https://doi.org/10.1051/jnwpu/20254320388

基于语义感知的变长序列数据预处理框架

王晓东[1], 王继维[1], 钟智昊[2], 杨欢[1], 姚红静[3], 郭阳明[3]

(1.西北工业大学 计算机学院, 陕西 西安 710072; 2.西北工业大学 软件学院, 陕西 西安 710072;
3.西北工业大学 网络空间安全学院, 陕西 西安 710072)

摘 要:深度学习框架处理变长序列时,通常采用填充(padding)或截断(truncation)的方式,以方便模型批量训练与处理。然而,填充会加剧内存占用,而截断则会使序列丧失原本的语义信息。因此,提出了一种基于语义感知的变长序列预处理框架,该框架利用典型的无监督学习方法,压缩多维度数据并减小信息损失。同时,基于最小化信息损失理论,采用信息熵度量语义丰富度,为变长表示分配权重,并通过语义丰富度进行融合。此外,实验表明该框架的信息损失相较传统的截断嵌入有所降低,所提方法在信息获取方面具有显著优势,在多个文本分类数据集上表现良好。

关 键 词:变长序列;数据预处理;填充;截断;语义信息;最大化信息
中图分类号:TP391.4 **文献标志码**:A **文章编号**:1000-2758(2025)02-0388-10

近年来,深度学习在计算机视觉、语音识别以及自然语言处理(NLP)等领域不断发展,取得了一定的成果。在 NLP 领域,为提升文本的学习效率,开展了大量有关词义捕获的研究。Mikolov 等[1]提出了 Word2Vec 字嵌入体系结构以学习每个单词的特征。Pennington 等[2]提出全局向量(GloVe)用于文本表示的词嵌入。它在一个巨大的语料库上根据周围的单词进行训练,进而得到每个单词的高维特征表示。其他的词嵌入技术还有 FastText[3], Context2vec[4]等。此外,还有一些研究聚焦于用深度学习方法解决下游任务的文本特征学习问题,如基于 CNN[5]、RNN[6]的模型。在传统研究中,CNN 常被用于视觉领域,2015 年 Kim 提出了文本分类模型 TextCNN[7],通过对句子进行一维卷积获得不同抽象层次的语义信息。与 TextCNN 相比,TextRNN 在捕获长语义信息方面效果更好,常以 RNN 变体 LSTM[8]或 GRU[9]作为主要结构。此外,Yang 等[10]提出了分层注意网络(HAN),通过将注意力机制引入 TextRNN 来多层次地学习文本语义信息。

在上述研究中,输入往往是多级序列,长度不一。基于深度学习框架建立的模型,如 RNN、CNN、Transformer[11]等,通常将张量作为输入,并通过 mini-batch 加速计算。这就给序列的直接训练带来了困难[12]。因此,深度学习中的变长序列标准化表示在近年来得到了越来越多的关注。

随着信息技术的成熟、数据集规模的不断增大,模型在整个数据集上的训练成本也越来越高[13]。为了加速计算,研究人员引入了 mini-batch 技术,将数据集划分成一定数量的批量进行训练[14]。现有的深度学习框架,如 TensorFlow[15]、PyTorch[16]等,通常采用填充(padding)或截断(truncation)的方法来处理变长序列,该方法将长序列表示截至固定长度或多次切分,将短序列包装成一个固定的长度。虽然基于填充或截断的方法可以在一定程度上解决变长序列的问题,但这种处理机制要么丢失了原本的语义,要么导致大量的数据冗余。针对上述问题,LSTM、GRU 等模型在传统填充截断的机制上进行改进,采用基于掩码的处理机制以区分填充信息和实际信息,使模型训练只作用于实际数据,而不会处理填充内容。这种机制通过掩码保持了序列的真实长度,并在计算损失的时候去掉填充部分。这种方式可以在一定程度上减少填充机制造成的信息冗余问题,但截断导致的语义丢失问题仍无法解决。

为此,本文提出了一种基于语义感知的自适应数据预处理框架,利用典型的无监督学习方法将不

收稿日期:2024-04-19
作者简介:王晓东(1985—),博士研究生
通信作者:郭阳明(1978—),教授 e-mail:yangming_g@nwpu.edu.cn

同的维度压缩到合适的大小,并使信息损失最小化。具体来说,本方法将变长序列沿着表示维度的投影嵌入到一个等比例的结构中,形成一个张量。考虑到原始序列所具有的特征各不相同,首先建立多个函数对特征进行提取。然后,将信息熵作为度量语义丰富性的标准,引入注意力分配权重,根据需求动态地调整不同语义特征之间融合的比例来融合表示,使得多个序列表示被进一步调整为小批量,随后被送入模型。这样,原始数据中的语义信息被尽可能地保留下来,有效解决了当前深度学习框架中截断处理机制造成的语义信息损失和关系截断问题。此外,本文从理论上采用信息熵度量信息损失,并证明了本策略的信息损失要小于截断嵌入的信息损失。所提出方法的整个计算过程可以作为多个模型的数据预处理管道,将数据预处理和模型训练过程解耦,从而可以在内存等廉价存储设备上进行计算。

本方法在 4 个典型的变长序列数据集上进行了实验,实验结果表明,本文提出的方法相比传统方法,即使在小维度上也能涵盖更多的信息,在提高模型训练精度与训练速度的同时,降低了存储设备的占用率。本文的具体贡献如下:

