $N = (y)$ 则和某个标签 $y \in \mathcal{Y}$ 绑定。记 N_T 为 T 的根结点。对于一个节点 $N \in T$ ，记 C_N 为根结点 T 到节点 N 路径上的条件的交集。

对于一个训练集 $X_{train} \subseteq \mathcal{X}$ 和一个黑盒模型 $f : \mathcal{X} \rightarrow \mathcal{Y}$ ，该方法的目标是学习一个决策树 $T : \mathcal{X} \rightarrow \mathcal{Y}$ 去近似 f 。首先使用 X_{train} 估计 \mathcal{X} 的分布 \mathcal{P} ，然后贪心的构造决策树 T : T 初始化为一个根结点，然后不断迭代分割其叶子结点。当分割叶子结点 $N \in T$ 时，使用动态采样策略得到一个新的输入 $x \sim \mathcal{P}$ ，并且 $x \in F(C_N)$ ，使用黑盒模型 f 计算其对应的标签 $y = f(x)$ ，并用这些数据验证划分的好坏。

首先使用 EM 算法得到一个拟合 X_{train} 的混合坐标轴对齐的高斯分布 \mathcal{P} 。用类似 CART^[21] 的方法，构造一棵大小为 k 贪心决策树 T^* 。初始化 T^* 为单节点 $N_{T^*} = (y)$ 树， y 是分布 \mathcal{P} 中的出现次数最多的标签。然后进行 $k - 1$ 次迭代划分 T^* 中的叶子结点：在每次迭代时，我们选择一个叶子结点 $N = (y)$ ，然后用一个内部节点 $N' = (N_L, N_R, C)$ 替代它，其中 $N_L = (y_L)$ 、 $N_R = (y_R)$ ， $C = (x_{i^*} \leq t^*)$ ：

$$(i^*, t^*) = \underset{i \in [d], t \in \mathbf{R}}{\operatorname{argmax}} G(i, t)$$

其中划分的收益 G 使用基尼杂质 (Gini Impurity) H 表示。

$$\begin{aligned} G(i, t) = & -H(f, C_N \wedge (x_i \leq t)) \\ & -H(f, C_N \wedge (x_i > t)) + H(f, C_N) \end{aligned}$$

$$H(f, C) = \left(1 - \sum_{y \in \mathcal{Y}} \Pr_{x \sim \mathcal{P}}[f(x) = y \mid C]^2 \right) \cdot \Pr_{x \sim \mathcal{P}}[C]$$

划分之后，叶子节点的标签为：

$$\begin{aligned} y_L &= \arg \max_{y \in \mathcal{Y}} \Pr_{x \sim \mathcal{P}}[f(x) = y \mid C_N \wedge (x_i \leq t)] \\ y_R &= \arg \max_{y \in \mathcal{Y}} \Pr_{x \sim \mathcal{P}}[f(x) = y \mid C_N \wedge (x_i > t)]. \end{aligned}$$

文献 [20] 中通过问卷调查的方式进行评估抽取出来的决策树的准确性和可解释性。在一个糖尿病相关的数据集上，作者征集了 46 名有机器学习相关背景的本科生，让他们以抽取出来的决策树为依据，完成一个糖尿病诊断相关的问卷，用这些志愿者的得分来衡量模型的可解释性。

14.3 自然语言处理算法解释分析方法

前一节介绍的通用可解释性算法应用到具体的自然语言处理任务中时，还需要针对具体任务的特点来调整。通用方法往往假设模型的输入空间或者隐空间是连续的（例如：数字图像信号、音频信号），但是自然语言处理任务处理的大都是单词，模型往往需要处理离散信号。因此在运用可解释算法时需要考虑模型不可微、搜索空间大等挑战。另一方面，针对自然语言处理任务本身的特点，研究人员也探索了许多聚焦于文本任务的可解释性方法。本节将从模型解释性分析算法，数据解释分析方法，以及可解释评估三个角度介绍一些自然语言处理任务中常用的解释性分析方法。

14.3.1 模型解释性分析算法

通用的解释性分析方法通过稍加改造大都可以应用于自然语言处理模型，主要需要处理的问题在于，通用解释分析算法通常是计算输入的每个维度的重要程度。但是自然语言处理任务的输入是由若干个单词组成，每个单词由包含若干维度的向量表示，因此如何从对每个维度重要性的解释转换为单词级别是需要研究的问题。此外，注意力机制以及探针任务是针对自然语言处理领域算法特性而设计的可解释模型。本节将从上述三个方面分别介绍针对自然语言处理算法的可解释模型。

1. 显著图分析方法

本章第 14.2.1 节介绍了显著图分析方法，可以通过基于梯度、传播或者遮挡的方法来衡量神经网络中特定单元或者输入数据特定维度的重要性。如文献 [22] 给出了基于一阶导数的显著图方法，衡量输入中每个单元对最终决策的贡献，采用一阶导数来近似。假设对于一个分类任务，输入 E 所对应的正确分类结果为 c ， $S_c(E)$ 表示模型针对输入 E 在类别 c 上的得分。显著图分析方法的目标是获取输入中的每个单元对于最终分类结果 c 的得分 $S_c(e)$ 的贡献进行评价。

