修改说明

尊敬的各位评审老师：

感谢各位老师提出的宝贵意见，对我很有启发和帮助。根据相关问题和意见，我们对论文进行了认真修改，并在正文中采用了蓝色标注。具体修改说明如下：

1. 针对第一审稿人专家初审意见(1)：”图片模糊，看不清，放大后失真。”

感谢评审老师给出的建议，本文重新绘制了文中图 1 至图 7，提高了图像分辨率。

2 针对第一审稿人专家初审意见(2)：“线性序列转换交代不清楚，$\mathcal{F}_t$具体的操作是什么？和论文中提到的词序转换有区别？符号解释不清楚，$\boldsymbol{x}_i^{(0,j)}$和$\boldsymbol{x}_i(0,j)$是否一样？”

根据评审老师的建议，“线性序列转换$\mathcal{F}_t$”重命名为“token 序列转换$\mathcal{F}_t$”，$\mathcal{F}_t$是 token 序列转换函数，就是实现词序转换。其中，$\boldsymbol{x}_i^{(0,j)}$和$\boldsymbol{x}_i(0,j)$是一样的，在正文中表述时写错了。现根据评审老师意见已在正文修改，使用蓝色标注，具体如下：

(1) token 序列转换$\mathcal{F}_t$ 在 Transformer 中，多头注意力大量使用全连接层，矩阵运算量大。本文提出 token 序列转换$\mathcal{F}_t$方法,该方法对 token 序列重排。给定初始 token 序列表示$\boldsymbol{x}_i^0=(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m)$，在每层语义感知机 SP 中，利用$\mathcal{F}_t$对$\boldsymbol{x}_i^0$进行 token 序列转换，如式(1)所示。

$$\begin{aligned}\boldsymbol{x}_i^{(0,j)}&=\mathcal{F}_t(\boldsymbol{x}_i^0)\\&=(\boldsymbol{x}_{j1},\boldsymbol{x}_{j2},\cdots,\boldsymbol{x}_{jm})\in\mathbb{A}_m^m\end{aligned}\tag{1}$$

其中，$\mathbb{A}_m^m$表示长度为m的 token 序列$(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m)$的全排列集合，$\boldsymbol{x}_i^{(0,j)}=(\boldsymbol{x}_{j1},\boldsymbol{x}_{j2},\cdots,\boldsymbol{x}_{jm})$为$\boldsymbol{x}_i^0$的第$j$次随机排序，即第$j$次 token 序列转换结果。

在训练样本不足时，token 序列转换$\mathcal{F}_t$可以丰富样本的多种词序表征，但这种词序表征可能存在语义歧义，这种歧义将在后续的多层语义感知机中学习消除。

3 针对第一审稿人专家初审意见(3)：“对于单词组成的句子/短语来说，单词的先后顺序不能轻易改变，词序转换会严重破坏句子/短语的语义，甚至转换后的句子/短语不是句子/短语，如何丰富样本的词序特征？”

词序转换或者说 token 序列转换确实会破坏句子、短语的语义，甚至转换后的句子/短语不是句子/短语。但在深度学习中，经多层非线性特征变换后，其特征也存在语义丢失现象。即便 Transformer、BERT 等模型，也仅在模型输入时添加一次位置编码，并未让特

征始终携带位置信息，不关注位置信息对特征的影响，在后续的变换中，没有对抗可能产生语义歧义的举措。而本文通过后续的多层语义感知机来学习消除这种语义歧义。具体地，如文中(2)多层语义感知机所述：

第j层语义感知机输出$\boldsymbol{x}_i^j$如式(2)所示：

$$\boldsymbol{x}_i^j = \mathrm{Norm}(-\mathcal{F}_t(\boldsymbol{x}_i^0) + \boldsymbol{x}_i^{j-1} + \mathcal{F}_m(\boldsymbol{x}_i^{j-1}, \boldsymbol{\theta}^j)) \tag{2}$$

其中，$\mathrm{Norm}(\cdot)$表示归一化，$\mathcal{F}_t(\boldsymbol{x}_i^0)$为序列转换结果，$\boldsymbol{x}_i^0$为初始 token 序列特征表示，$\boldsymbol{x}_i^{j-1}$为第$j-1$层语义感知机输出，$\mathcal{F}_m(\boldsymbol{x}_i^{j-1}, \boldsymbol{\theta}^j)$表示特征$\boldsymbol{x}_i^{j-1}$经第$j$层语义感知机中的 MLP 学习后的特征表示，$\boldsymbol{\theta}^j$为第$j$个 MLP 的参数。

