关于LSTM模型与Transformer模型在知识蒸馏中的对比
——以文本分类任务为例

黄奕铭 刘佳琦 贾卓霖 孙海川 杨昆山

2022 年 12 月 4 日

摘要

目录

1 背景介绍

1.1 文本分类任务

随着互联网的不断发展,网络上的文本数据日益增多,如果能对这些数据进行有效分类,那么更有利于从中挖掘出有价值的信息,因此文本数据的管理和整合显得十分重要。文本分类是自然语言处理任务中的一项基础性工作,主要应用于舆情检测及新闻文本分类等领域,目的是对文本资源进行整理和归类。

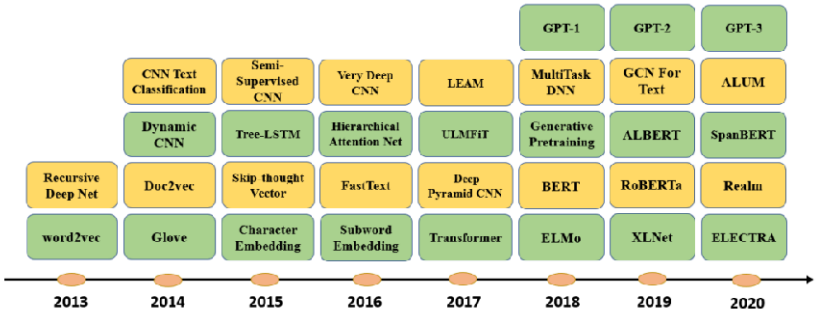

图 1: 目前为止的文本分类方法

1.2 深度学习的可解释性

随着深度学习的广泛应用，人类越来越依赖于大量采用深度学习技术的复杂系统，然而，深度学习模型的黑盒特性对其在关键任务应用中的使用提出了挑战，引发了道德和法律方面的担忧，因此，使它们具有可解释性是使它们令人信服首先要解决的问题。于是，关于可解释的人工智能领域的研究应运而生，主要集中于向人类观察者明确解释模型的决策或行为。因此，为进一步深入研究建立更高效且具有可解释性可以为深度学习模型确立良好的基础。目前研究重点从解释深度学习模型的逻辑规则、决策归因和内部结构表示这三个方面出发介绍了几种可解释性研究的典型模型和算法外还指出了三种常见的内置可解释模型的构建方法；最后，评价方法有忠实度、准确性、鲁棒性和可理解性这四种的评价指标，指明了深度学习可解释性未来可能的发展方向。

1.3 知识蒸馏与大模型微调

高性能的深度学习网络通常是计算型和参数密集型的，难以应用于资源受限的边缘设备.为了能够在低资源设备上运行深度学习模型，需要研发高效的小规模网络.知识蒸馏是获取高效小规模网络的一种新兴方法，其主要思想是将学习能力强的复杂教师模型中的“知识”迁移到简单的学生模型中.同时，它通过神经网络的互学习、自学习等优化策略

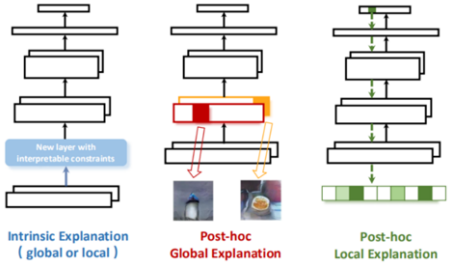

图 2: 可解释方法的大致分类

和无标签、跨模态等数据资源对模型的性能增强也具有显著的效果.基于在模型压缩和模型增强上的优越特性，知识蒸馏已成为深度学习领域的一个研究热点和重点.