对于 E 施加微小噪音可以得到 e ，同样可以通过模型得到 $S_c(e)$ ，由于在深度网络条件下 $S_c(e)$ 是高度非线性函数，可以采用一阶泰勒展开进行近似，从而可以将其转换为线性表示：

$$S_c(e) \approx w(e)^T + b \quad (14.10)$$

其中 $w(e)$ 表示 S_c 关于输入 e 的导数：

$$w(e) = \frac{\partial(S_c)}{\partial e} |_e \quad (14.11)$$

导数的绝对值表示某一特定维度对最终决策的贡献度大小，因此显著性得分 $S(e)$ 定义为：

$$S(e) = |w(e)| \quad (14.12)$$

图14.9给出了使用一阶导数显著图方法的分析示例。模型采用句子级情感倾向分析语料进行

训练，对于语句“我喜欢这部电影”，分类为“褒义”类别的显著图进行了可视化。每个单词的嵌入表示由 100 维向量表示，每个维度的显著性归一化到 0 到 1，颜色由浅至深进行表示。从显著性分析结果上，该句子分类着重依赖了“喜欢”和“电影”两个单词。

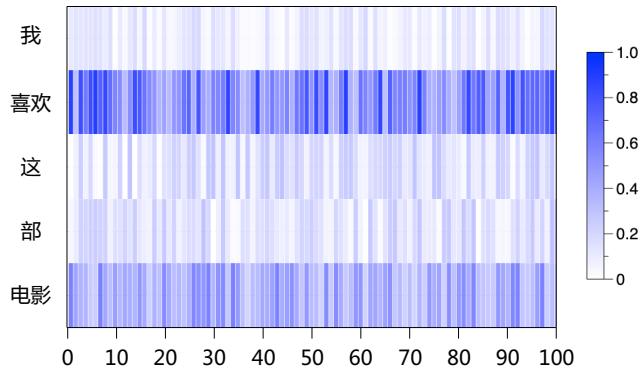


图 14.9 显著图分析方法结果样例

通过一阶导数显著图方法可以得到输入单词每个维度的重要性，但是文本输入是离散的，每个单词嵌入表示由多个维度组成，因此如何通过每个维度的重要性确认单词的重要性也是需要研究的问题。常见的方法可以对单词每个维度的重要性求平均、均值、最大值等方法得到单词的显著性值。

2. 注意力分析

注意力机制是基于神经网络自然语言处理模型中重要的一个部件：对于输入文本的不同部分，神经网络赋予它们不同的权重，并基于这些权重构建文本处理任务的预测结果。以自注意力机制为基础的 Transformer 模型，以及基于其构建的大规模预训练模型在成为现阶段自然语言处理任务的标配。因此，对注意力的分析提供了理解自然语言处理模型决策过程的一种重要途径。本节我们将以预训练模型 Transformer 结构（如 BERT^[23], ALBERT^[24] 等）中的注意力为例，简介如何通过注意力分析解释模型的内在运行机理。

预训练模型在大量无标注的语料上进行自监督训练获得文本表示。它们在众多自然语言处理任务中获取了良好的表现，但这些模型的表示学习过程仍是黑盒状态：我们并不清楚在自监督学习的过程中模型学习到了什么类型的知识，以及这些知识如何被使用到具体的下游任务中。为更好的理解预训练模型，需要开发针对它们的可解释性分析工具，而其中重要的一类工具是探究预训练模型的注意力中是否蕴含了与语言学结构向匹配的知识。

文献 [25] 尝试探索预训练语言模型（BERT^[23]）是否学习到了语法结构。它主要通过详细分析 BERT 模型内部注意力分布，观察模型从输入中学到的结构信息。具体来看，为了研究注意力头中包含的语言学信息，将 Transformer 结构中的每个注意力头看作简单的分类器，通过标准评测数

据集观察它们在预测语法结构的效果。以识别依存句法分析中的“nsubj”关系为例介绍算法流程。

对于注意力头 h , 为句子中每个位置 $1 \leq j \leq n$, 寻找最相关的词 $w_{\arg \max_i \alpha(h)_i^j}$, 其中 $\alpha(h)$ 表示注意力头 h 注意力分数分布, $\alpha(w, h)_i^j$ 表示 w_j 与 w_i 之间的注意力权重。在标注数据中对于输入词序列 w_1, \dots, w_n , 若词对 (w_i, w_j) 存在 nsubj 关系, 则记 $l(w_i, w_j) = 1$, 否则为 0。将得到的关系词对集合记为 $S_l(w) = \{j : \sum_{i=1}^n l(w_i, w_j) > 0\}$ 。那么评估注意力头 h 是否学到了 nsubj 关系, 可以统计所有词对 $(w_{\arg \max_i \alpha(h)_i^j}, w_j)$ 在 $S_l(w)$ 中出现的频率, 从而得到其精度:

$$\text{Precision}(h) = \frac{1}{N} \sum_{w \in \text{corpus}} \sum_{j \in S_l(w)} l(w_{\arg \max_i \alpha(h)_i^j}, w_j)$$