1) 提出了一种语义感知的变长序列自适应处理方法,将非结构化的序列转换为格式化的张量,解决了由填充和截断引起的语义冗余、缺失和语义关系截断等问题,显著提高了计算效率。

2) 开发了一个处理变长序列的可插拔式通用预处理框架。该框架将模型的数据预处理与训练过程解耦,以装配任意模型,并在预处理过程中感知语义信息量,以自主适应多种投影方法。

3) 利用信息熵从理论上度量了截断嵌入的信息损失,并证明了与传统的填充或截断法相比,本文提出的方法能保留更多的语义信息。此外,开展了比较和分析实验。结果表明,本文方法在较少的维度上达到了较高的精度,不仅提高了训练速度,改善了模型训练效果,还降低了存储设备的占用率。

1　模型介绍

在本节中,提出了语义感知的变长序列自适应处理方法,以解决主流深度学习模型在处理变长序列批量训练时遇到的问题。该方法与模型的训练过程解耦,形成一个可插拔式通用预处理框架。该框架如图 1 所示,包含 2 个训练阶段:①预处理任务,使用无监督学习方法学习序列表示;②下游任务,使用交叉熵损失学习分类器。

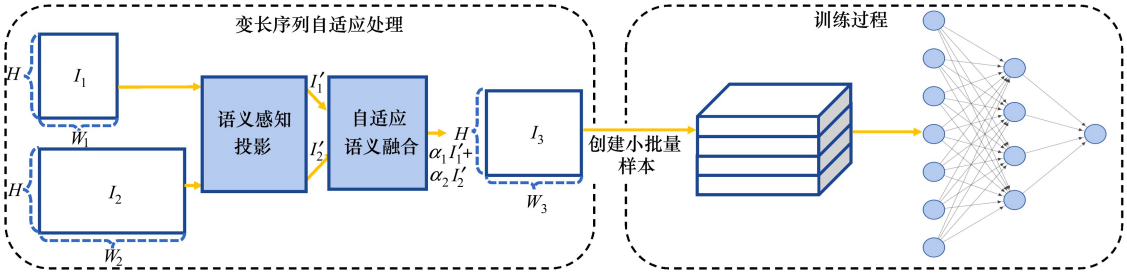


图 1　语义感知的自适应数据预处理框架

具体而言,如图 1 所示,I_1 和 I_2 是变长序列。本文提出了一个语义感知模块来感知它们的语义,该模块将输入沿着表示维度自适应地投影到等长或等比例的结构中。此外,引入了一个基于语义的注意力机制,利用语义信息作为权重对特征表示进行融合。然后,将融合后的表示组成小批量输入到模型中进行训练。

1.1　信息损失分析

在本节中,将从理论上证明,截断的信息损失要大于投影嵌入的信息损失。这里的“投影嵌入”是指利用投影技术将变长序列从高维表示空间映射到低维分布式表示空间。假定有序列 $\boldsymbol{X}$,$\boldsymbol{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n]$,$\boldsymbol{x}_i \in \mathbf{R}^n$,本文通过冯诺依曼熵[17] 来度量信息,利用矩阵的特征值或奇异值来计算信息熵,如(1)式所示。

$$H=\sum_{i=1}^{N} H_i = \sum_{i=1}^{N} -\frac{\lambda_i}{\sum_{n=1}^{k}\lambda_n}\log_2\frac{\lambda_i}{\sum_{n=1}^{k}\lambda_n} \tag{1}$$

语义感知的投影方法利用主成分分析将变长序列投影到固定长度。用主成分分析法计算样本协方差矩阵的特征值,并将它们从大到小排列,得到变换矩阵。然后,根据变换矩阵的线性变化,将原始样本投影到一个新坐标系中。由此,投影得到的定长序列 $\boldsymbol{X}' = [x'_1, x'_2, \cdots, x'_k]$ 为方差最大投影线性组合,且其对应的特征值为 $[\lambda_1, \lambda_2, \cdots, \lambda_k]$, $\lambda_1 > \lambda_2 > \cdots > \lambda_k$。

对于截断法,通过随机截断变长序列得到的定长序列 $\boldsymbol{X}' = [x'_1, x'_2, \cdots, x'_k]$,所对应的特征值 $[\lambda_1, \lambda_2, \cdots, \lambda_k]$ 也是随机的。根据(1)式,矩阵信息熵与特征值的大小呈正相关,因此在对同样的变长序列进行处理后,投影法对应矩阵的特征值之和大于随机截断法。也就是说,通过投影法得到的定长序列 $\boldsymbol{X}'$ 的信息熵比传统截断法得到的信息熵大。

为进一步证明,本文在 MR 数据集上开展了实验,通过冯诺依曼熵比较投影法与截断法所得的定长序列信息熵大小。对 MR 数据集上的每个序列计算信息熵,结果如图 2 所示(仅展示前 20 个)。可以明显地观察到,投影法得到的每个序列的信息量均大于截断法,即投影法得到的定长序列 $\boldsymbol{X}'$ 的信息熵比传统截断法得到的信息熵大。