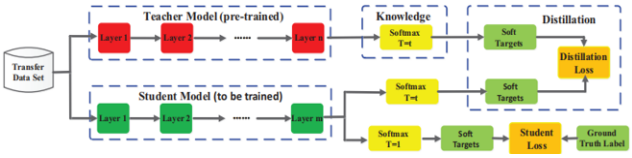

图 3: 知识蒸馏示意图

1.4 目前研究的问题与我们的解决方法

目前，针对蒸馏学习的可解释性方法较少，而且这些工作的基线性能都不是很高。

而对于LSTM和Transformer的比较的工作也不多。

而我们针对这个问题提出面向注意力机制基于实例可视化的可解释方法，并利用Glove预嵌入的词向量公平地比较LSTM和Transformer的性能，示意图如下图所示。

2 相关工作

2.1 自然语言处理中的可解释性

基于深度神经网络的大型预训练语言模型在众多自然语言处理任务上都取得了巨大的成功，如文本分类、阅读理解、机器翻译等，目前已经广泛应用于工业界。然而，这些模型的可解释性普遍较差，即难以理解为何特定的模型结构和预训练方式如此有效，亦无法解释模型做出决策的内在机制，这给人工智能模型的通用化带来不确定性和不可控性。因此，设计合理的方法来解释模型至关重要，它不仅有助于分析模型的行为，也可以指导研究者更好地改进模型。现有工作的主流想法仍是将自然语言模型与自动机等价

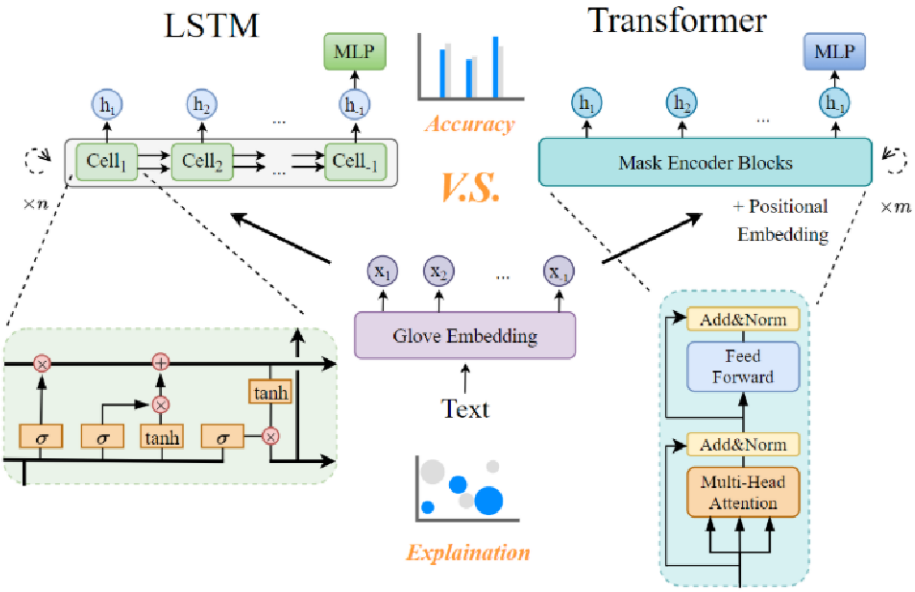

图 4: 我们的方法

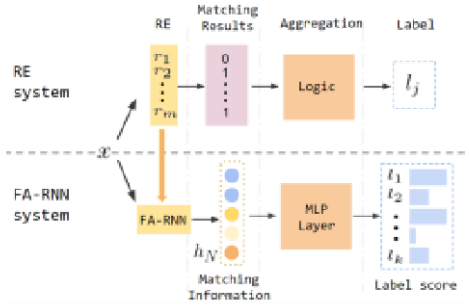

图 5: EMNLP20中以自动机的思路去研究NLP的可解释性的例子[4]