其中 N 表示语料库中所有关系词对集合的总词数。

使用这种方法, 文献 [25] 观察到 BERT 的注意力头在 Penn Treebank 依存关系标注数据集和 CoNLL-2012 指代关系数据集上均达到了较高的准确率, 证明了它在自监督过程中确实学到了一些语法结构特征, 如图 14.10 所示。

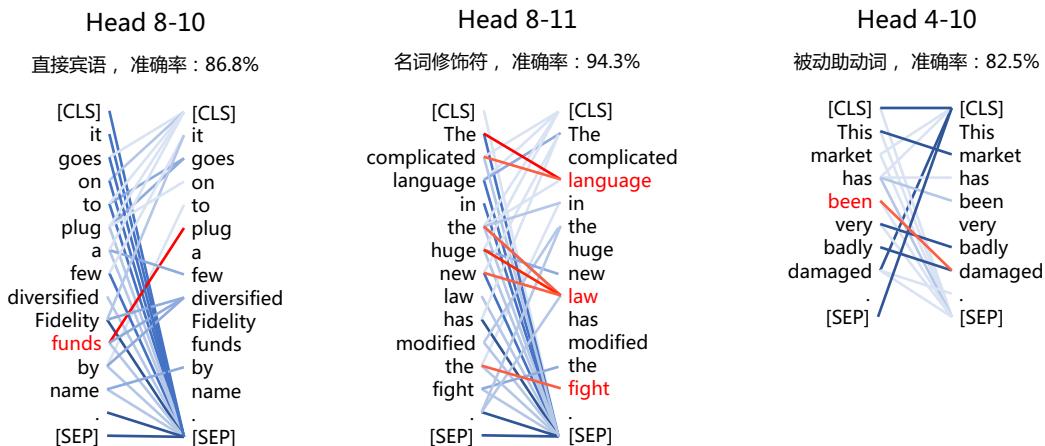


图 14.10 BERT 中不同注意力头在语法结构识别结果示例^[25]

3. 探针任务

设计探针任务 (Probe Tasks) 是一类可以面向隐藏表示的解释性方法。比如, 希望探究预训练语言模型 (如 BERT) 的某隐层表示是否包含词性信息, 可以通过构建“探针分类器”来尝试根据一个词的隐层向量预测该词在句子中的词性。具体构建过程可以通过以下几步完成:

- (1) 选择带有词性标注的数据集 $\mathcal{D} = (\mathbf{x}_i, y_i)_{i=0}^{|\mathcal{D}|}$, 其中 \mathbf{x}_i 为句子, $\mathbf{x}_i = \{w_{i1}, w_{i2}, \dots, w_{i|x|}\}$, y 为词性类别 $y = \{p_1, p_2, \dots, p_{|y|}\}$ 。
- (2) 选定所关心的预训练模型隐层 l

(3) 构建探针分类器的训练(测试)数据集 $\mathcal{D}' = \{(\mathbf{g}_i, y_i)\}$, 其中 $\mathbf{g}_i = \{h_{i1}^{(l)}, h_{i2}^{(l)}, \dots, h_{i|x_i|}^{(l)}\}$, $h_{i*}^{(l)}$ 为输入 x_i 在第 l 层的隐层表示。

(4) 利用构建的数据集合 \mathcal{D}' 训练并测试模型精度

探针模型对于词性的预测准确程度可以作为一种对隐层向量的解释: 若从一个隐层能够很好的预测词性信息, 说明该隐层较好的包含了词性信息。通常情况下, 为了更好的解释隐层(而不是预测词性), 探针分类器通常选择较简单的网络结构(如线性分类器, 单层神经网络等方法)。现有的探针任务主要包括**句子级别**^[26] 和**词级别**^[27] 两大类。

句子级别的探针任务是根据模型上训练得到的句子的向量表示, 来探究模型是否学习到了句子级别的语义信息, 主要包括以下任务:

- 浅层信息的探针任务

- 长度探测 (Sentence Length), 预测句子长度, 该任务于测试句向量 (Sentence Embedding) 是否保留了句子长度的相关信息。
- 词成分探测 (Word Content), 预测一组中频单词是否在句子中出现过, 该任务用于测试句向量是否学习到了单个单词的信息。

- 句法信息的探针任务

- 语序探测 (Bigram Shift), 调换输入中两个单词的位置, 然后对扰动后的输入做二分类判断是否调换过位置, 可以用于探测句向量是否保留了词序信息。
- 句法树深度探测 (Tree Depth), 预测句子语法树的深度, 可以用于探测句向量是否包含了句子的层次结构信息。
- 浅层成分块探测 (Top Constituent), 预测句子的浅层成分(即, 句子成分句法树第二层(S结点的下一层)的语法标签)。将一个句子的浅层成份拼接构成待探测的标签。例如: 一个句子由一个形容词短语 (ADVP)、一个名词短语 (NP)、一个动词短语 (VP) 构成, 则对应的预测标签为“ADVP NP VP”。为减少标签数量, 可以选取出现频率最高的几个做为标签集合。