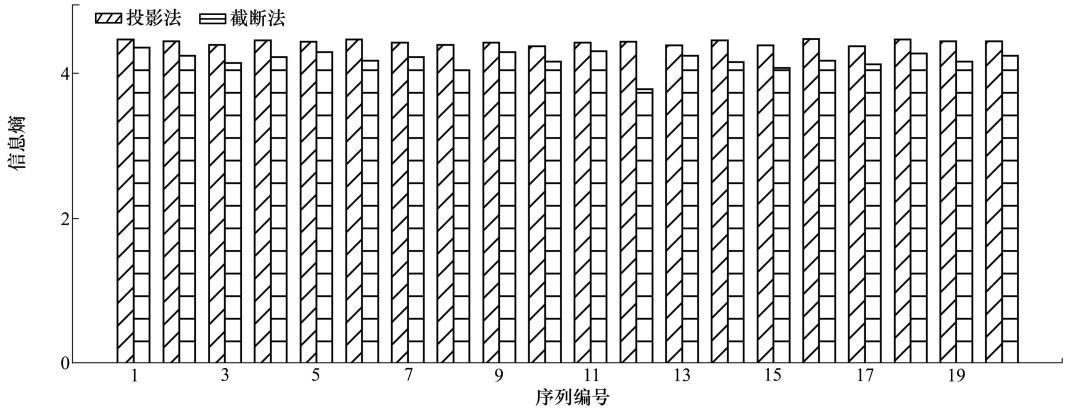


图 2 投影法与截断法所得信息熵对比

1.2 语义感知的自适应数据预处理

当前的主要问题是将变长序列处理成固定长度序列时,不但要尽可能多地保留语义信息,而且要减少冗余信息。因此,本文提出一种语义感知的自适应方法,将语义信息量作为语义丰富度的度量标准。其核心是利用一个自适应指标来度量语义丰富度,并根据语义丰富度为变长序列的预处理分配权重。

1.2.1 语义感知投影

本节提出了语义感知的投影模块,可以将输入的内容沿着序列维度方向转化为同等大小的结构,能够整合文本中包含的丰富信息。由于原始数据的文本表示是分布式的,如何应用统一的投影方法来实现合理的投影是一个值得思考的问题。在此,本文考虑采用一个灵活模块来提取语义信息。如图 3 所示,构建了多成分分析函数定义输入的投影模块,以获得最佳编码,使输入空间在特征向量上的投影具有最大方差。

具体而言,假定得到的序列化数据(句子)的输入是一个矩阵 $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N]$, $\boldsymbol{x}_i \in \mathbf{R}^n$,其中 $\boldsymbol{x}_i$

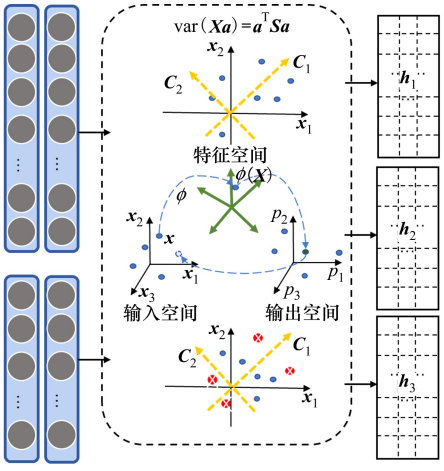


图 3 语义感知投影

为使用全局向量表示的 n 维词汇嵌入。为了进行批量训练,就要找到一组线性投影,使得原始变长序列投影到该空间中,并使输入空间在特征向量上的投影方差最大,以最大化地保持原始信息。$\boldsymbol{x}_i$ 的线性组合可以表示为

$$\boldsymbol{h}_i = \sum_{i=1}^{N} \boldsymbol{a}^{\mathrm{T}} \boldsymbol{x}_i = \boldsymbol{a}^{\mathrm{T}} \boldsymbol{X} \tag{2}$$

式中:$\boldsymbol{h}_i$ 为原始数据投影后的表示;$\boldsymbol{a} \in \mathbf{R}^n$ 为投影向量。通过使用多成分分析函数启发式地构造 $\boldsymbol{h}_i$,具体计算过程如(3) ~ (5) 式所示。

$$\bar{\boldsymbol{x}} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}_i \tag{3}$$

$$\tilde{\boldsymbol{I}}_c := \boldsymbol{X} - \bar{\boldsymbol{x}} \tag{4}$$

$$[\lambda_1, \lambda_2, \cdots, \lambda_k], [\boldsymbol{a}_1, \boldsymbol{a}_2 \cdots, \boldsymbol{a}_k] \leftarrow \mathrm{Decomposition}(\tilde{\boldsymbol{I}}_c) \tag{5}$$

式中:$\tilde{\boldsymbol{I}}_c$ 为中心化处理后的矩阵;Decomposition(·)是计算主成分的相关算法,它是一个灵活模块,允许多个函数对数据进行学习。λ_i,$\tilde{\boldsymbol{I}}_c$ 为求解输入空间 $\boldsymbol{I}_c$ 的协方差矩阵的特征值与特征向量。对于线性可分数据,通过主成分分析法(PCA),将协方差矩阵 $\boldsymbol{\Sigma} = \frac{1}{N} \cdot \tilde{\boldsymbol{I}}_c \cdot \tilde{\boldsymbol{I}}_c^{\mathrm{T}}$ 对角化,经(6) 式最优化,找到方差最大的线性组合,得到对应矩阵的前 k 维[$\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_k$] 就是所需的基空间。

$$\max \boldsymbol{a}^{\mathrm{T}} \boldsymbol{\Sigma} \boldsymbol{a} - \boldsymbol{\lambda}(1 - \boldsymbol{a}^{\mathrm{T}} \boldsymbol{a}) \tag{6}$$

