基于机器学习的情感分析方法

一、课题综述

1.1 课题说明

小组成员任务划分：王政尧和刘依扬负责对数据进行处理以及训练传统机器学习模型；牛昱琛负责训练深度学习模型，并研究 BERT 的可解释性，报告为三人共同完成。三人对该实验项目有同等贡献度。

1.2 课题目标

课题的目标为使用机器学习技术进行文本的情感分析任务，即给定一个文本，判断该文本所蕴含的情感为积极、消极或其他类别。课题选用的数据集来源于社交媒体平台，包含了大量用户发布的文本数据以及人工标注好的情感类别。首先，我们对数据集进行预处理，包括文本清洗、去噪、分词等步骤，以确保数据的质量和分析的准确性。随后，我们运用 Bag of Words、TF-IDF 和词(句)向量化等技术提取文本特征，并构建多个机器学习模型，如支持向量机和逻辑回归等，进行情感分类任务。除此之外，我们还探讨了深度学习模型在情感分析中的应用，如 RNN、LSTM 和 BERT 等。我们将训练并优化这些模型，以进一步提高情感分析的准确度和效率。最后，我们将对经训练的各模型进行比较分析，探讨不同模型的效果及其架构差异，并着重于模型的可解释性。

1.3 课题数据集

本课题使用的数据集源自 Hugging Face 数据集库中的 "DAIR-AI/emotion"（https://huggingface.co/datasets/dair-ai/emotion）。这个数据集专门用于自然语言处理中的情感分析任务，包括了六种主要的情感类别：快乐（Happiness）、悲伤（Sadness）、愤怒（Anger）、恐惧（Fear）、惊讶（Surprise）和爱（Love）。总计超过 16000 条文本数据被收录在此数据集中，这些文本涵盖了从社交媒体帖子到新闻报道，乃至个人日记等多种来源。这些文本不仅展现了多样的情感色彩，还呈现出各种不同的语言风格和表达形式。本数据集的主要任务是将这些文本内容准确地归类到它们对应的情感类别。

二、实验内容

2.1 原始数据预览与分析

数据集的总体观察表明，各类情感数据的分布存在显著不均衡。这种数据倾斜可能导致机器学习模型的训练效果受限，因为模型可能会偏向于更频繁出现的

类别。为了解决这一问题，我们计划采用数据增强技术，以实现数据分布的均衡化。

text string · lengths 7 300	label class label 6 classes
i didnt feel humiliated	0 sadness
i can go from feeling so hopeless to so damned hopeful just from being around someone who cares and is awake	0 sadness
im grabbing a minute to post i feel greedy wrong	3 anger
i am ever feeling nostalgic about the fireplace i will know that it is still on the property	2 love
i am feeling grouchy	3 anger
ive been feeling a little burdened lately wasnt sure why that was	0 sadness
ive been taking or milligrams or times recommended amount and ive fallen asleep a lot faster but i also feel like so funny	5 surprise

2.2 数据预处理

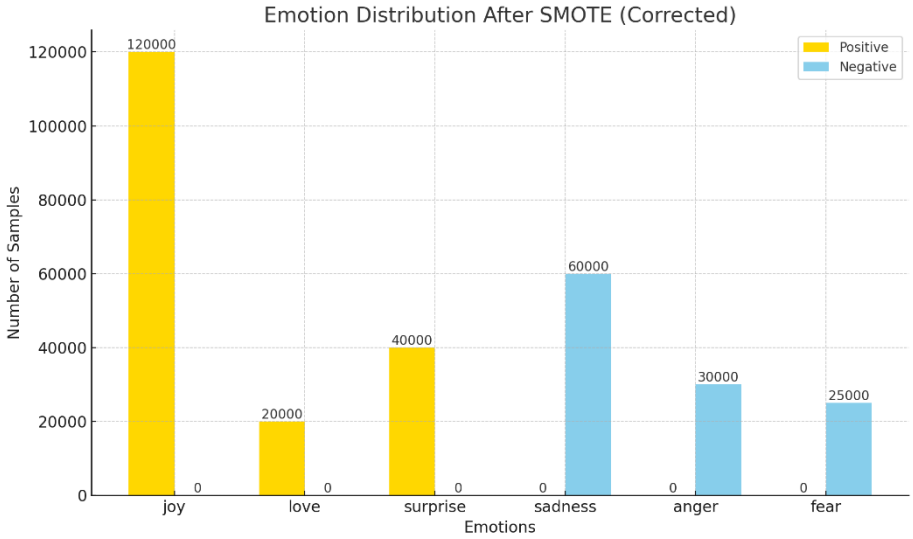
首先进行数据清洗工作，去除文本中的标点、emoji、特殊字符等，同时所有大写全部转为小写。之后为平衡不同类别的样本数量，我们使用 SMOTE (Synthetic Minority Over-sampling Technique) 技术对样本进行模拟，增加少数类别的样本数量。算法如下所示：