- 语义信息的探针任务

- 时态探测 (Tense), 预测句子时态。
- 主语单复数探测 (Subject Number), 预测主语的单复数。
- 宾语单复数探测 (Object Number), 预测宾语的单复数。
- 动名替换探测 (Semantic Odd Man Out), 随机替换句子中的动词或名词替换为其他的动词或者名词, 二分类判断是否进行过替换。
- 主从顺序探测 (Coordination Inversion), 随机交换两个并列的分句的前后顺序(如以“and”连接的两个句子), 用于探测句向量是否学习到了语言逻辑相关的信息。

词级别的探针任务作为对句子级别的探针任务的补充, 关注单词级别的语义信息, 主要包括以下任务:

- 词汇语义相似性 (Lexical Semantic Similarity)，评测人工标注的单词对的语义相似度评分和词向量的余弦相似度评分的相关性 (Spearman's Rank Correlation)。
- 词类比 (Word Analogy)，对于单词间的类比关系对 $w_a : w_b = w_c : w_d$ (如：“男人”：“国王”=“女人”：“女王”)，该任务需要在给定 w_a, w_b, w_c 的情况下，预测 $w_a : w_b = w_c : x$ 中的 x 。

探针任务一定程度上解释了模型对于不同语料，不同任务学习到了哪些相关的知识，但是探测任务的准确性高就一定证明了模型学习到了我们所探测的性质吗？也很有可能模型只是去拟合了数据的分布，John 等人提出可以采用控制任务的方法来一定程度上评测探针任务是否有效^[28]。最基础的控制任务就是随机打乱探针任务中的标签后重新进行预测。若打乱后的预测结果依旧很好，那可能就说明了模型拟合了数据，而并不是真正学习了句子的语义表示。

14.3.2 数据解释分析方法

统计机器学习方法不仅依赖网络结构和损失函数等算法结构，训练数据也起到了非常重要的作用，对最终模型结果产生重要影响，因此如何衡量训练语料对于模型预测结果的影响也是可解释性研究中重要研究内容。文献 [29] 针对数据对模型预测结果的影响开展了研究，该论文获得机器学习领域重要国际会议 ICML 2017 (International Conference on Machine Learning) 最佳论文奖。

文献 [29] 提出将训练语料中某个数据对模型某个预测的影响拆解为两个问题：1) 如果将训练语料中的某个样本去掉，重新训练得到的新模型，利用该模型做出的预测，会发生什么样的变化？2) 如果对训练语料中的某个样本进行微小的扰动，重新训练得到新模型的预测结果会有什么样的变化？我们可以形式化地定义上述问题，输入样本空间为 \mathcal{X} ，输出目标空间为 \mathcal{Y} ，给定训练语料集合 $Z = \{z_1, z_2, \dots, z_n\}$ ，其中 $z_i = (x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ 。给定一个样本 z 和模型参数 $\theta \in \Theta$ ，损失函数为 $L(z, \theta)$ ，相应的 $\frac{1}{n} \sum_{i=1}^{n-1} L(z_i, \theta)$ 为经验损失。给定训练样本和损失函数，可以通过经验风险最小化准则训练模型参数，即 $\hat{\theta} \stackrel{\text{def}}{=} \arg \min_{\theta \in \Theta} \sum_{i=1}^{n-1} L(z_i, \theta)$ 。

针对在训练语料中去除某个训练点 z ，模型针对某个测试样本发生变化的问题，可以通过训练包含和不包含 z 的数据集合，得到参数 $\hat{\theta}$ 和 $\hat{\theta}_{-z}$ ，通过参数变化 $\hat{\theta}_{-z} - \hat{\theta}$ 来衡量。但是这种方法需要重新对模型进行训练，对于需要大规模计算的深度学习模型来说，所需要的计算量和时间过多。影响函数 (Influence function) 方法提供了通过对 z 的进行微小加权来近似计算的方法。假设对目标训练数据 z 给于一个非常小的加权 ϵ ，则 $\hat{\theta}_{\epsilon, z} \stackrel{\text{def}}{=} \arg \min_{\theta \in \Theta} \sum_{i=1}^{n-1} L(z_i, \theta) + \epsilon L(z, \theta)$ 。通过文献 [30] 中分析结论，可以得到：

$$\mathcal{I}_{\text{up, params}}(z) \stackrel{\text{def}}{=} \left. \frac{d\hat{\theta}_{\epsilon, z}}{d\epsilon} \right|_{\epsilon=0} = -H_{\hat{\theta}}^{-1} \nabla_{\theta} L(z, \hat{\theta}) \quad (14.13)$$

其中 $H_{\hat{\theta}} \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^n \nabla_{\theta}^2 L(z_i, \hat{\theta})$ 是海森矩阵 (Hessian)。由于当 $\epsilon = -\frac{1}{n}$ 时相当于移除 z ，可以线性近似移除 z 后的参数变化 $\hat{\theta}_{-z} - \hat{\theta} \approx -\frac{1}{n} \mathcal{I}_{\text{up, params}}(z)$ 。