对于线性不可分数据,通过核主成分分析法(KPCA),用一个非线性映射把原始矩阵 $\boldsymbol{X}$ 映射到高维空间,得到新矩阵 $\phi(\boldsymbol{X})$,再对新矩阵 $\phi(\boldsymbol{X})$ 进行主成分投影。映射关系如(7)式所示。

$$\phi(\boldsymbol{X}): \mathbf{R}^D \rightarrow \mathbf{R}^K, K \gg D \tag{7}$$

式中,ϕ 为非线性映射。为了获取更多包含不同语义信息的表示,语义感知投影模块还可以灵活地嵌入其他投影方法。比如通过稀疏 PCA 找到能够最大化重构数据的稀疏成分集合等方法。

1.2.2　自适应语义融合

在自适应语义融合模块中,语义信息被用作自适应融合表示的标准。确切地说,信息熵作为一种不确定信息的度量手段,被用于度量语义细节。信息熵的定义如(1)式所示。

本文采用信息熵作为自适应权重来度量表示中的语义信息,以实现原始信息的最大化。权重定义了原始信息的保存程度,权重越高,信息保存得越完整。从(1)式中得到的自适应权重体现了投影表示的信息度量结果。综上所述,将信息熵作为注意力机制关注不同表征的关键部分,根据需求动态调整不同语义特征之间融合的比例,将其融合以实现自适应语义。

其中,本文使用 softmax 函数在 0~1 之间映射 $H(\boldsymbol{x}_i)$,以获取每个表示的权重 α_i,且权重总和为 1,$\boldsymbol{h}_i$ 是原始数据的投影。

自适应语义融合模块的流程图如图 4 所示。首先通过语义感知的投影模块获得表示。然后,通过语义的信息熵度量以及与语义的自适应融合得到最终的表示。

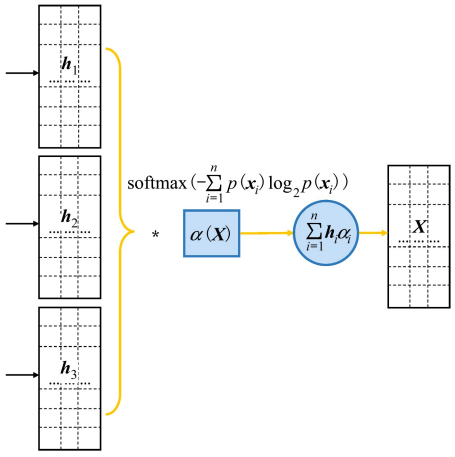


图 4　自适应语义融合

2　实　验

本文方法是一种启发式的无监督学习方法,以解析的方式求解其特征空间,即利用数学分析技术来直接计算最优的投影矩阵或变换参数,然后将投影后的表示数据用于下游的任务性能评估。因此,本节在 IMDB[18]、MR[19]、SST1[20] 和 Agnews[21] 4 个数据集上对提出的方法进行评估。考虑从以下两方面评估方法的有效性:

1) 与传统方法相比,可以在尽可能少的维度上涵盖尽可能多的信息;

2) 改进后的方法可以减少训练时间,提升训练效果,同时降低存储设备的占用率。

2.1　实验设置

2.1.1　数据集

本文在 IMDB[18]、MR[19]、SST1[20] 和 Agnews[21] 4 个数据集上对提出的方法进行了评估。

1) IMDB:由 IMDB 网站的 50 000 条电影评论组成,包括 25 000 条训练样本和 25 000 条测试样本,用于检测评论中的积极或消极情绪。

2) MR:和 IMDB 一样,电影评论数据集有1 000 个文件。每条评论为一句话,可分为积极情绪或消极情绪。

3）SST1：SST1 则被应用于具有细粒度标签的分析语料库。整个数据集包含 8 544 个训练样本，1 101个验证样本和 2 210 个测试样本。

4）Agnews：此数据集收集了 2 000 个新闻源的 100 多万篇新闻文章，主要用于新闻分类任务。目前采用了 4 个最大的新闻类别：世界、体育、商业和科学/技术。每个类别的训练样本数为 30 000，测试样本数为 1 900。

数据集汇总统计如表 1 所示，参数分别为类别、数据类型、数据集样本量以及测试集样本量。

表 1 数据集汇总统计

数据集	类别	数据类型	总样本量	测试样本量
IMDB	2	long	50 000	25 000
MR	2	long	1 000	交叉验证
Agnews	4	short	120 000	7 600
SST1	2	short	118 55	2 210

