ChatGPT

可用于 OpenAI API 模型的黑盒文本分类解释方法

为了在 **仅通过 OpenAI API 调用**（无法访问模型权重）的文本分类任务中替代 LIME，实现关键词贡献的解释与可视化，可以考虑以下几种方法。这些方法均支持对模型预测提供 **rationale/关键词高亮** 的解释输出，并满足透明性要求。

1. SHAP（Shapley 加法解释）

适用性： SHAP 是一种模型无关的特征归因方法，适用于黑盒模型。对于文本分类，可将每个单词视为特征，利用 SHAP 计算每个单词对模型预测的贡献度 [1] [2]。它不需要访问模型内部权重，只需能够调用模型预测接口。

依赖要求： 需要使用现有的 SHAP 库（例如 Python 的 `shap` 包）实现 *Kernel SHAP*。该库会对模型进行多次调用采样，从而估计 Shapley 值。安装和使用该库是主要依赖；除此之外无其他复杂依赖。

使用方式： 定义一个调用 OpenAI 接口的预测函数，然后使用 Kernel SHAP 对单词存在/缺失进行采样评估每个特征的边际贡献 [1]。对于每个输入文本，SHAP 返回该文本中各单词的 Shapley 值。值为正表示该单词推动模型朝某类别预测，值为负则相反。

解释能力： SHAP 考虑 **所有可能的特征组合**对预测的影响，因而提供理论上一致的特征重要性度量 [3]。它能够给出**局部解释**（单条预测的关键字）和**全局解释**（整体最重要的特征） [4]。对于单个文本，检查 SHAP 值可以识别出对特定预测最有影响的关键单词 [2]。这种方法的解释质量高，具有良好的一致性和公平性（源于博弈论的 Shapley值原理），能够帮助揭示模型决策中最重要的词语 [5]。

HTML 可视化： SHAP 提供了丰富的可视化工具。对于文本数据，SHAP 可以高亮显示输入句子中各单词，并以颜色深浅表示其对预测的正负贡献 [5]。例如，可使用 SHAP 自带的文本可视化（如 **force plot** 或 **text plot**）在前端呈现高亮结果 [1]。这些可视化可以保存为交互式的 HTML 文件，或通过将单词及其贡献值输出，手动包装 `<span>` 等标签加上背景色，实现前端高亮展示。

2. Anchors（锚定解释方法）

适用性： Anchors 是由 LIME 作者提出的另一种模型无关本地解释方法，适用于黑盒分类模型 [6]。该方法特别适合需要**高精度解释**的场景：它寻找一组"锚定"条件——例如文本中某些 **关键单词的存在**——使得当这些条件满足时，模型的预测基本不变 [6]。对于 OpenAI API 提供的分类结果，我们可以使用 Anchors 找出**足以支撑当前预测的关键词集合**，即模型决策的充分条件。

依赖要求： Anchors 有现成的开源实现（例如 Python 的 `anchor` 库，已集成在 Seldon 的 Alibi 库中) [7]。使用该方法需要安装相应库或引用作者提供的实现。Anchors 利用强化学习和采样搜索策略寻找锚定特征，因此相对于 LIME/SHAP计算量更大 [8]。如果环境不便安装重型库，也可以尝试自行实现一个简化版本（需要编写代码随机遮蔽部分词语并评估模型预测，一般较复杂，不如直接使用现有实现）。

使用方式： 对于待解释文本，Anchors 算法反复**采样扰动**该文本（例如随机替换或删除部分单词），并观察在包含某些单词时预测是否始终保持不变 [9]。通过迭代搜索，找到一组单词作为**锚定条件**：当这些单词出现时，无论其

他词如何变化，模型输出大概率不变。这组锚定词即为模型此时决策的依据。Anchors 输出一个 **IF-THEN规则**形式的解释，例如："如果文本包含词 X 和 Y，则模型预测为正类"。

解释能力： Anchors 强调解释的**精准度**而非覆盖所有特征。它提供的 rationale 通常是少数几个对预测**至关重要**的词语，具有高度可信度（因为只要锚定词在，预测基本不会变）[6]。这种解释直观易懂（类似于决策规则），能够明确指出哪些关键词在**驱动模型预测**。但由于只提供锚定集合，可能无法像 SHAP 那样量化每个词的细粒度贡献、也不一定覆盖所有重要词汇。总体而言，其解释简洁明了，符合比赛要求的"rationale/关键词"可视化目标。

HTML 可视化： Anchors 的输出可视为需要高亮的关键条件词集合。前端可以将这几个锚定单词高亮（例如着重标出或用特定颜色标记）。由于 Anchors 不直接输出带权重的所有词列表，而是输出满足条件的词集合，**可视化**时只需突出显示这些锚词即可（例如将它们用粗体或底色标出)。Anchors 原生实现主要以文本或规则形式给出解释，但开发者可很容易地将结果转换为 HTML 高亮展示关键字。