基于 $\mathcal{I}_{\text{up,params}}(z)$, 通过如下解析解衡量在对训练语料 z 进行提权后, 对测试点 z_{test} 的影响:

$$\begin{aligned}\mathcal{I}_{\text{up,loss}}(z, z_{\text{test}}) &\stackrel{\text{def}}{=} \frac{dL(z_{\text{test}}, \hat{\theta}_{\epsilon, z})}{d\epsilon} \Big|_{\epsilon=0} \\ &= \nabla_{\theta} L(z_{\text{test}}, \hat{\theta})^T \frac{d\hat{\theta}_{\epsilon, z}}{d\epsilon} \Big|_{\epsilon=0} \\ &= -\nabla_{\theta} L(z_{\text{test}}, \hat{\theta})^T H_{\hat{\theta}}^{-1} \nabla_{\theta} L(z, \hat{\theta})\end{aligned}\quad (14.14)$$

对训练语料中某个数据 $z = (x, y)$ 进行微小改动, 对模型预测的影响的问题。首先定义改动后的数据 $z_{\delta} \stackrel{\text{def}}{=} (x + \delta, y)$, $\hat{\theta}_{z_{\delta}, -z}$ 表示通过使用 z_{δ} 代替 z 后的训练语料, 根据经验风险最小化训练得到的参数。为了估计该影响, 定义 ϵ 权重下从 z 更换为 z_{δ} 的模型参数为:

$$\hat{\theta}_{\epsilon, z_{\delta}, -z} \stackrel{\text{def}}{=} \arg \min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^n L(z_i, \theta) + \epsilon L(z_{\delta}, \theta) - \epsilon L(z, \theta) \quad (14.15)$$

类似公式14.13, 可以得到:

$$\begin{aligned}\frac{d\hat{\theta}_{\epsilon, z_{\delta}, -z}}{d\epsilon} \Big|_{\epsilon=0} &= \mathcal{I}_{\text{up,params}}(z_{\delta}) - \mathcal{I}_{\text{up,params}}(z) \\ &= -H^{-1} (\nabla_{\theta} L(z_{\delta}, \hat{\theta}) - \nabla_{\theta} L(z, \hat{\theta}))\end{aligned}\quad (14.16)$$

与前面的计算类似, 也可以采用线性近似得到 $\hat{\theta}_{z_{\delta}, -z} - \hat{\theta} \approx \frac{1}{n}(\mathcal{I}_{\text{up,params}}(z_{\delta}) - \mathcal{I}_{\text{up,params}}(z))$ 。如何 x 是连续的, 并且 δ 足够小, 还可以进一步的对公式14.16进行估计。假设输入样本空间 $\mathcal{X} \in \mathbb{R}^d$, 参数空间 $\Theta \in \mathbb{R}^p$, 并且 L 对 θ 和 x 可导。因为 $\|\delta\| \rightarrow 0$, 因此 $\nabla_{\theta} L(z_{\delta}, \hat{\theta}) - \nabla_{\theta} L(z, \hat{\theta}) \approx [\nabla_x \nabla_{\theta} L(z, \hat{\theta})]\delta$, 其中 $\nabla_x \nabla_{\theta} L(z, \hat{\theta}) \in \mathbb{R}^{p \times d}$ 。代入公式14.16, 可以得到:

$$\frac{d\hat{\theta}_{\epsilon, z_{\delta}, -z}}{d\epsilon} \Big|_{\epsilon=0} \approx -H_{\hat{\theta}}^{-1} [\nabla_x \nabla_{\theta} L(z, \hat{\theta})]\delta \quad (14.17)$$

因此, $\hat{\theta}_{z_{\delta}, z} - \hat{\theta} \approx -\frac{1}{n}H_{\hat{\theta}}^{-1}[\nabla_x \nabla_{\theta} L(z, \hat{\theta})]\delta$, 并由此可以到相应的影响函数:

$$\begin{aligned}\mathcal{I}_{\text{pert,loss}}(z, z_{\text{test}}) &\stackrel{\text{def}}{=} \nabla_{\delta} L(z_{\text{test}}, \hat{\theta}_{z_{\delta}, -z}) \Big|_{\delta=0} \\ &= -\nabla_{\theta} L(z_{\text{test}}, \hat{\theta})^T H_{\hat{\theta}}^{-1} \nabla_x \nabla_{\theta} L(z, \hat{\theta})\end{aligned}\quad (14.18)$$

在定义了上述两种影响函数后, 直接按照上述公式进行计算的所需的计算量很大。主要原因在于需要对 n 个训练样本计算海森矩阵并取平均和求逆。同时还需要对训练语料中的所有样本, 针对