3 方法

3.1 tiny LSTM的性质

遗忘门部分

$$f_t = \sigma\left(U_f h_{t-1} + W_f x_t\right)$$
$$k_t = c_{t-1} \odot f_t$$

输入门部分

$$i_t = \sigma\left(U_i h_{t-1} + W_i x_t\right)$$
$$g_t = \tanh\left(U_g h_{t-1} + W_g x_t\right)$$
$$j_t = g_t \odot i_t$$
$$c_t = j_t + k_t$$

输出门部分

$$o_t = \sigma\left(U_o h_{t-1} + W_o x_t\right)$$
$$h_t = \tanh\left(c_t\right) \odot o_t$$

3.2 tiny Transformer的性质

对于Transformer Block的运算，有

$$Input : Q, K, V$$
$$A = W_Q T \cdot W_K T,$$
$$T = (Softmax(A) W_V T) W_O,$$
$$T = LayerNorm(T)$$
$$T = FFN(T)$$

其中前馈FFN网络过程应该为

$$T = MLP_1(T)$$
$$T = GELU_1(T)$$
$$T = LayerNorm_1(T)$$
$$T = MLP_2(T)$$
$$T = GELU_2(T)$$
$$T = LayerNorm_2(T)$$

3.3 Bert作为教师模型的微调策略

由How to Finetune Bert in Classficication Text Task [5][1]的工作我们可以知道，我们对于 Bert base模型进行如下的分层学习率会取得最佳效果

$$For\ BertLayer_i : LR_i = 0.95^{(12-i)}, i \in \{1, 2, ..., 12\}$$

3.4 有易到难的蒸馏学习策略

我们令损失函数由硬标签损失和软标签损失[2][3]以α为系数进行调和相加，即

$$SoftLabels = Softmax(teacher(Text)/T)$$

$$SoftLogits = Softmax(Text/T)$$

$$HardLabels = Softmax(teacher(Text))$$

$$HardLogits = Softmax(Text)$$

$$DistillLoss = \alpha \cdot XELoss(HLogits, HLabels) + (1 - \alpha) \cdot L2Loss(SLogits, SLabels)$$

示意图如图所示

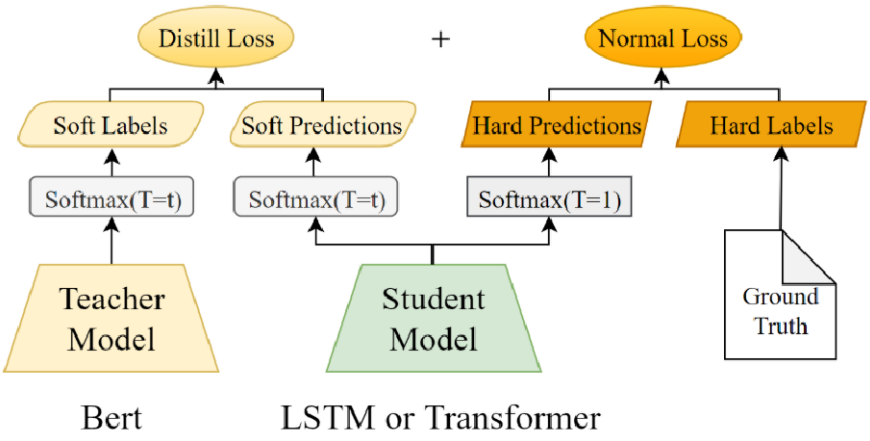

图 6: 我们的训练策略

对于α的变化，我们取初始值为0.8，令其随轮次变化，我们一共训练10轮

$$\alpha_i = 0.8 - 0.1 \cdot (i - 1), i \in \{1, 2, ..., 8\}$$

$$\alpha_i = 0, i \in \{9, 10\}$$

3.5 Bert注意力可视化

对于注意力机制，显然token对token的注意力图

$$A = Softmax(\frac{QW_Q \cdot (KW_K)^T}{\sqrt{dim}})$$

将其可视化显然会有较好的结果

4 实验

4.1 超参设置

对于Bert的微调，我们有

表 1: Bert微调配置

配置	实际选择
优化器	Adam
平均移动参数(β_1, β_2)	(0.9, 0.999)
head学习率	1e-5
主干学习率衰减系数	0.95
批次大小	imdb:4 和 sst2: 16
训练轮数	10
训练步数	imdb:6000 和 sst2:16000
损失函数	交叉熵损失