2.1.2 评估指标

本文采用 F_1 分数（主要评价指标）、平均训练时间和数据大小 3 个评价指标评估方法的有效性。F_1 计算公式为：

$$F_1 = \frac{2 \times P_{re} \times R_{ec}}{P_{re} + R_{ec}} \tag{8}$$

$$P_{re} = \frac{N_{TP}}{N_{TP} + N_{FP}} \tag{9}$$

$$R_{ec} = \frac{N_{TP}}{N_{TP} + N_{FN}} \tag{10}$$

式中，N_{TP}，N_{TN}，N_{FP} 和 N_{FN} 分别代表真阳性、真阴性、假阳性和假阴性；精确率 P_{re} 指的是结果中预测为阳性的样本中真阳性的概率；召回率 R_{ec} 指的是原始阳性样本中最终被正确预测为阳性的概率。

2.1.3 基线

考虑到本文提出的方法要与训练阶段解耦，使用几种简单但高效的深度学习方法来对本文方法进行评估，其中包括基于 RNN（bi-GRU[9]、bi-LSTM[8] 与 RNN-Capsule[22]）、基于 CNN（CNN[23] 与 Text-CNN[7]）和基于注意力的模型（HAN[10] 与 Transformer[11]）。

2.1.4 实现细节

实验采用的深度学习框架为 TensorFlow 2.0。实验硬件为 1 台装有英特尔 8 核 i7-8700K CPU（3.70 GHz，64 GB CPU 内存）和 1 块 GeForce RTX 2060OC 卡（6 GB GPU 内存）的机器。

采用多数据集上对比实验的方式，以证明提出的方法可以用尽可能少的维度涵盖尽可能多的信息。具体而言，比较了填充法的输入长度大于本文方法时的性能。为了确保对比的公平性，实验在批量大小和特征尺寸上采用了相同的配置。在实验中，使用 200 维的全局向量对所有模型进行预训练的词嵌入。

2.2 与相关研究的对比

在本节中，为了通过实验证明本文方法可以包含更多的信息，比较了不同输入长度下的基线性能。给出了 2 种数据预处理的设置，包括传统方法和本文方法。前者将数据集的每个序列长度进行填充。后者则对序列尺寸进行压缩，将二者放入模型中比较性能。

具体而言，IMDB 和 MR 是由文档组成的数据集，其中的每条评论又由多个句子构成。对于这样的长文本，本文将其输入维度投影为 10，即 1 个句子包含 10 个单词。相比之下，传统方法的输入维度为 50，即 1 个输入句子由 50 个词组成，是本文方法输入长度的 5 倍。在这种情况下比较基线模型的性能。

如表 2 所示，可以看出，采用本文方法所装配的模型在性能上均优于传统方法。具体而言，IMDB 数据集包含 50 000 条格式规范的影评。与 IMDB 相比，MR 数据集中的评论更偏向口语化，每条评论的句子数量更多，但总评论数较少，仅有 1 000 条。因此，所有模型在 IMDB 上的表现均优于 MR。此外，IMDB 的评论相对规范，因此本文方法与传统方法的性能差异不大，仅为 1%～2%。在 MR 评论偏口语化的情况下，本文方法的效果明显好于传统方法，约为 4%～8%。综上所述，本文方法可以使表示囊括更多的语义信息。

对于短文数据集 AGnews 和 SST1，每篇评论只包含 1 个句子，本文将其输入维度投影为 5，即 1 个句子包含 5 个词，而传统方法则将输入处理为 10，即 1 个输入句子包含 10 个词。此时，传统方法的输入长度是本文方法输入长度的 2 倍，将本文方法与传统方法在不同模型上的性能进行比较。表 3 显示了与表 2 相同的结论，即本文提出的方法在所有模型上的性能均优于传统方法。具体而言，Agnews 是一个用于新闻主题分类的数据集，数据量庞大且类别间差异显著，因此模型在 Agnews 上的表现更加突

出。此外,本文方法比传统方法在性能上提升了约 3%~5%。而对于 SST1 数据集,由于句子较短,模型分类性能不甚理想,本文方法与传统方法的性能差异度并不显著。

表 2　本文方法与传统方法在 IMDB 和 MR 上的性能比较

模型种类	模型	IMDB		MR	
		改进前 F_1	改进后 F_1	改进前 F_1	改进后 F_1
RNN-based	bi-GRU	79.64±0.26	80.65±0.32	56.97±1.07	64.46±0.95
	RNN-Capsule	78.11±0.33	79.93±0.34	58.12±0.65	62.16±0.53
	bi-LSTM	79.18±0.41	80.66±0.40	58.69±1.48	66.83±0.69
CNN-based	CNN	77.76±0.34	79.93±0.06	56.35±1.53	61.37±0.61
	Text-CNN	76.08±0.77	77.14±0.26	60.60±1.02	65.47±1.28
Attention-based	HAN	78.28±0.13	79.56±0.11	63.15±1.43	67.56±1.23
	Transformer	79.65±0.22	81.76±0.32	64.19±1.23	68.89±0.23