测试样本都计算影响函数。使用隐式海森向量积 (Hessian-vector Product) 来近似计算, 定义 $s_{\text{test}} \stackrel{\text{def}}{=} H_{\hat{\theta}}^{-1} \nabla_{\theta} L(z_{\text{test}}, \hat{\theta})$ 。由此 $\mathcal{I}_{\text{up}, \text{loss}}(z, z_{\text{test}}) = \nabla_{\theta} L(z_{\text{test}}, \hat{\theta})^T H_{\hat{\theta}}^{-1} \nabla_{\theta} L(z, \hat{\theta})$ 可以转化为 $\mathcal{I} = -s_{\text{test}} \cdot L(z, \hat{\theta})$ 。文献 [29] 提出了两种近似计算方法, 详细过程可以参考文献内容。

14.3.3 可解释评估

目前自然语言处理的评估方法通常基于公开数据集合, 使用统计机器学习中常用的准确率、精度、召回、F1 值等指标进行评价。虽然这种评价方法极大地推动自然语言处理的高速发展, 但是近年来也逐渐暴露出了单一的粗粒度指标无法很好的区分不同系统之间的在细粒度任务维度上的优势和劣势等问题。可解释评估 (Interpretable Evaluation) 旨在通过对特定任务设计多个不同的可解释属性, 细粒度地评估模型在一个或多个数据集上不同属性类的性能, 并对模型偏差、数据集偏差及二者之间的相关性进行评价。可解释评估的主要流程包括:

- (1) 属性定义: 针对特定任务设计多个不同的可解释属性 (例如, 针对命名实体识别任务设置实体类型、实体长度等属性);
- (2) 样本分桶: 计算待测试的样本的属性值, 并将样本放入符合相关属性的桶中
- (3) 分桶性能评估: 评估每个分桶中的样本的性能, 进行细粒度评估。

Fu 等人在文献 [31] 中针对命名实体识别任务的可解释评估如图14.11所示。针对命名实体识别任务定义了包括实体长度, 标签一致性, 实体密度, 句子长度等在内的属性。针对预先的定义的命名实体识别任务属性, 计算测试样例“Life in New York is fun.”中的“New York”的所对应的属性值。本例中, 针对实体长度属性, “New York”所对应的属性值为 2, 因此将本测试样例放入实体长度为 2 的分桶中。当将整个测试集中的实体分类放入对应的存储桶之后, 分别计算每个分桶中的实体识别性能。

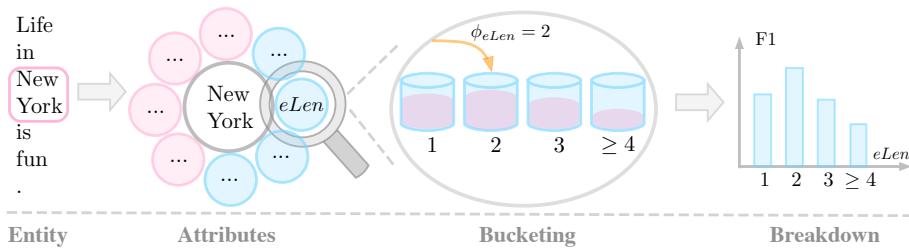


图 14.11 命名实体识别任务可解释评估示例^[31]

在可解释评估基础上, 文献 [31] 中还提出了两种模型诊断 (Model Diagnosis) 方法:

- (1) 自我诊断: 给定模型和特定的评估属性 (例如, 实体长度), 获得测试样本的性能在其中获得最高值和最低值的桶。可以帮助诊断特定模型在哪些条件下表现良好或较差;
- (2) 比较诊断: 给定两个模型 M_1 和 M_2 以及特定属性, 对比两个系统之间的性能差距达到最高值和最低值的桶。可以指示系统之间在哪些条件下可能优势和劣势。

图14.12给出了在6个命名实体识别数据集中的模型诊断结果，其中 M_1 和 M_2 表示两个模型。属性值被分为四类：特小(XS)、小(S)、大(L)和特大(XL)。在自我诊断直方图中，绿色(红色)的X轴刻度标签表示系统在该桶上获得最佳(最差)性能。灰色柱子表示性能最差，蓝色的柱子表示最佳性能和最差性能之间的差距。通过图14.12所给出的基于BERT的命名实体模型的自我诊断的结果可以观察到，如果实体在训练集和测试集上的标签一致程度(eCon, tCon)较低的情况下或者实体频率(eFre)较低的情况下，模型的结果较差。通过CRF和MLP的对比诊断，还可以发现CRF可以在长实体上相较于MLP有更好的性能。