对于两种模型的对比训练，我们有

4.2 bert的微调结果

在sst2上, 微调bert可见损失减少最终的正确率约为93.8
详细数据如下表所示
在imdb上，微调bert可见损失减少最终的正确率约为92.8
详细数据如下表所示所示

4.3 Tiny LSTM和Tiny Transformer的对比训练结果

在tensorboard可视化后的训练曲线如下：分别为LSTM和Transformer在sst2数据集上
在imdb数据集上(都为一层)，一层的Transformer过拟合了。

表 2: Tiny LSTM 和 Tiny Transformer的训练配置

配置	实际选择
优化器	Adam
平均移动参数(β_1, β_2)	(0.9, 0.999)
学习率	1e-5
批次大小	imdb:4 和 sst2: 16
训练轮数	10
训练步数	imdb:6000 和 sst2:16000
损失函数	交叉熵损失
软标签损失占比	0.8
软标签占比衰减	0.1

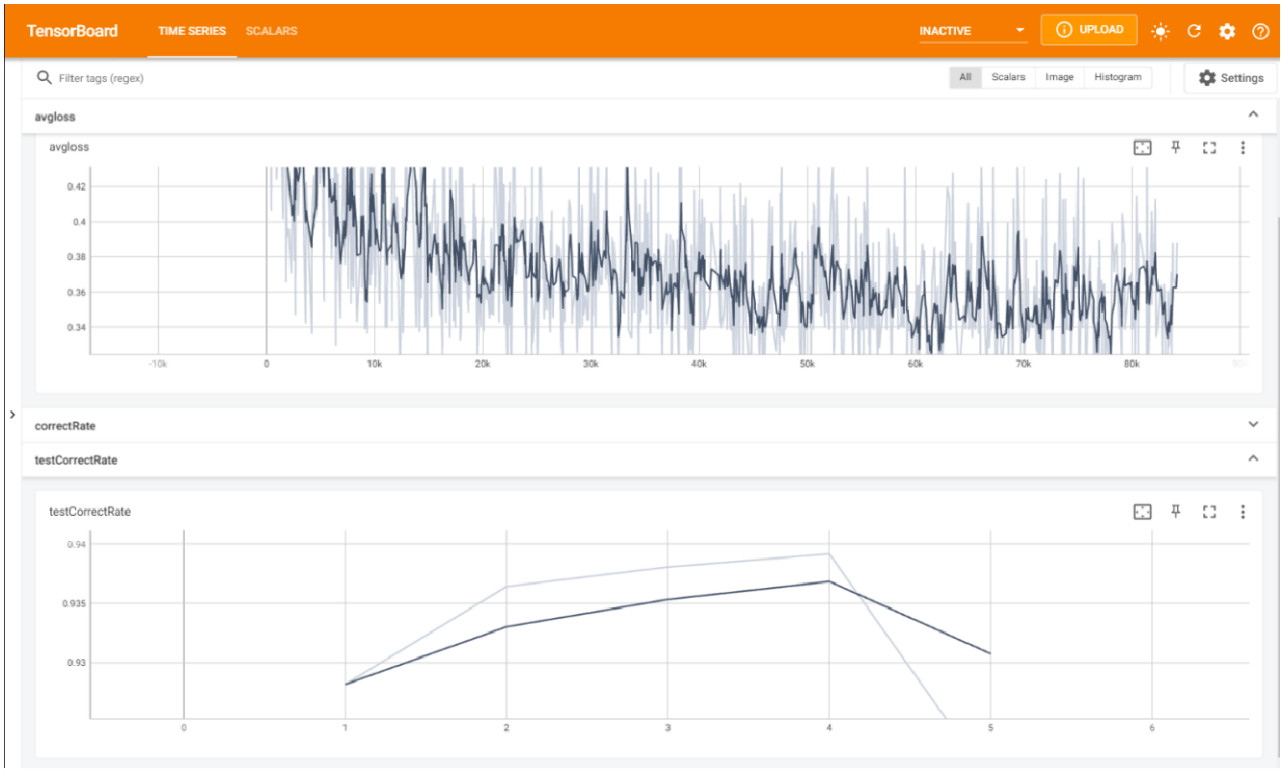

图 7: bert在sst2上的微调，loss与测试集正确率

表 3: bert在sst2上的微调

训练轮数	1	2	3	4	5
平均损失	0.411	0.392	0.378	0.369	0.355
测试集上准确率	92.82	93.33	93.58	93.87	93.2