包含$\mathcal{F}_t$函数的语义感知机能够感知 token 的位置信息，并学习消除 token 序列转换$\mathcal{F}_t$生成的错误语义。如式(1)所示，每次经过$\mathcal{F}_t$转换，样本$\boldsymbol{x}_i$的 token 序列会发生变化，同一个 token 的位置会不同，即词向量不同。这样丰富了样本的特征表达，但生成的 token 序列由于位置的变化，有可能产生语义错误，故式(2)中，在$\mathcal{F}_t(\boldsymbol{x}_i^0)$前加负号进行学习消减。然后与上一层的语义感知机输出$\boldsymbol{x}_i^{j-1}$、第j层的 MLP 输出$\mathcal{F}_m(\boldsymbol{x}_i^{j-1}, \boldsymbol{\theta}^j)$相加，最后归一化输出$\boldsymbol{x}_i^j$为第$j$层语义感知机的学习的特征表示。经过$M$层语义感知机的学习，最终输出的$\boldsymbol{x}_i^M$为样本$\boldsymbol{x}_i$的特征表示。

4　针对第一审稿人专家初审意见(4)：“论文中提到的 DMSP 采用的是集成学习的思想，虽然和 Boosting 思想不同，但和 Stacking 思想相似。且集成学习需要训练多个基分类器，甚至基分类器之间的训练存在一定的依赖关系。本文前面的出发点是减少模型的参数，降低模型的复杂度，但采用集成学习会增加模型的复杂度，训练时间变长，和本文的出发点冲突。”

本文的 DMSP 不是嵌入了集成学习算法，而是将每层感知机学到的特征结果，输入到一个 Softmax 分类器中，根据每层的基分类器的分类结果，用于控制模型的深度，其目的是优化模型的深度。最终的样本分类结果，是在已得到的多个基分类器上的分类结果的加权输出。因此，不会额外增加模型的复杂度。

5　针对第一审稿人专家初审意见(5)：“多头注意力的空间复杂度不是 O(tmd),以文中提及的字母进行举例，m 个样本，每个样本的嵌入维度是 d，假定 self-Attention 的个数为 t，那么每个 self-Attention 中 Q,K,V 的权重矩阵的大小为 m*(d/t),t 个头合并后的总大小为 m*d。”.

感谢审稿人指出的错误，本文查阅原论文[1]，并重新计算时间复杂度和空间复杂度，已在正文 1.3 节修改，并用蓝色标注，具体修改如下：

$\mathcal{F}_t$是 token 序列转换函数，本质上是洗牌算法，如式(1)所示，计算简单，时间复杂度为$O(m)$，空间复杂度为$O(d)$。

多头注意力使用大量全连接层，矩阵运算量大。单个自注意力都需要构造$\boldsymbol{Q}_i$、$\boldsymbol{K}_i$和$\boldsymbol{V}_i$矩阵，其计算如式(3)。

$$\boldsymbol{Q}_i=\boldsymbol{X}\boldsymbol{W}_i^{\boldsymbol{Q}},\boldsymbol{K}_i=\boldsymbol{X}\boldsymbol{W}_i^{\boldsymbol{K}},\boldsymbol{V}_i=\boldsymbol{X}\boldsymbol{W}_i^{\boldsymbol{V}} \tag{3}$$

其中，$\boldsymbol{X}\in\mathbb{R}^{m\times d}$是 token 序列特征表示,$\boldsymbol{W}_i^{\boldsymbol{Q}},\boldsymbol{W}_i^{\boldsymbol{K}},\boldsymbol{W}_i^{\boldsymbol{V}}\in\mathbb{R}^{d\times\frac{d}{t}}$，分别是生成$\boldsymbol{Q}_i,\boldsymbol{K}_i,\boldsymbol{V}_i$的权重矩阵，$i=1,\cdots,t$,$t$表示注意力个数。

单个自注意力的计算如式(4)所示。

$$\text{head}_i=\text{Attention}(\boldsymbol{Q}_i,\boldsymbol{K}_i,\boldsymbol{V}_i)=\text{Softmax}\left(\frac{\boldsymbol{Q}_i\boldsymbol{K}_i^{\top}}{\sqrt{d_k}}\right)\boldsymbol{V}_i \tag{4}$$

其中，$\text{head}_i\in\mathbb{R}^{m\times\frac{d}{t}}$表示第$i$个自注意力的输出，$d_k=\dfrac{d}{t}$。

多头注意力的计算如式(5)。

$$\text{MHA}=\text{Cat}(\text{head}_1,\cdots,\text{head}_t)\boldsymbol{W}^o \tag{5}$$