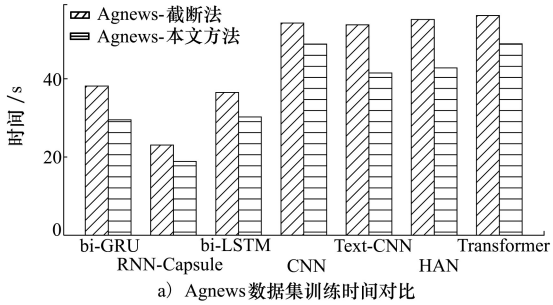
表 3　本文方法与传统方法在 Agnews 和 SST1 上的性能比较

模型种类	模型	Agnews		SST1	
		改进前 F_1	改进后 F_1	改进前 F_1	改进后 F_1
RNN-based	bi-GRU	85.82±0.23	89.66±0.13	51.96±0.01	52.35±0.15
	RNN-Capsule	84.54±0.24	88.96±0.52	51.31±0.06	51.62±0.36
	bi-LSTM	85.54±0.18	89.76±0.15	47.64±0.26	51.01±0.50
CNN-based	CNN	83.83±0.18	86.68±0.07	51.21±0.08	51.55±0.27
	Text-CNN	82.69±0.12	87.24±0.32	50.01±0.49	50.25±0.97
Attention-based	HAN	84.19±0.02	88.35±0.12	51.71±0.21	51.85±0.97
	Transformer	84.26±0.31	89.25±0.02	50.31±0.11	51.25±0.12

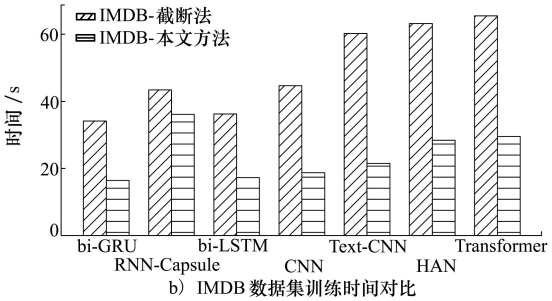
表 2 和表 3 的结果证明,用本文方法对输入进行预处理可以在较少的维度下包含更多的信息。与基于填充或截断的传统方法相比,改进后的方法在尽可能保存原始信息的情况下还能捕获到更多的特征。其主要原因在于语义感知自适应预处理方法使最终的表示囊括了更多的语义信息,即用最少的字数获得了更丰富的语义。

此外,基于 RNN 的模型(bi-GRU、RNN-Capsule、bi-LSTM)的链式结构和注意力机制使模型能够捕获更多的上下文信息。在序列间具有长时间依赖性的文本数据上的性能要优于基于 CNN 的(Text-CNN)模型。

为了便于分析,图 5 和图 6 分别给出了模型在不同数据集上的训练时间和数据规模,以比较传统方法和本文方法之间的差异。



a) Agnews数据集训练时间对比



b) IMDB数据集训练时间对比

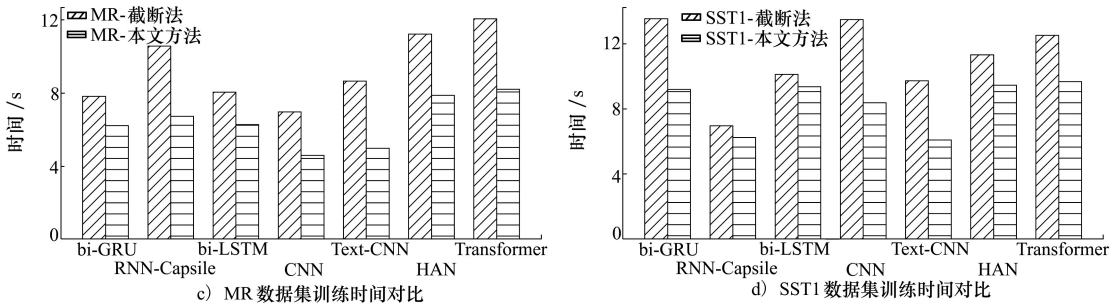


图 5 不同维度的截断法与本文法在各数据集上的训练时间对比

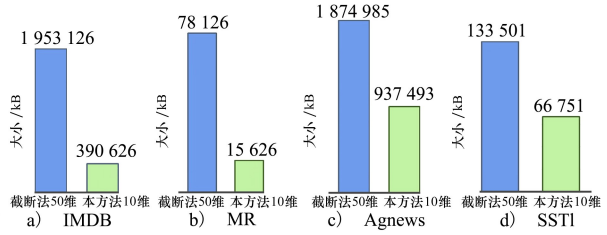


图 6 截断法与本文方法在各数据集上的多维度数据规模对比

从图 5 可以看出,与传统方法相比,运用本文方法在全部 4 个数据集上训练模型的时间要少得多,有效提高了模型的训练效率。此外,与 SST1 和 MR 相比,IMDB 和 Agnews 的数据量更大,因此需要更长的训练时间。图 6 显示了传统方法和本文方法在对输入进行预处理时的内存消耗,其中蓝条和绿条分别代表传统方法和本文方法的数据规模,可直观地发现,用传统截断法进行数据预处理会消耗更多的内存。本文通过压缩数据规模,减小了将数据加载到 GPU 的时间,节省了额外开销,提升了训练效率。

2.3 实验结果与分析

本节在进一步分析的基础上进行了实验,分别用传统方法以及本文方法对现有模型进行装配并比较性能,证明了本文方法可有效对数据预处理与训练过程进行解耦,装配任意模型的同时提高了性能。

对于长文本数据集 IMDB 和 MR,本文假定一句话中包含 50 个单词。对于 Agnews 和 SST1 数据集,本文假定 1 个句子包含 10 个单词,并采用传统方法和本文方法进行实验。表 4 和表 5 显示了 F_1 得分的比较结果,表明本文方法在全部 4 个数据集上均有突出的表现。