SMOTE Algorithm

Input: *Minority class sample set* S, *Over* $-$ *sampling ratio* N
Output: *Synthetic sample set* S'

Initialize synthetic sample set S' *as empty*
For each sample s_i *in* S:
  Find k *nearest neighbors of* s_i *in* S
  For each nearest neighbor s_{nj}:
    For j *from* 1 *to* N (*integer*):
      Generate a random number λ *between* 0 *and* 1
      Create new sample $s_{new} = s_i + \lambda * (s_{nj} - s_i)$
      Add s_{new} *to* S'
Return synthetic sample set S'

处理后的数据集各类别样本的数量分布情况如下图所示，可以看到过采样后的样本分布更为均衡。



2.3 使用传统机器学习分类方法

2.3.1 使用 TF-IDF 进行特征提取

在传统机器学习任务中，训练模型所需的输入是具有给定维度的特征向量。因此，如何从文本中有效提取特征信息成为了关键问题。目前广泛使用的三种文本特征提取方法包括：

1. 词袋模型（BoW）：这是一种简单的特征提取技术，通过构建一个词汇库，将文本转换为词汇出现次数的向量。每个文档都被表示为一个长向量，向量中的每个元素代表对应词汇在文档中的频率。
2. 潜在狄利克雷分配（LDA）：LDA 是一种主题模型，它假设文档是由隐含主题的混合生成的。通过这种方式，它可以捕捉文档中词汇的共现关系，用于文本数据的降维和特征提取。
3. 词频-逆文档频率（TF-IDF）：TF-IDF 是一种统计方法，用以评估一个词语对于一个文档集或一个语料库中的其中一份文档的重要性。它是通过比较词语在特定文档中的频率和在整个语料库中的分布频率来计算得出的，以此来强调那些在特定文档中重要但在整个文档集中不常见的词语。

经过对比分析，我们最终选择了 TF-IDF 作为特征提取方法。TF-IDF 可以有效地区分出各个类别中独有的情感表达词汇，从而增强模型对于情感倾向的识别能力。相比之下，词袋模型可能会忽略词汇的权重差异，而 LDA 虽然能够揭示文本的主题，但对于细腻的情感变化捕捉可能不够敏感。因此，综合考虑准确性和计算效率，TF-IDF 为我们的情感分析任务提供了一个理想的特征集。提取特征算法如下：

Text Vectorization using TF-IDF Algorithm

$Input$: $A\ set\ of\ documents\ D, containing\ n\ documents\ \{d_1, d_2, \dots, d_n\}$

$Output$: $TF - IDF\ weighted\ feature\ matrix\ V$

1: *Initialize an empty vocabulary set Vocab and an empty TF − IDF matrix V*
2: *For each document* d_i *in D*:
3: *Tokenize and remove stop words to produce a set of terms* T_i
4: *Update the vocabulary set Vocab to include all unique terms*
5: *Prune Vocab according to* $\max_{features}$ *and* $\max_{df}$ *parameters*
6: *For each term* t_j *in the pruned Vocab*:
7: *Calculate the document frequency* $df(t_j)$
8: *Calculate the inverse document frequency* $idf(t_j) = \log\left(n — df(t_j)\right)$
9: *For each document* d_i *in D and each term* t_j *in Vocab*:
10: *Calculate the term frequency* $tf(t_j, d_i)$
11: *Calculate the TF − IDF weight* $w(t_j, d_i) = tf(t_j, d_i) * idf(t_j)$
12: *Form the feature vector* v_i *consisting of* $w(t_j, d_i)$ *for all terms in Vocab*
13: *Add the feature vector* v_i *to the TF − IDF matrix V*
14: *Return the matrix V*

使用上述方法对三个句子进行特征提取，结果以词云表示如下：

2.3.2 搭建机器学习模型并进行训练

我们分别使用了两种机器学习模型，SVM 和 Logistic 回归。在 SVM 中我们通过一对多方法实现多分类任务，将每个类别视为正类别，而将其他所有类别视为负类别，创建多个二元分类器。在训练过程中，每个分类器学习如何区分一个类别与其他所有类别，然后通过置信度分数或概率值来对新数据点进行分类，最终选择具有最高置信度分数的类别作为分类结果。在 Logistic 回归中，模型由线性组合和逻辑斯蒂函数。由于报告页数有限，模型的具体结构以及训练过程见附件中的代码。