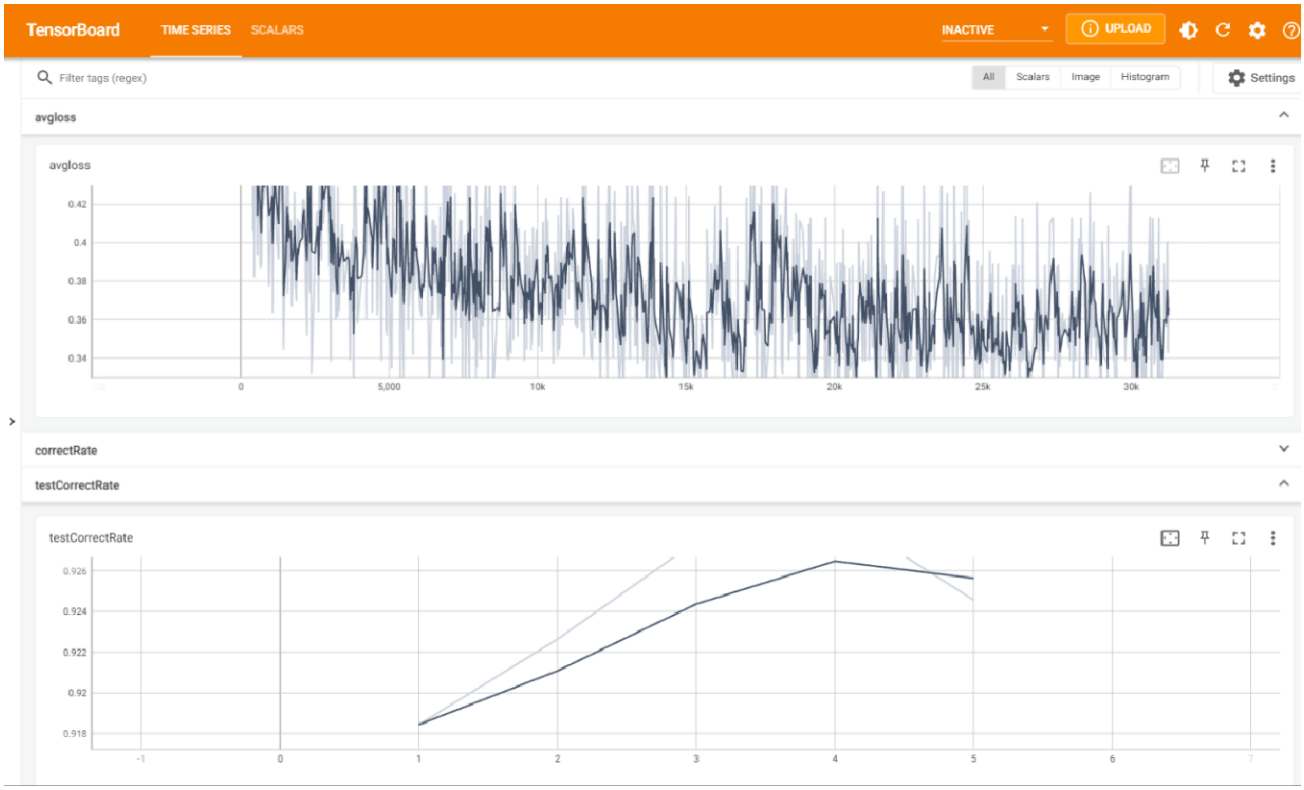

图 8: bert在imdb上的微调，loss与测试集正确率

表 4: bert在imdb上的微调

训练轮数	1	2	3	4	5
平均损失	0.428	0.398	0.388	0.374	0.365
测试集上准确率	91.92	92.11	92.43	92.81	92.52

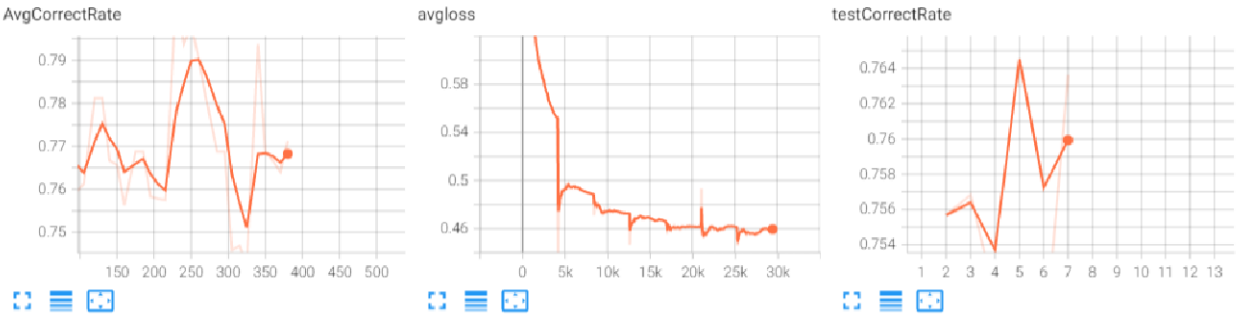

图 9: 一层LSTM在sst2上的微调，loss、测试集正确率和训练集准确率

表 5: 一层LSTM在sst2训练

训练轮数	1	2	3	4	5	6	7
平均损失	0.581	0.562	0.493	0.487	0.471	0.462	0.454