其中，$\text{MHA}\in\mathbb{R}^{m\times d}$表示多头注意力的输出，$\text{Cat}(\text{head}_1,\cdots,\text{head}_t)$表示将$\text{head}_1$至$\text{head}_t$拼接，$\boldsymbol{W}^o\in\mathbb{R}^{d\times d}$表示全连接层权重，将拼接结果做线性变换。

在深度模型中，通常使用模型的参数量(Parameters)衡量模型空间复杂度，用浮点运算数(Floating Point Operations, FLOPs)衡量模型时间复杂度。在多头注意力中，每个头的参数量为$3d\times\dfrac{d}{t}$，FLOPs 为$(2d-1)\times\dfrac{d}{t}$，则t个头总参数量为$3d^2$，FLOPs 为$(2d-1)\times d$，最后一个全连接层参数量为d^2，FLOPs 为$(2d-1)\times d$。则，对头数为t的多头注意力，其参数量为$4d^2$，空间复杂度为$O(d^2)$，其 FLOPs 为$(2d-1)\times(d+1)$，则其时间复杂度可表示为$O(d^2)$。

token 序列转换$\mathcal{F}_t$和多头注意力的时间复杂度和空间复杂度对比如表 1 所示。

表 1：token 序列转换$\mathcal{F}_t$和多头注意力的时间复杂度和空间复杂度对比

方法	时间复杂度	空间复杂度
token 序列转换$\mathcal{F}_t$	$O(m)$	$O(d)$
多头注意力	$O(d^2)$	$O(d^2)$

从表 1 可以看出，从时间复杂度和空间复杂度分析，token 序列转换$\mathcal{F}_t$明显优于多头注意力，因此，本文提出的多层语义感知机模型对比 Transformer 有更好的时间性能和空间性能。

其次，对文中式(14)做如下修改：

$$\tau = d^2 + d + d \times C + C \tag{14}$$

因为对输入输出维度均为d的全连接层，其隐藏层权重$\boldsymbol{W} \in \mathbb{R}^{d^2}$，偏置$\boldsymbol{b} \in \mathbb{R}^{d}$。而对输入维度为$d$输出维度为$C$的全连接层，其隐藏层权重$\boldsymbol{W} \in \mathbb{R}^{d \times C}$，偏置$\boldsymbol{b} \in \mathbb{R}^{C}$。所以$\tau = d^2 + d + d \times C + C$。

6　针对第一审稿人专家初审意见(6)：“实验中看到模型的效果比 baseline 效果好，模型参数量也有一定的优势，但未交待不同模型的训练时间。“

感谢审稿人给出的建议。在深度模型中，通常使用模型的参数量(Parameters)衡量模型空间复杂度，用浮点运算数(Floating Point Operations, FLOPs)衡量模型时间复杂度。为对比模型时间性能，本文在实验部分增加了 FLOPs 统计分析，在表 13-表 16 中增加了一列“FLOPs”数据，在正文中使用蓝色标注，具体如下。

表 13　在 Ama.4 数据集上各种模型的复杂度及分类结果比较

Model	Model Complexity				Classification Evaluation	
	IDepth	FDepth	Parameters(M)	FLOPs(G)	Accuracy	Macro-F1
Transformer	1	/	0.16	1.00	83.20%	84.29%
MLP	4	/	0.27	1.73	88.50%	88.44%
SMSP*	10	/	0.38	2.41	**91.62%**	**91.37%**
SMSP-E	5	/	0.37	2.36	87.91%	87.88%
DC*	5	4	0.30	1.89	**92.05%**	**91.93%**
DMSP*	5	4	0.30	1.89	**90.10%**	**90.17%**

表 14　在 AGN.4 数据集上各种模型的复杂度及分类结果比较

Model	Model Complexity				Classification Evaluation	
	IDepth	FDepth	Parameters(M)	FLOPs(G)	Accuracy	Macro-F1
Transformer	1	/	0.16	1.00	68.18%	68.50%
MLP	3	/	0.14	0.89	77.25%	77.21%
SMSP*	12	/	0.46	2.88	**82.85%**	**81.33%**
SMSP-E	3	/	0.22	1.42	79.03%	78.79%
DC*	5	4	0.30	1.89	**80.42%**	**80.39%**
DMSP*	5	3	0.22	1.42	**79.75%**	**79.78%**