3. 集成梯度（Integrated Gradients）

适用性： 集成梯度是一种**白盒**解释方法，需要获取模型对输入的梯度信息[10]。它通常用于可微分的深度学习模型，对于**开放的**文本分类模型（如自行训练的神经网络）非常有效。然而，对于 OpenAI API 提供的 GPT-3.5/4 这类黑盒模型，无法直接计算梯度，因此**无法直接使用**集成梯度方法。只有在有等价的本地模型或 OpenAI 提供梯度接口的情况下才能应用。如果比赛允许使用一个在相似数据上训练的替代模型，便可对该模型使用集成梯度来解释预测，然后近似用于GPT模型的输出解释。

依赖要求： 需要深度学习框架（如 TensorFlow/PyTorch）以及相关工具库（例如 `captum` 或 Alibi）来计算梯度。模型本身必须可访问其参数和计算图。若使用现有实现，可安装 `captum`（PyTorch）或 Alibi（TensorFlow）中的集成梯度解释器。在纯 OpenAI API 情况下无额外依赖，因为无法使用。

使用方式： 集成梯度通过选定一个**基线输入**（例如全空白或无信息的输入），然后逐步将输入从基线变换为原始文本，累积模型梯度来评估每个特征的贡献[10]。具体而言，对文本分类模型，通常以全零或 `[PAD]` 序列作为基线，将其逐步演变成真实句子，累加每一步词嵌入梯度。最终得到每个单词的归因分值[10]。这种方法满足**敏感性**和**实现公平性**等归因准则，往往比简单梯度更加稳定。

解释能力： 如果能够应用，集成梯度能为每个输入词提供一个**准确且可微分析的贡献分值**。它考虑了从无信息状态到输入文本的整个路径上的梯度变化，因而在反映模型决策依据方面更具**可信度和一致性**[10]。实验表明集成梯度在许多情况下能生成高质量的逐词重要性评分，突出真正影响模型预测的关键词。同时，它对微小扰动不敏感，解释结果相对平滑。总体来说，其解释具有较高的可靠性和清晰度。但再次强调，在纯黑盒设置下无法直接获取此方法的结果。

HTML 可视化： 集成梯度输出每个单词的贡献值后，可轻松用于 HTML 高亮。通常做法是根据贡献值的大小和正负，对文本中各单词添加<span style>样式：例如正向贡献的词用绿色背景（深浅对应贡献度大小），负向贡献的词用红色背景。这样，前端即可直观展示哪些词推动模型预测结果、哪些词起反方向作用。由于集成梯度本身不提供现成的 HTML，可视化需要开发者将其计算结果映射为相应的前端呈现。但这些都是简单的基于分值的渲染，符合比赛对前端高亮关键词的要求。

4. 基于扰动的词重要性（Occlusion/Permutation 方法）

适用性： 这种方法不依赖任何外部库，是一种**零依赖的替代方案**，非常适合无法安装复杂解释库的环境。它完全将模型视为黑盒，仅通过多次调用模型接口实现解释 [11]。其思想是在**不修改模型**的情况下，直接观察**输入某部分变动对预测的影响**。对于文本分类，就是逐个**移除或遮蔽单词**，观察模型预测概率的变化。这一方法简单直接，适用于所有可以调用预测概率的黑盒模型。

依赖要求： 无需额外库，仅需编写代码循环调用 OpenAI API 接口。开发者需要能够获取模型对给定文本的预测分数或概率分布，以衡量移除某词前后预测的差异。如果只能获得离散标签，也可以通过统计标签改变与否来近似重要性。

使用方式： 对于一条待解释文本，首先获取模型对完整文本的预测（及其置信度）。然后针对文本中的每个单词，构造一个**删去该单词的输入**（或将其替换为一个无信息的标记，如 "[UNK]"），再调用模型获得新预测 [12]。比较新旧预测：通常以**预测概率的下降量**作为该词对原预测支持的贡献度 [13]。例如，原文本模型预测为正类概率90%，移除某词后变为70%，则该词贡献了20%的正类概率。如果移除词导致目标类别概率上升，说明该词原先对该类别**起反作用**。依次计算出每个单词的重要性评分。