表 4 基于现有模型在 IMDB 和 MR 上的方法性能比较

模型种类	模型	IMDB		MR	
		改进前 F_1	改进后 F_1	改进前 F_1	改进后 F_1
RNN -based	bi-GRU	79.64±0.26	81.99±0.22	56.97±1.07	65.40±0.75
	RNN-Capsule	78.11±0.33	79.23±0.14	58.12±0.65	63.23±0.43
	bi-LSTM	79.18±0.41	82.12±0.12	58.69±1.48	64.86±0.97
CNN -based	CNN	77.76±0.34	81.14±0.20	56.35±1.53	58.00±1.50
	Text-CNN	76.08±0.77	81.80±0.17	60.60±1.02	64.11±0.60
Attention -based	HAN	78.28±0.13	82.45±0.24	61.32±0.21	65.32±0.80
	Transformer	79.65±0.22	83.68±0.12	60.54±0.24	64.67±0.76

表 5 基于现有模型在 Agnews 和 SST1 上的方法性能比较

模型种类	模型	Agnews		SST1	
		改进前 F_1	改进后 F_1	改进前 F_1	改进后 F_1
RNN -based	bi-GRU	85.82±0.23	89.30±0.13	51.96±1.29	51.22±0.15
	RNN-Capsule	84.54±0.24	88.26±0.45	51.31±0.06	52.13±0.23
	bi-LSTM	85.54±0.18	90.01±0.09	47.64±0.26	52.19±0.01
CNN -based	CNN	83.83±0.18	88.43±0.33	51.21±0.83	51.39±0.70
	Text-CNN	82.69±0.12	88.81±0.18	50.51±0.49	51.30±0.67
Attention -based	HAN	82.95±0.34	87.92±0.15	51.25±0.26	52.45±0.32
	Transformer	83.45±0.65	88.34±0.27	50.65±0.45	53.65±0.42

特别是在 IMDB 和 MR 上的 F_1 得分大约提升了 2%~4%。以上 2 个数据集的序列中包含词数较多,因此模型可以捕获更多的信息。值得注意的是,MR 数据集是一个表达偏口语化的具有复杂序列结构的数据集,而本文方法作用于该数据集上的性能十分显著,表明本文方法可以提取到更多的语义信息,与对比实验得出的结论相同。在 Agnews 上的得分结果则更加突出,较传统方法提升了约 5%。这是因为该数据集含有 120 000 条文本序列。这证明了本文方法在样本充足的情况下同样效果显著。在 SST1 上的表现仅提升了约 1%,差异并不明显,主要因为样本以短句为主,包含的信息有限。因此,实验结果表明本文方法可有效将数据预处理与训练过程解耦,从而装配出任意模型并提高性能。

为了分析可变长度的影响,本文比较了不同长度(包括 10,50 和 100)的语义感知投影和截断后的信息熵。具体来说,取前 100 个样本的不同长度情况下用投影方法与传统方法处理的序列样本包含的信息熵进行比较分析。比较结果如图 7 所示,实线代表本文所提出的方法,虚线代表传统方法,从图中可以看出,在序列长度相同的条件下,实线一直高于虚线,即提出的方法在相同长度条件下都比传统方法获得的序列具有更大的信息熵。同时也可以证明,序列长度与信息熵大小呈正相关,序列越长包含的信息量越大。

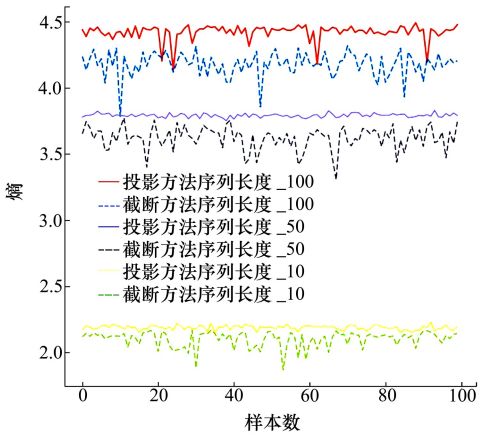


图 7　不同长度的投影和传统截断方法的信息熵比较

此外,本文在保持输入长度一致的情况下对每个模型分别使用 2 种方法得到的序列进行训练时间的比较实验。从图 8 可知,相比于传统方法,本文方法得到的序列实现在 4 个数据集的训练时间相对较短。并且,在输入长度一致且训练时间持平的情况下,本文方法在各模型上的性能仍优于传统方法。