表 15　在 So.4 数据集上各种模型的复杂度及分类结果比较

Model	Model Complexity				Classification Evaluation	
	IDepth	FDepth	Parameters(M)	FLOPs(G)	Accuracy	Macro-F1
Transformer	1	/	0.16	1.00	80.98%	80.90%
MLP	1	/	0.15	0.94	86.69%	86.64%
SMSP*	12	/	0.46	2.88	**92.51%**	**92.34%**
SMSP-E	3	/	0.22	1.42	**87.50%**	**87.52%**
DC*	5	3	0.22	1.42	81.45%	81.39%
DMSP*	5	5	0.23	1.44	**91.47%**	**91.75%**

表 16　在 20ngp.4 数据集上各种模型的复杂度及分类结果比较

Model	Model Complexity				Classification Evaluation	
	IDepth	FDepth	Parameters(M)	FLOPs(G)	Accuracy	Macro-F1
Transformer	1	/	0.16	1.00	73.09%	73.11%
MLP	2	/	0.15	0.94	77.98%	77.91%
SMSP*	8	/	0.31	1.94	79.93%	79.78%
SMSP-E	3	/	0.22	1.42	**83.89%**	**83.64%**
DC*	5	3	0.22	1.42	**82.11%**	**82.25%**
DMSP*	5	5	0.23	1.44	**84.86%**	**84.87%**

7　针对第二审稿人初审提问(1)：“方法中利用每层句子词序变换来丰富词序表征，但当句子序列较长时，位置信息会更加复杂，模型是否可以处理这种情况，是否需要更深的模型深度（更大的参数量）满足其获取足够多的位置信息，从而保证模型效果？“和第二审稿人专家初审修改意见(2)：“实验设置中句子长度最大为 64，是否可以观察一下句子长度更长时模型的表现。“

token 序列转换不需要额外保存位置信息，因此当句子序列较长时，不需要更深的模型深度，也就不会导致更大的参数量。根据审稿专家的建议，将句子最大长度分别设置为 64、128 和 256，观察所提出的 SMSP 和 DMSP 模型的表现，在 AGN.4 数据集实验上的对比实验结果如图 1 和图 2 所示。

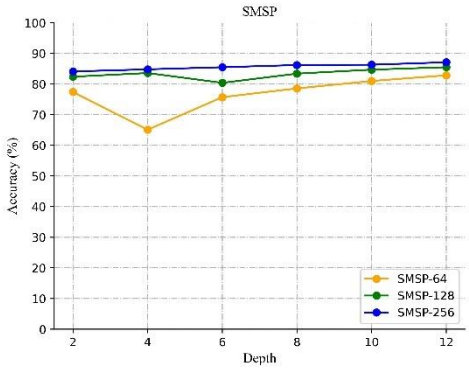

图 1：不同序列长度的 SMSP 模型 Accuracy 随深度的变化

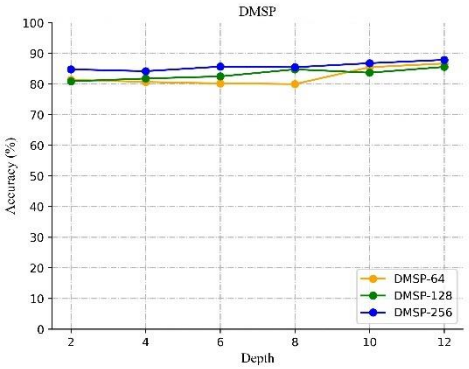

图 2：不同序列长度的 DMSP 模型 Accuracy 随深度的变化

从图 1 和图 2 可以看出，当句子最大序列长度增大时，要保证模型的正确率，并不需要增加模型深度。如图 1 所示，同样深度下，比较最大句子长度分别是 256、128 和 64 的 SMSP 模型的正确率，SMSP-256 高于 SMSP-128，SMSP-128 高于 SMSP-64，也就是说，要保证 SMSP 模型的正确率，对于较长的句子序列，不需要增加模型深度，也就不会增加参数量。如图 2 所示的 DSMP 模型的也是类似的结果，当深度为 2~8 时，相同深度下，DMSP-256 的正确率大于 DMSP-128，DMSP-128 的正确率大于 DMSP-64，深度为 10 和 12 时，DMSP-256 的正确率最大，DMSP-128 与 DMSP-64 相差不大。因此，当句子序列较长时，不需要更深的深度来保证模型效果。

8　针对第二审稿人初审提问(2)：“作者强调方法优势之一在于参数量小、模型复杂度低，尽管单层的参数量相比于 Transformer 有所降低，但根据实验结果该模型的有效性需要一定的深度来保证，反而计算更为简单的 MLP 不需要，因此设计模型的意义和模型的实际效果是否产生了冲突？“