两类模型在测试集的结果如下所示：

SVM	Accuracy	Precision	Recall	F1
Sadness	-	0.91	0.90	0.91
Joy	-	0.86	0.86	0.86
Love	-	0.91	0.93	0.92
Anger	-	0.81	0.80	0.80
Fear	-	0.95	0.94	0.94
surprise	-	0.78	0.74	0.76
avg	0.90	0.87	0.86	0.87

Logistic	Accuracy	Precision	Recall	F1
Sadness	-	0.93	0.87	0.90
Joy	-	0.90	0.83	0.86
Love	-	0.94	0.91	0.93
Anger	-	0.85	0.75	0.80
Fear	-	0.96	0.93	0.94
surprise	-	0.83	0.69	0.76
avg	0.85	0.90	0.83	0.86

2.3.3 结果可视化

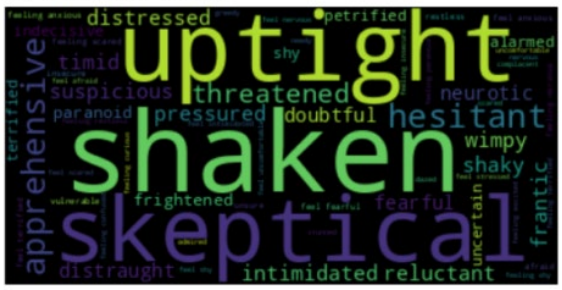
分析 SVM 中各特征矩阵的权重，对每个标签构建下列词云，其中单词越醒目代表权重越高：

“sadness”

	feature	weight
0	greedy	5.261224
1	wronged	4.882603
2	dissatisfied	4.851639
3	outraged	4.613606
4	petty	4.558006

“joy”

	feature	weight
1000	shaken	4.994509
1001	uptight	4.826215
1002	skeptical	4.817414
1003	hesitant	4.793320
1004	apprehensive	4.739991



2.4 使用深度学习分类方法

Transformer 结构的出现引领了自然语言处理领域的变革，因其在并行计算、长距离依赖性、参数效率和性能等方面的优势，逐渐取代了传统的 RNN 结构，成为 NLP 领域的主要架构。因此，我们主要使用了 BERT 作为深度学习模型的代表。

2.4.1 模型介绍

1. BERT 简介

BERT（Bidirectional Encoder Representations from Transformers）是一种基于 Transformers 的开创性的自然语言处理模型，专为深入理解语言而设计。通过在大量未标记的文本上预训练，BERT 学习了深度的双向语言表示，在各种自然语言处理任务上取得了显著的成功，包括文本分类、问答和情感分析等。

2. 模型结构

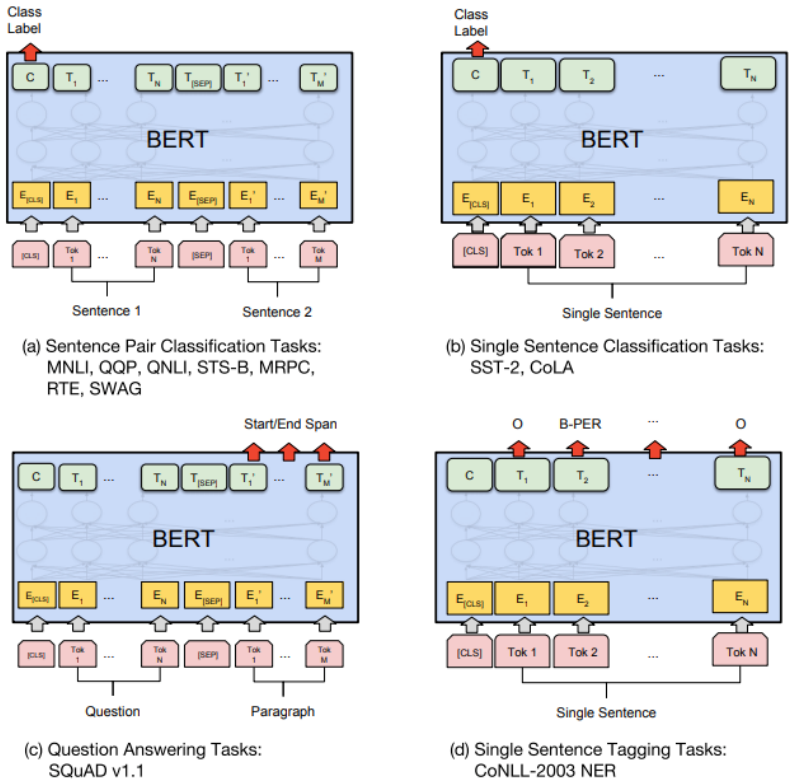
BERT 的核心结构是基于 Transformer 的多层双向编码器。首先，它通过 WordPiece 嵌入方法将输入的句子转换为 token 序列，并在每个序列的开头加上一个特殊的分类标记[CLS]。这些 token 转化为嵌入向量，与位置编码结合，以保