解释能力： 这种**逐词遮蔽**的办法（也称"遮挡法"或 Leave-One-Out）能够产出直接明了的解释：重要性分数高的词就是对模型判定影响最大的关键词 [13]。它完全基于模型实际输出变化，因而相对直观可信。此外，它属于**真实的黑盒方法**，不需要任何模型内部信息即可评估特征影响 [11]。这种方法的结果往往和人类直觉一致，例如在情感分类中，移除"excellent"导致模型不再判为正向，则显然"excellent"是正面关键词。不过，该方法也有局限：如果模型对局部扰动不敏感（例如上下文依赖强），可能低估一些词的重要性；同时逐词调用模型的开销较大，在长文本或大批量情况下效率较低。

HTML 可视化： 由于已经为每个单词获得了一个定量的贡献分值，前端高亮十分容易实现。可以采用颜色梯度或透明度来展示：例如，用背景颜色深浅来表示贡献度大小，用颜色（红/绿）表示正负影响。开发者只需将每个词用 `<span>` 包裹并附上对应的样式（例如 `style="background-color: rgba(0,255,0,0.5)"` 这类）即可生成 HTML 格式的高亮文本。这样的输出能够直观展现 **"哪几个关键词是模型判定的依据"**，非常契合比赛中 transparency 模块对 **rationale/关键词可视化** 的要求。

5. 其它补充方法（轻量级备选方案）

除了上述主流方法，还可以考虑一些特殊情况下的备选手段：

- **基于大型模型自身的解释：** 利用 GPT-3.5/4 本身的能力，请求模型输出对其分类决策的解释或让其用标记符号标出重要词。这种方法零依赖且实现简单（直接在 prompt 中要求模型给出解释或高亮词语）。它能生成**可读的自然语言原因说明**或直接在文本中标注关键词。然而，该解释基于模型的自我报告，**未必忠实**于实际决策依据，可能出现"看似合理但并非模型真正依据"的情况。因此，可将其作为辅助手段或在资源受限时的**后备方案**。

- **训练可解释的替代模型：** 如果有足够的样本和时间，可以用 GPT API 大量标注数据，然后训练一个简单的可解释模型（如逻辑回归或小型树模型）近似模拟GPT的分类结果。这样的模型本身具有透明的特征权重，可直接用于解释。例如，逻辑回归会给出每个词对分类的权重系数，正负表示推动或抑制某类的作用。这些权重可用于高亮输入文本中的重要词。该方案需要一些训练过程，但**依赖轻量**（只需常规机器学习库），可作为无法频繁调用GPT时的fallback。

上述方法各有侧重：SHAP 提供全面精确的逐词贡献度，Anchors 给出高精度的关键条件解释，集成梯度在白盒情况下效果佳，扰动法简单直接零依赖，而让模型自解释或训练替代模型则可在特定限制下采用。综合考虑实现难度、运行效率和解释质量，**推荐优先使用** 基于扰动的词重要性 方法作为零依赖方案，或 **SHAP** 作为成熟的高质量方案；在有算力或实现需求时尝试 Anchors；若能访问模型梯度则考虑集成梯度。所有这些方法都能输出可用于前端渲染的 HTML 格式高亮结果，满足比赛对于 **"透明性模块输出可视化rationale/关键词"** 的要求。各方案的选择应根据实际限制和平衡解释准确性与开销来确定。

参考文献：

- Ribeiro et al., *"Anchors: High-Precision Model-Agnostic Explanations"*, AAAI 2018.
- Lundberg & Lee, *"A Unified Approach to Interpreting Model Predictions (SHAP)"*, NeurIPS 2017.
- Sundararajan et al., *"Axiomatic Attribution for Deep Networks (Integrated Gradients)"*, ICML 2017.
- [1] [2] SHAP 方法的计算原理及其局部/全局解释能力描述
- [5] SHAP 对文本分类中单词影响的可视化示例说明
- [6] Anchors 方法及其"锚定"条件的定义与耗时特点说明
- [13] [14] 遮蔽法（Occlusion）在NLP中作为黑盒解释的原理（对比移除特征前后模型预测变化）

---

[1] [2] [3] [4] [6] [8] [9] Demystifying Explainable AI in NLP: LIME, SHAP, and ANCHOR Explained
https://www.toolify.ai/ai-news/demystifying-explainable-ai-in-nlp-lime-shap-and-anchor-explained-475899

[5] SHAP for text-based data
https://www.linkedin.com/pulse/shap-text-based-data-vizuara-tldoc

[7] 16 Scoped Rules (Anchors) – Interpretable Machine Learning
https://christophm.github.io/interpretable-ml-book/anchors.html

[10] Integrated gradients for text classification on the IMDB dataset — Alibi 0.9.7.dev0 documentation
https://docs.seldon.io/projects/alibi/en/latest/examples/integrated_gradients_imdb.html

[11] [12] [13] [14] Explaining Natural Language Processing Classifiers with Occlusion and Language Modeling
https://arxiv.org/pdf/2101.11889