训练集上准确率	76.5	77.5	76.5	76.8	76.0	78.9	83.1
测试集上准确率	-	75.6	75.65	75.83	76.41	75.72	75.92

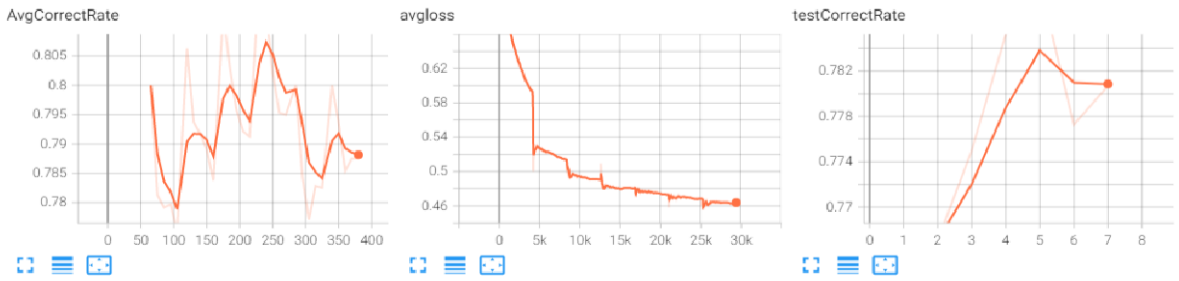

图 10: bert在imdb上的微调，loss、测试集正确率和训练集准确率

表 6: 一层Transformer在sst2训练

训练轮数	1	2	3	4	5	6	7
平均损失	0.621	0.592	0.533	0.497	0.481	0.474	0.464
训练集上准确率	80.01	78.5	79.0	78.85	79.8	80.59	79.1
测试集上准确率	-	75.6	75.65	75.83	76.41	75.72	75.92