持序列中单词的顺序信息。然后，这些向量输入到多层 Transformer 编码器中。在每一层编码器内，模型通过自注意力机制和前馈网络来学习 token 之间的复杂关系。经过逐层处理，BERT 生成包含丰富上下文信息的特征表示，这为各种下游任务提供了深入的语言理解。最后，[CLS]标记的输出特征被用于分类任务的最终预测。模型有两个版本：BERT Base（12 层，768 个隐藏单元，12 个自注意力头）和 BERT Large（24 层，1024 个隐藏单元，16 个自注意力头）。在本实验中，出于对计算资源的考虑，我们使用了参数量较小的 base 版本。

3. 预训练+微调范式。

不同于之前的机器学习方法直接在给定数据集上进行训练，BERT 采用了预训练和微调的范式(transfer learning)。在预训练阶段，BERT 在大型、未标记的文本数据集上学习语言的通用特征，不依赖于特定的下游任务。这样，它能够理解单词、短语和句子的复杂关系。BERT 的预训练过程包括两种主要任务：Masked Language Model (MLM) 和 Next Sentence Prediction (NSP)。在 MLM 中，BERT 随机遮蔽输入句子中的某些单词，然后尝试预测这些被遮蔽的单词。这种方法使得模型学习到了每个单词的双向上下文。在 NSP 任务中，BERT 预测给定的两个句子是否在原文中相邻。这有助于模型学习理解句子之间的关系，从而更好地处理需要理解句子间关系的任务，如问答和自然语言推理。

之后，在微调阶段，模型在特定任务的数据集上进行进一步训练，调整预训练期间学到的参数，以最大限度地适应和优化对该任务的执行。例如，在情感分析任务中，BERT 会学习如何根据给定文本的内容预测情感倾向，如正面或负面



(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

(b) Single Sentence Classification Tasks: SST-2, CoLA

(c) Question Answering Tasks: SQuAD v1.1

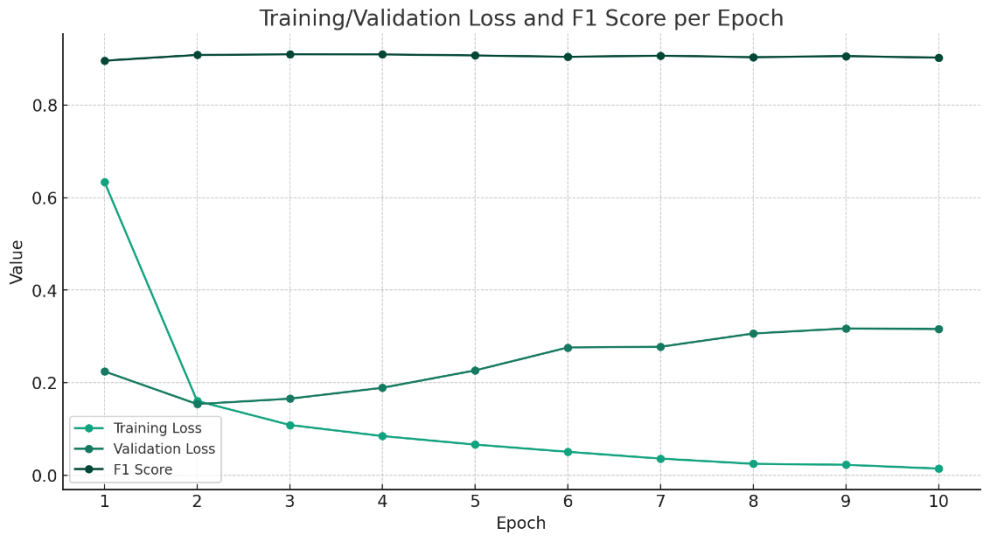
(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

情感。通过这种方式，BERT 结合了预训练学习到的语言通用特征和针对特定任务的细粒度调整，以提高在特定任务上的表现。

2.4.2 模型训练

使用 Hugging Face 库对 BERT 模型进行微调，首先加载"dair-ai/emotion"数据集，之后对文本进行预处理以匹配 BERT 的输入格式。接着，加载预训练的 BERT 模型并添加一个适合情感分类的顶层（模型以及预训练参数使用 bert-base-uncased）。通过定义训练参数并使用 Trainer 类来组织训练和评估流程，对模型进行微调，最后在验证集上评估模型的性能，从而完成对 BERT 模型的微调过程。详细信息见附件中的代码。