在规模较小的数据集上，MLP 模型容易过拟合。浅层（即层数少）的 MLP 模型虽然在参数量上优于 SMSP 模型，但其性能却比 SMSP 差。由于 SMSP 模型单层参数量极少，通过增加层数，以相对较少的参数量增加为代价，可以获得更高的模型性能，在后面的实验中已得以验证。

9　针对第二审稿人专家初审意见提问(3)：“作者针对每个数据集都只保留了 4 个类别，而没有评估在原始数据集全部类别上的分类效果，这样做的原因是什么？”、针对第二审稿人专家初审意见提问(4)：“对于数据集的阐述有点模糊，没有引用数据集来源，在小规模数据集的前提下划分出 15%作为测试集，这样的评估是否精确？为什么不直接使用

原始数据集给定的测试集？”、第二审稿人专家初审修改意见(3)：“给出每个数据集都只保留了 4 个类别的原因或者补充评估在原数据集分类类别下的分类效果。”和第二审稿人专家初审修改意见(4)：“补充一下数据集的引用链接，以及如何处理得到的现有数据集。”

感谢审稿人批评指正，已在正文中标引了数据集来源。

首先，针对“为什么数据集使用 4 个类而不是全部类别的全部样本“这一问题做出解释：使用原始数据集全部类别的全部样本，Transformer 和本文所提方法均可有较好表现，这也是目前深度学习方法的制约，需要大量的标注数据。而本文要研究的是在标注训练样本不足时，如何提高深度学习算法的性能。因此，需要在训练集规模较小情况下，验证本文方法的有效性，保留全部类别数据规模比较大，因此每种数据集只保留 4 个类别和部分数据。

其次，针对“数据集按 85：15 划分的评估是否准确?“这一问题做出解释：鉴于本文选取的数据集中，如 Sogou、20NewsGroups 并未划分训练集和测试集，因此统一对数据集中每个类别的数据打散后，按照 85:15 划分训练集和测试集。实验中，训练集和测试集的划分比例也不是完全局限在 85:15 上，也尝试了其它的比例，各种方法的实验对比效果基本是类似的。只要他们使用同一数据集和相同的划分比例，那么各模型在这种情况下的比较是公平的。

最后，本文重新阐述了数据集描述，在正文中已使用蓝色标注，具体如下：

本文选择了 AGNews[1]、Amazon[2]、Sogou[3]、20 news groups[4]四个数据集。其中，Sogou、20 news groups 并未划分训练集和测试集，需要手工划分训练集和测试集。因此统一对数据集中每个类别的数据打散后，对每个类别的数据打散后，随机抽取 85%作为训练数据，剩余组成测试集，分别简记为 AGN.4、Ama.4、So.4 和 20ngp.4。各数据集的数据所属类别及样本量如表 3 所示。

10　针对第二审稿人专家初审修改意见(1)：“目前文中只介绍了图像分类中多头注意力替换的相关工作，如果有针对自然语言处理领域的研究，可以补充作为相关工作简要介绍。”

感谢审稿人给出的意见，但截至本文写作完成时，尚未有自然语言处理领域的相关工

[1] https://huggingface.co/datasets/ag_news
[2] https://huggingface.co/datasets/amazon_us_reviews
[3] https://huggingface.co/datasets/sogou_news
[4] https://huggingface.co/datasets/newsgroup

作的参考文献，这也是本文对此开展了研究原因。

11　针对第二审稿人专家初审修改意见(4)：“实验部分大多数表格没有给较好的效果予以特殊的标记，有的表格特殊标记过多，从表格中直接获取有用信息比较困难，可以修改一下特殊标记。”

感谢审稿人给出的建议，本文重新绘制了表格，删除了冗余标记。

12　针对第二审稿人专家初审修改意见(6)：“文章中存在个别错字情况，比如实验部分最后一段“明显幅超越”，以及一些非专有名词直接使用了英文描述，比如 Encoder，请改正。”

感谢审稿人的批评指正，我们认真修改了全文，修改了错别字，使用中文名替代了非专有名词。例如，Encoder 使用“编码器”替换，SA 使用”自注意力”替换，MHA 使用“多头注意力”替换等。

以上是论文的修改说明，再次感谢审稿专家，谢谢你们对本文提出的宝贵意见！

此致

敬礼!

参考文献

[1]. Vaswani, Ashish, et al. Attention Is All You Need[C].Proceedings of the 31st。International Conference on Neural Information Processing Systems. Red Hook, NY,USA: Curran Associates Inc. 2017: 5998–6008.