表 7: 一层LSTM在IMDB上训练

训练轮数	1	2	3	4	5	6	7
平均损失	0.681	0.622	0.583	0.577	0.631	0.584	0.564
训练集上准确率	86	88.5	89.0	90.85	84.8	74.59	76.1
测试集上准确率	-	69.6	70.65	70.83	72.41	73.72	74.92

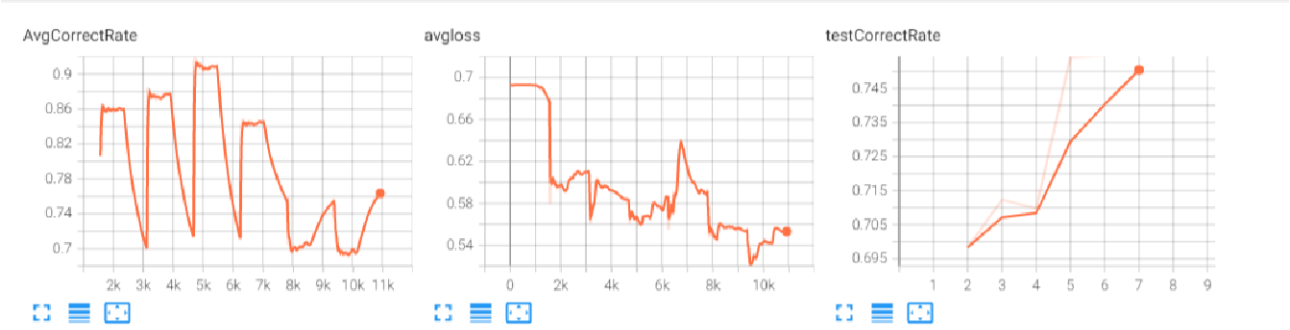

图 11: 一层LSTM在imdb上的微调，loss、测试集正确率和训练集准确率

数据集	模型		
	Bert	LSTM	Transformer
SST2	92.8	76.5	78.9
IMDB	93.8	75.1	50.0

表 8: 对比训练的结果

Bert,一层的LSTM和Transformer的性能对比如下
我们的模型推理结果如下

show

	Sample	Predict	Ground Truth	Correct
0	hide new secretions from the parental units	0	0	True
1	contains no wit , only labored gags	0	0	True
2	that loves its characters and communicates som...	1	1	True
3	remains utterly satisfied to remain the same t...	1	0	False
4	on the worst revenge-of-the-nerds clichés the ...	0	0	True
5	that 's far too tragic to merit such superfici...	1	0	False
6	demonstrates that the director of such hollywo...	1	1	True
7	of saucy	1	1	True
8	a depressed fifteen-year-old 's suicidal poetry	0	0	True
9	are more deeply thought through than in most `...	1	1	True
10	goes to absurd lengths	0	0	True