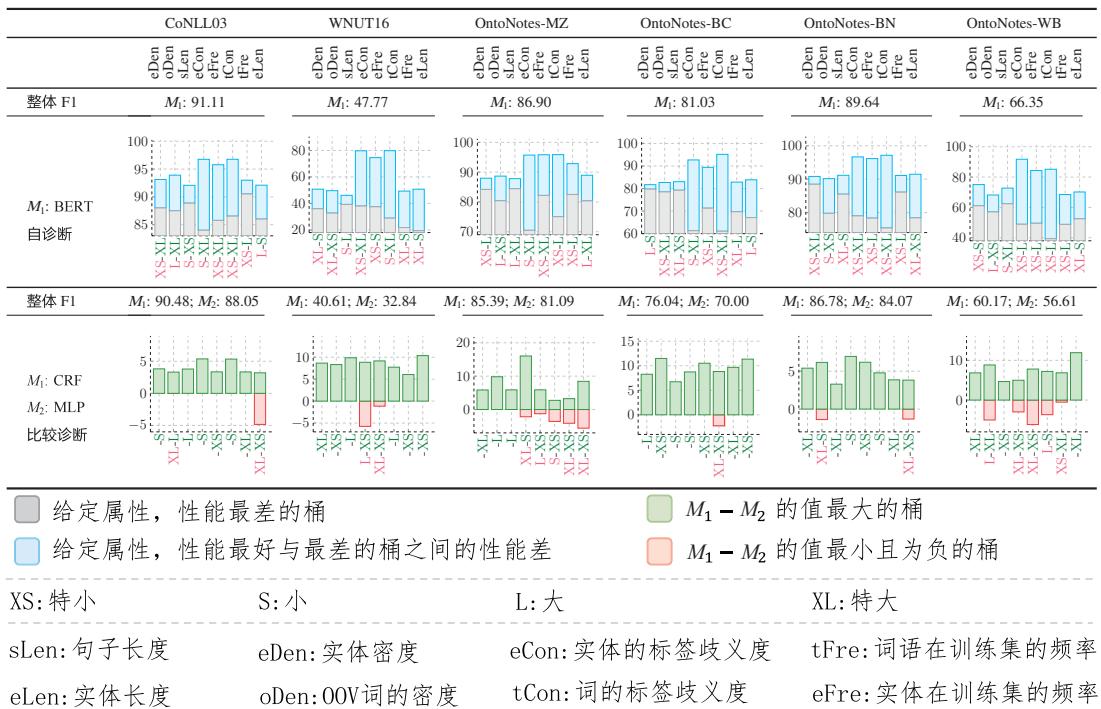


图 14.12 命名实体识别任务模型诊断结果示例^[31]

可以看到，可解释评估的分析结果相较于传统的单一评价指标，可以更好的进行模型错误类型统计、归类与分析，从而对模型的性能来源做出解释。错误样本分类的依据可以是语法、语义或任务相关的特征。具有能够发现某个任务的困难样本类型(例如，未登录词、标签不一致)；能够对比不同模型的优缺点(例如，模型A比模型B在某类型样本上错误更少)；还能够通过对比发现不同结构所起的作用(例如，基于LSTM的模型比基于CNN的模型在句子较长的样本上错误更少，说明LSTM能更好地建模长距离依赖关系)。在此基础上，Liu等人提出了可解释排行榜Explaina Board^[32]，使研究人员可以采用人机交互的评估方式，利用模型自我诊断、系统辅助诊

断和数据偏差分析等可解释评估方法，对模型优势和劣势，以及数据集的偏差进行更细粒度的分析。

14.4 延伸阅读

14.5 习题

参考文献

- [1] 杨强等. 可解释人工智能导论[M]. 电子工业出版社, 2022.
- [2] Council of European Union. (eu)2016/679 general data protection regulation(gdpr)[Z]. 2018.
- [3] Kim B, Wattenberg M, Gilmer J, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)[C]//International conference on machine learning. PMLR, 2018: 2668-2677.
- [4] Blunsom P, Camburu O M, Foerster J, et al. Can i trust the explainer? verifying post- hoc explanatory methods[J]. CoRR, 2019.
- [5] Adebayo J, Gilmer J, Muelly M, et al. Sanity checks for saliency maps[J]. Advances in neural information processing systems, 2018, 31.
- [6] Yeh C K, Hsieh C Y, Suggala A, et al. On the (in) fidelity and sensitivity of explanations[J]. Advances in Neural Information Processing Systems, 2019, 32.
- [7] Warnecke A, Arp D, Wressnegger C, et al. Evaluating explanation methods for deep learning in security[C]//2020 IEEE european symposium on security and privacy (EuroS&P). IEEE, 2020: 158-174.
- [8] Ribeiro M T, Singh S, Guestrin C. "why should I trust you?": Explaining the predictions of any classifier[C/OL]//Krishnapuram B, Shah M, Smola A J, et al. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016. ACM, 2016: 1135-1144. <https://doi.org/10.1145/2939672.2939778>.
- [9] Shrikumar A, Greenside P, Kundaje A. Learning important features through propagating activation differences[C/OL]//Precup D, Teh Y W. Proceedings of Machine Learning Research: volume 70 Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017. PMLR, 2017: 3145-3153. <http://proceedings.mlr.press/v70/shrikumar17a.html>.