模型训练效果如下图所示。可以发现，随着 epoch 增加，训练损失持续下降，表明模型在训练集上拟合得越来越好。在前几个 epoch 中，验证损失下降，说明模型在验证集上的泛化能力在提升。然而，从第 5 个 epoch 开始，验证损失开始上升，这是过拟合的信号，意味着模型在训练集上过度拟合，泛化能力下降。根据上述分析，我们选择了在验证损失最低且 F1 得分相对较高的 epoch 的模型。图中显示，在第 2 个 epoch 时，验证损失达到最低，而 F1 得分也相对较高，因此我们选择了在第 2 个 epoch 训练结束后的模型参数。



三、分析讨论

3.1 模型对比

将上面三种模型参数量以及测试结果进行对比，如下图所示：

System	# params	Accuracy	Precision	Recall	F1
SVM		0.90	0.87	0.86	0.87
Logistic		0.85	**0.90**	0.83	0.86
$BERT_{base}$	110M	**0.92**	0.88	**0.89**	**0.88**

接下来我们对对比的结果进行分析与讨论，我们首先提出结论，之后对其进行解释：

3.1.1 SVM 的表现略优于 Logistic 回归

- **特征提取方法的适应性：**报告指出，您使用了 TF-IDF 作为特征提取方法。TF-IDF 倾向于强调在特定文档中重要但在整个语料库中不常见的词语。这种方法在捕捉独特和有区分性的情感词汇方面效果显著，而 SVM 在处理这样的高维、稀疏数据时比 Logistic 回归更加有效。SVM 通过最大化数据点与决策边界之间的间隔，能够更好地利用这些特征来进行准确的分类。
- **间隔最大化：**SVM 的核心是寻找最大间隔边界，这意味着它不仅仅是寻找分割两个类别的决策边界，而是寻找能够以最大间隔分开两个类别的边界。这种最大间隔原则使 SVM 在某些情况下比逻辑回归更鲁棒。

3.1.2 BERT 的表现优于传统机器学习模型

- **深度和复杂度：**BERT 是一个基于 Transformers 的深度学习模型，具有大量的层和参数（110M）。这种结构使得 BERT 能够捕获和理解语言中更复杂和深层次的模式，特别是在处理具有复杂语义和句法结构的自然语言文本时。相比之下，SVM 和 Logistic 回归是更简单的模型，它们可能无法捕获文本数据中的所有细微差别和复杂关系。
- **上下文理解：**BERT 通过预训练和微调的范式，在大型未标记文本数据集上学习语言的通用特征。这让 BERT 能够理解单词、短语和句子的复杂关系，尤其是在理解上下文方面。例如，在情感分析任务中，BERT 可以更准确地理解词汇在特定上下文中的含义，而传统机器学习方法如 SVM 和 Logistic 回归则更多地依赖于手工提取的特征，可能无法充分利用上下文信息

3.2 模型可解释性研究

和其他深度模型一样，BERT 也是一个“黑盒模型”。尽管 BERT 的结构和训练过程是已知的，但是其内部的多层次的自注意力和前馈网络的复杂交互使得理解模型为何做出特定预测或产生特定输出变得非常困难。

在传统的软件工程中，系统通常是由人类编写的明确规则组成，这些规则的执行路径清晰明了。相反，BERT 等深度学习模型通过在海量数据上的训练学习到如何执行任务，其决策过程涉及高度非线性的计算和数以万计的隐藏层神经元活动，这些活动很难被分解解释。因此，尽管我们可以观察到输入数据和输出结果，但中间的处理过程——模型如何从输入得出输出——往往不够透明。

为了深入了解我们微调得到的 BERT 是如何进行情感分类的，我们使用了两种可视化工具(BertViz 和 SHAP)，来揭示模型在处理具体输入时的内部决策过程。SHAP（SHapley Additive exPlanations）通过量化每个特征对模型预测的贡献度来提供全局解释，而 BertViz 则专注于局部解释，即可视化模型内部的注意力机制，展现模型在预测时重点关注的区域。使用这些工具，我们可以更好地理解模型的

行为，比如它是如何区分正面和负面情绪的，哪些词汇和语句结构对预测结果影响最大等。这种分析不仅帮助我们验证模型的有效性，也能够在一定程度上增加模型决策的透明度，提高人类对模型预测的信任。