图 12: 推理示意

5 分析

我们可视化了bert的注意力，可以看到最后一层对“important”这个词有较大的注意力。

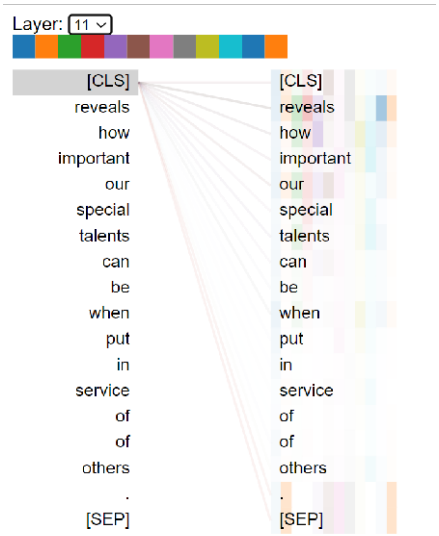

图 13: bert最后一层的可视化

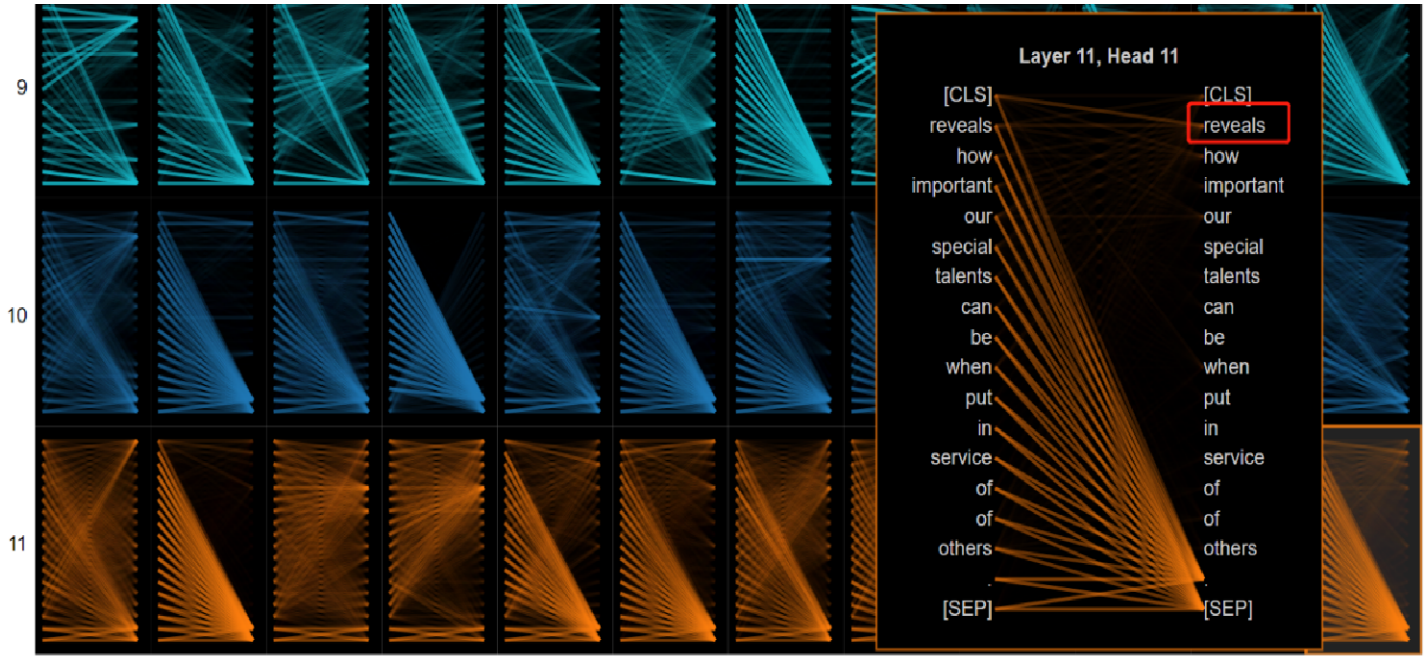

图 14: bert各层的head可视化

6 结论

我们的工作成功微调了Bert并进行了可视化，也成功的比较了一层的LSTM在Transformer上的性能。

参考文献

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[2] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021.

[3] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

[4] Chengyue Jiang, Yinggong Zhao, Shanbo Chu, Libin Shen, and Kewei Tu. Cold-start and interpretability: Turning regular expressions into trainable recurrent neural networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3193–3207, 2020.

[5] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification? In *China national conference on Chinese computational linguistics*, pages 194–206. Springer, 2019.