24 自然语言处理导论 -- 张奇、桂韬、黄萱菁

- [10] Simonyan K, Vedaldi A, Zisserman A. Deep inside convolutional networks: Visualising image classification models and saliency maps[C/OL]//Bengio Y, LeCun Y. 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings. 2014. <http://arxiv.org/abs/1312.6034>.
- [11] Smilkov D, Thorat N, Kim B, et al. Smoothgrad: removing noise by adding noise[J/OL]. CoRR, 2017, abs/1706.03825. <http://arxiv.org/abs/1706.03825>.
- [12] Sundararajan M, Taly A, Yan Q. Axiomatic attribution for deep networks[C/OL]//Precup D, Teh Y W. Proceedings of Machine Learning Research: volume 70 Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017. PMLR, 2017: 3319-3328. <http://proceedings.mlr.press/v70/sundararajan17a.html>.
- [13] Mudrakarta P K, Taly A, Sundararajan M, et al. Did the model understand the question?[C]// Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2018: 1896-1906.
- [14] Shapley L S. A value for n-person games, contributions to the theory of games, 2, 307–317[M]. Princeton University Press, Princeton, NJ, USA, 1953.
- [15] Strumbelj E, Kononenko I. Explaining prediction models and individual predictions with feature contributions[J/OL]. Knowl. Inf. Syst., 2014, 41(3):647-665. <https://doi.org/10.1007/s10115-013-0679-x>.
- [16] Erhan D, Bengio Y, Courville A, et al. Visualizing higher-layer features of a deep network[J]. 2009.
- [17] Simonyan K, Vedaldi A, Zisserman A. Deep inside convolutional networks: Visualising image classification models and saliency maps[J]. arXiv preprint arXiv:1312.6034, 2013.
- [18] Mahendran A, Vedaldi A. Visualizing deep convolutional neural networks using natural pre-images [J]. International Journal of Computer Vision, 2016, 120(3):233-255.
- [19] Nguyen A, Dosovitskiy A, Yosinski J, et al. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks[J]. Advances in neural information processing systems, 2016, 29.
- [20] Bastani O, Kim C, Bastani H. Interpreting blackbox models via model extraction[J/OL]. CoRR, 2017, abs/1705.08504. <http://arxiv.org/abs/1705.08504>.

- [21] Breiman L, Friedman J, Olshen R, et al. Classification and regression trees—crc press[J]. Boca Raton, Florida, 1984.
- [22] Li J, Chen X, Hovy E, et al. Visualizing and understanding neural models in nlp[C]//Proceedings of NAACL-HLT. 2016: 681-691.
- [23] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[C]//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). 2019: 4171-4186.
- [24] Lan Z, Chen M, Goodman S, et al. ALBERT: A lite BERT for self-supervised learning of language representations[C/OL]//8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net, 2020. <https://openreview.net/forum?id=H1eA7AEtvS>.
- [25] Manning C D, Clark K, Hewitt J, et al. Emergent linguistic structure in artificial neural networks trained by self-supervision[J/OL]. Proc. Natl. Acad. Sci. USA, 2020, 117(48):30046-30054. <https://doi.org/10.1073/pnas.1907367117>.
- [26] Conneau A, Kruszewski G, Lample G, et al. What you can cram into a single vector: Probing sentence embeddings for linguistic properties[J/OL]. CoRR, 2018, abs/1805.01070. <http://arxiv.org/abs/1805.01070>.
- [27] Vulic I, Ponti E M, Litschko R, et al. Probing pretrained language models for lexical semantics [C/OL]//Webber B, Cohn T, He Y, et al. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020. Association for Computational Linguistics, 2020: 7222-7240. <https://doi.org/10.18653/v1/2020.emnlp-main.586>.
- [28] Hewitt J, Liang P. Designing and interpreting probes with control tasks[C/OL]//Inui K, Jiang J, Ng V, et al. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019. Association for Computational Linguistics, 2019: 2733-2743. <https://doi.org/10.18653/v1/D19-1275>.
- [29] Koh P W, Liang P. Understanding black-box predictions via influence functions[C]//International conference on machine learning. PMLR, 2017: 1885-1894.

26 自然语言处理导论 -- 张奇、桂韬、黄萱菁

- [30] Cook R D, Weisberg S. Residuals and influence in regression[M]. New York: Chapman and Hall, 1982.
- [31] Fu J, Liu P, Neubig G. Interpretable multi-dataset evaluation for named entity recognition[J]. arXiv preprint arXiv:2011.06854, 2020.
- [32] Liu P, Fu J, Xiao Y, et al. Explainaboard: An explainable leaderboard for nlp[C]//Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations. 2021: 280-289.

索引

Accountability, 3

Activation Maximization, 9

Algorithm Decomposability, 3

Algorithm Transparency, 3

Applicable border, 3

Explainable Artificial Intelligence, XAI, 1

Interpretability, 1

Interpretable Evaluation, 20

Post-hoc Explanation , 3

Saliency Map, 6

Shapley Value, 9

Submodular Optimization, 11

全局解释, 5

可解释人工智能, 1

可解释性, 1

可解释评估, 20

局部解释, 5

显著图, 6

次模优化, 11

沙普利值, 9

激活最大化, 9

算法事后解释, 3

算法可担责性, 3

算法可解构性, 3

算法适用边界, 3

算法透明度, 3