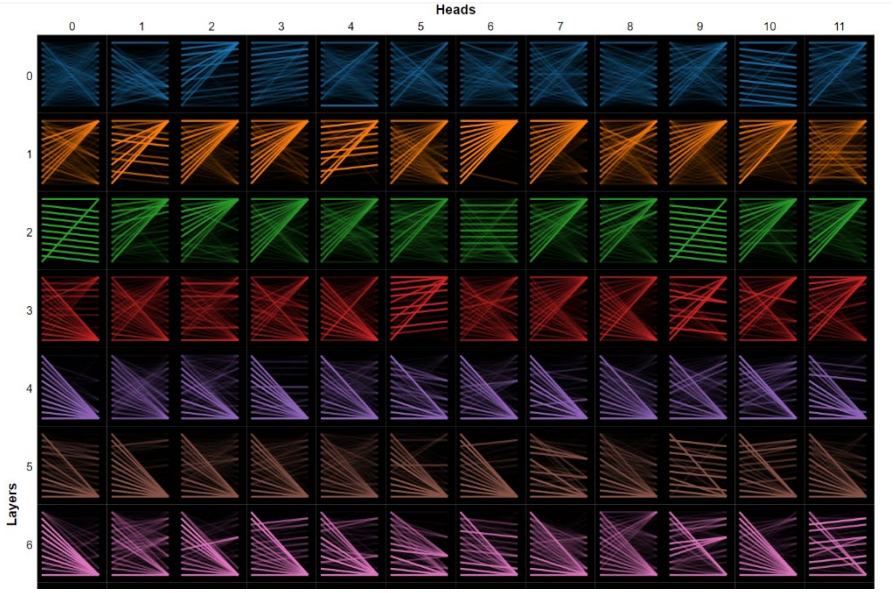
4.1 使用 BertViz 进行注意力可视化

BertViz 是一个专门设计来揭示 BERT 等基于 Transformer 的模型内部工作机制的可视化工具。它详细地展现了模型中的注意力机制，即模型如何分配对输入句子中每个 token 的关注。在 BERT 模型的每个自注意力层中，对于输入序列中的每个 token，模型都会生成一系列的查询（Q）、键（K）和值（V）向量。通过计算查询向量与所有键向量的点积，然后应用 softmax 函数，模型得到了一个注意力权重分布，这个分布反映了每个 token 在考虑当前 token 时应得到的关注程度。BertViz 对每一个多头自注意力中的注意力权重进行可视化，通过绘制从一个 token 到另一个 token 的线条来展示这些注意力权重。线条颜色的深浅对应于权重的大小，表明模型在考虑当前 token 时对序列中其他 token 的关注程度。

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

我们通过观察大量的例子总结了 BERT 的一些规律，下面通过一个具体的例子进行说明。

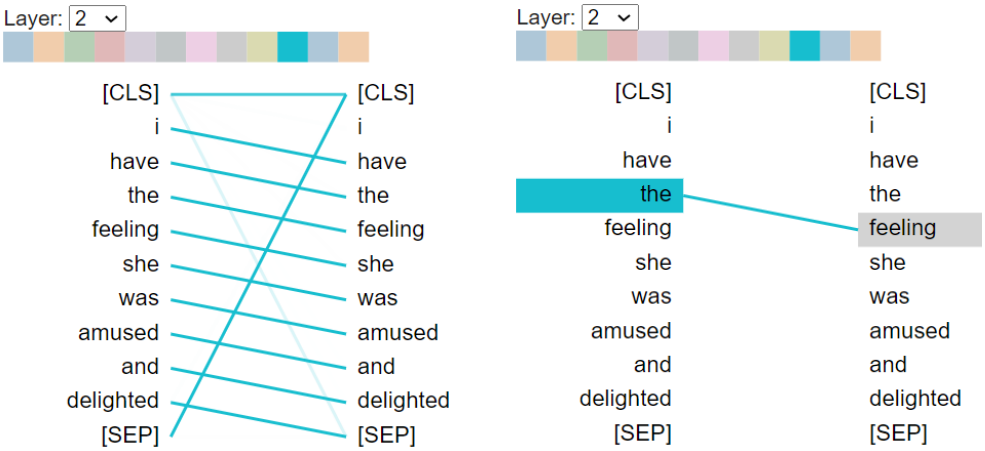
给模型输入一个句子“i have the feeling she was amused and delighted”，BERT 所有层的每一个多头注意力分布图如下所示（这里只截取了前 7 层的视图）：



可以发现，不同层和不同头的注意力分配模式各不相同，表明 BERT 学习到丰富多样的注意力模式，之后我们对这些注意力模式逐一进行分析，并总结出了几种常见的模式。

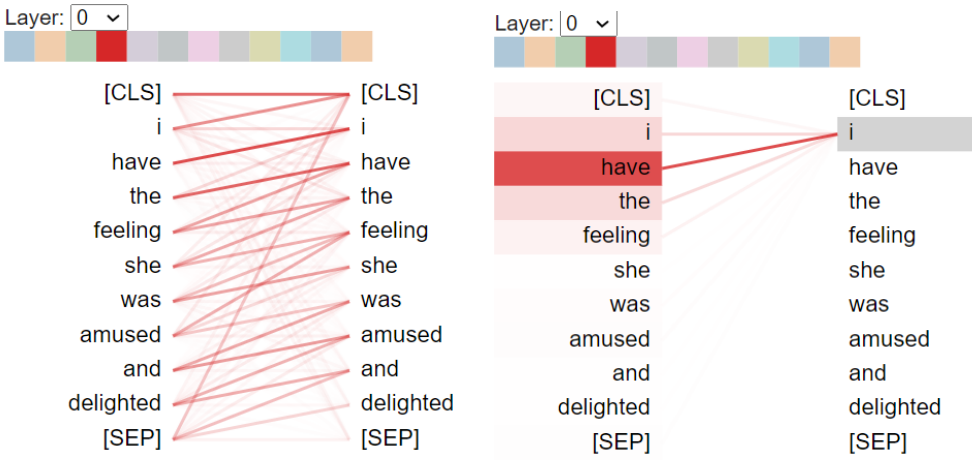
1. Next-word attention patterns

在这种模式中，特定位置的大部分注意力都指向序列中的下一个词。下面的注意力图属于第 3 层，第 1 个头。可以看到，几乎所有左侧词的注意力都指向了序列中的下一个词。



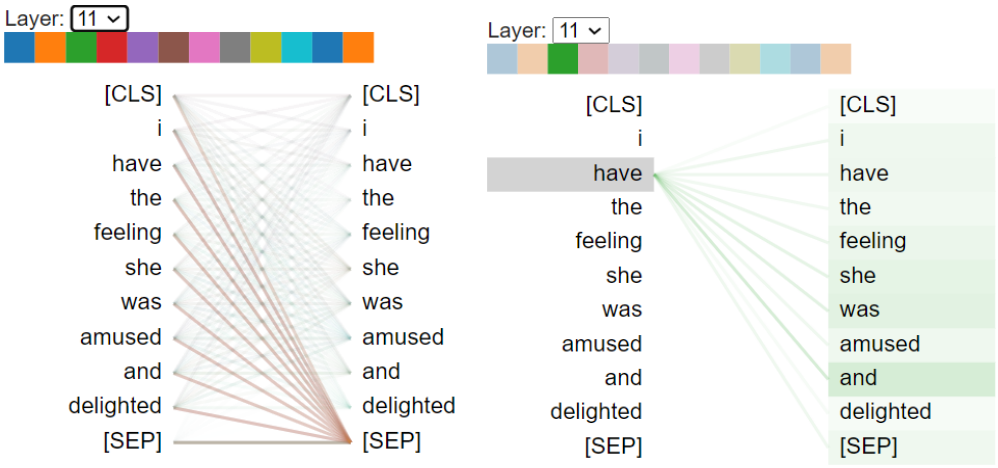
2. Previous-word attention patterns

与模式 1 正好相反，在这种模式中，特定位置的大部分注意力都指向序列中的前一个词。下面的注意力图属于第 1 层，第 4 个头。可以看到，几乎所有左侧词的注意力都指向了序列中的前一个词。



3. Bag of Words attention pattern

在这种模式中，左侧的每一个单词都会与右侧的所有词相联系。这一模式更多的出现在模型的最后几层，倾向于将注意力分散到多个词上。可以推测模型正在尝试理解整个句子的高级语义关系，因为在模型的较高层，注意力机制更加关注于整体的语境和句子的深层含义。



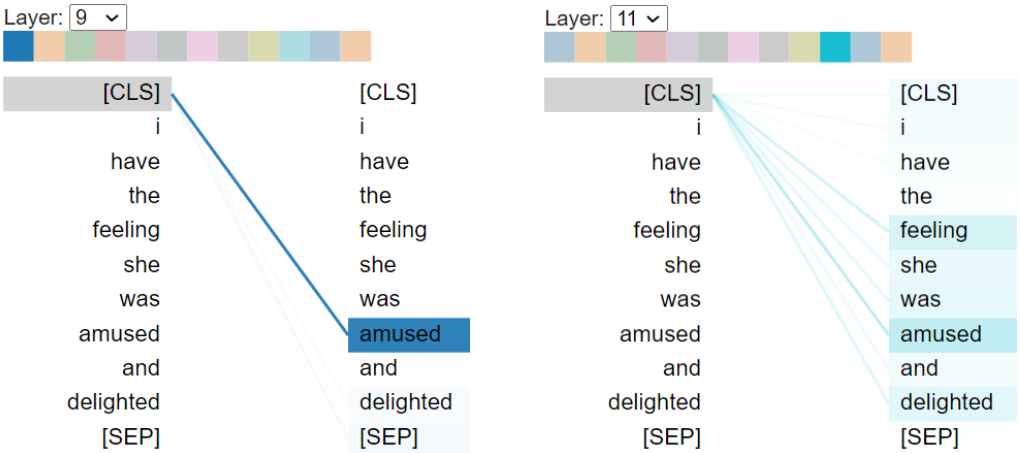
4. [CLS] token attention pattern

在句子的情感分类中，BERT 利用[CLS]标记来捕获整个句子的上下文信息，并将[CLS]标记的最终隐藏状态传递给一个简单的分类器，该分类器会根据这个汇总的句子表示来预测句子的情感标签。因此，在这一模式中，我们重点关注[CLS]是如何被得到的。

我们选取了两张典型的注意力图。在左图中，CLS 标记的注意力聚焦在特定的词“amused”上。“amused”这个词对于整个句子的情感倾向是非常重要的。这种注意力模式反映了模型如何识别关键词汇，而这些词汇对于理解整个句子的情感来说是至关重要的。

在右图中，注意力分布显示了在模型的较高层，CLS 标记的注意力分布在序列中的多个词上。可以发现其对 feeling, amused 和 delighted 赋予了较高的权重，这表明这些词对情感的分析有这更重要的作用，但同时模型也考虑了其他词汇。

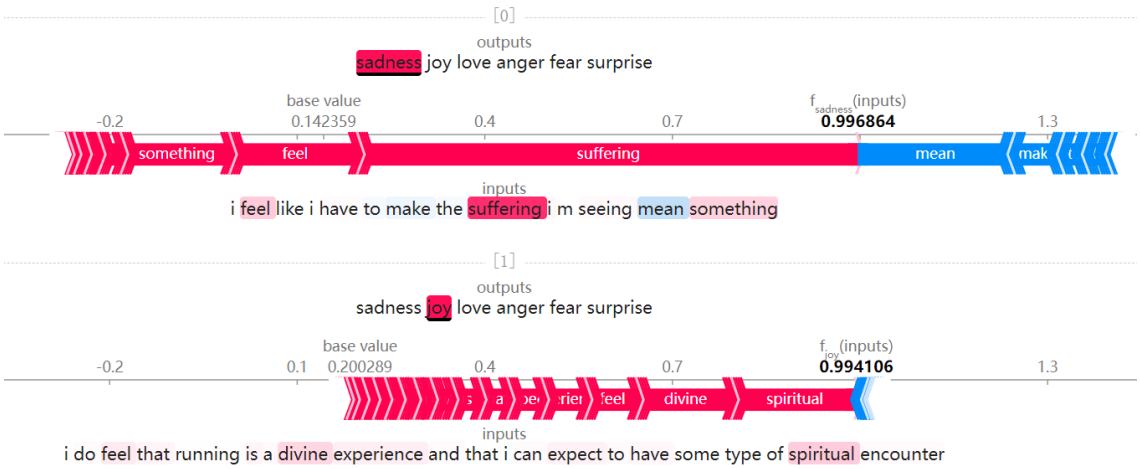
根据上述的分析，我们可以得出 BERT 不仅仅是在考虑单个词汇，而是在更高的层次上综合考虑了整个句子的上下文，从而对句子进行情感分类。



4.2 使用 SHAP 进行可视化

下图展示了使用 SHAP（SHapley Additive exPlanations）对两个不同文本输入进行情感分析的结果。SHAP 是一种解释模型预测的工具，它通过计算每个特征对模型输出的贡献来提供解释。在图中，红色条表示负面贡献（即使模型输出的分数降低），蓝色条表示正面贡献（即使模型输出的分数升高）。条的长度表示贡献的大小。

对于第一个例子，“suffering”一词对于“悲伤”情感标签有很大的正面贡献，而其他单词对其他情感标签的贡献较小。在第二个例子中，“divine”和“spiritual”两个词对“喜悦”标签有很大的正面贡献。这些可视化帮助我们理解模型为何会给出特定的情感预测，提高了模型的透明度和可解释性。



参考文献

[1] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.

[2] Medhat, Walaa, Ahmed Hassan, and Hoda Korashy. "Sentiment analysis algorithms and applications: A survey." Ain Shams engineering journal 5.4 (2014): 1093-1113.

[3] Vig, Jesse. "BertViz: A tool for visualizing multihead self-attention in the BERT model." ICLR workshop: Debugging machine learning models. Vol. 23. 2019.

[4] Kokalj, Enja, et al. "BERT meets shapley: Extending SHAP explanations to transformer-based classifiers." Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation. 2021.