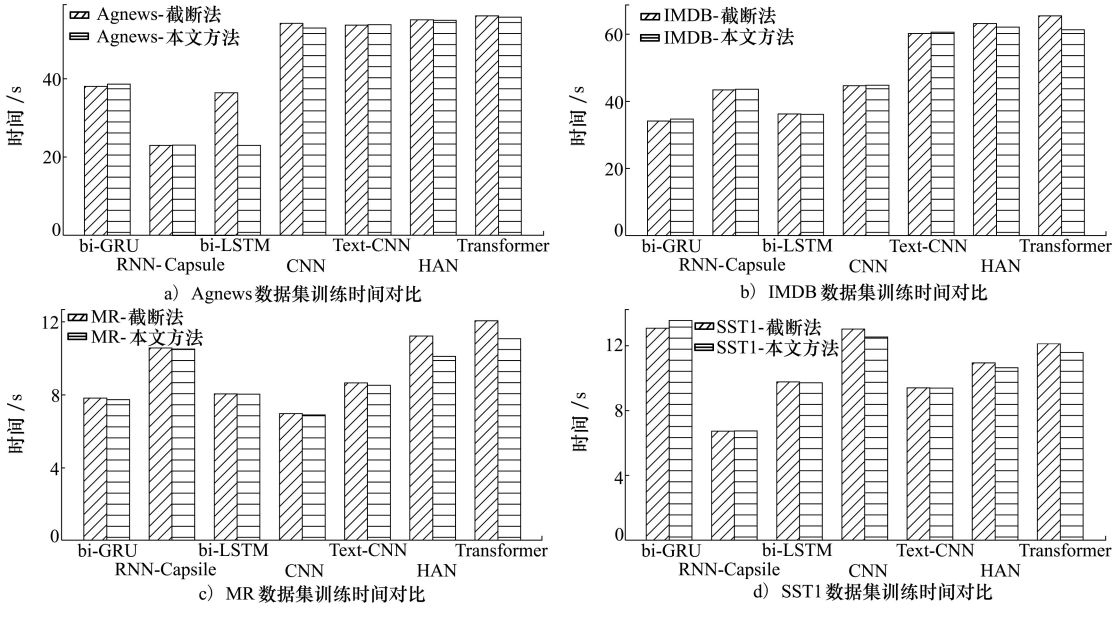


图 8　相同维度的截断法与本文方法在各数据集上的训练时间对比

3　结　论

本文提出了一种语义感知的自适应数据预处理方法,以解决深度学习框架在小批量训练上处理变长序列时导致的语义损失或信息冗余问题。为了提升学习效率,当前主流的深度学习模型需要将变长序列填充或截断至完整的张量结构,形成批量以进

行训练,这会造成语义损失或信息冗余。本文提出的方法可以感知语义,并自适应地调整数据预处理方法,使变长序列转化为固定长度,有效减少引入冗余,提高计算效率。

下一步的研究工作要将该方法与更多使用变长序列的领域相结合,如时间序列数据分析、语音识别等。此外,可以将更多的下游任务成果嵌入到模型中,以提升模型的泛化性。

参考文献:

[1] MIKOLOV T, SUTSKEVER I, CHEN K, et al. Distributed representations of words and phrases and their compositionality[C]//Advances in Neural Information Processing Systems, 2013

[2] PENNINGTON J, SOCHER R, MANNING C D. Glove: global vectors for word representation[C]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 2014: 1532-1543

[3] JOULIN A, GRAVE E, BOJANOWSKI P, et al. Bag of tricks for efficient text classification[C]//Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, 2017: 427-431

[4] BARKAN O, KOENIGSTEIN N. Item2vec: neural item embedding for collaborative filtering[C]//2016 IEEE 26th International Workshop on Machine Learning for Signal Processing, 2016: 1-6

[5] CONG S, ZHOU Y. A review of convolutional neural network architectures and their optimizations[J]. Artificial Intelligence Review, 2023, 56: 1905-1969

[6] ORVIETO A, SMITH S L, GU A, et al. Resurrecting recurrent neural networks for long sequences[C]//International Conference on Machine Learning, 2023: 26670-26698

[7] KIM Y. Convolutional neural networks for sentence classification[C]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 2014: 1746-1751

[8] BANSAL M, GOYAL A, CHOUDHARY A. A comparative analysis of K-nearest neighbor, genetic, support vector machine, decision tree, and long short term memory algorithms in machine learning[J]. Decision Analytics Journal, 2022, 3: 100071

[9] WEERAKODY P B, WONG K W, WANG G, et al. A review of irregular time series data handling with gated recurrent neural networks[J]. Neurocomputing, 2021, 441: 161-178

[10] YANG Z, YANG D, DYER C, et al. Hierarchical attention networks for document classification[C]//Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016: 1480-1489

[11] HAN K, WANG Y, CHEN H, et al. A survey on vision transformer[J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2022, 45(1): 87-110

[12] OYEDOTUN O K, KONSTANTINOS P, DJAMILA A. A new perspective for understanding generalization gap of deep neural networks trained with large batch sizes[J]. Applied Intelligence, 2023, 53(12):15621-15637

[13] BARTOLDSON B R, KAILKHURA B, BLALOCK D. Compute-efficient deep learning: algorithmic trends and opportunities[J]. Journal of Machine Learning Research, 2023, 24(1): 77

[14] NOKHWAL S, CHILAKALAPUDI P, DONEKAL P, et al. Accelerating neural network training: a brief review[C]//Proceedings of the 2024 8th International Conference on Intelligent Systems, 2024: 31-35

[15] MARTIN A, AGARWAL A, BARHAM P, et al. Tensorflow: large-scale machine learning on heterogeneous distributed systems [J/OL]. (2016-03-16)[2024-03-21]. https://arxiv.org/abs/1603.04467? file=1603.04467

[16] PASZKE A, GROSS S, MASSA F, et al. Pytorch: an imperative style, high-performance deep learning libraryry[C]//Advances in Neural Information Processing Systems, 2019

[17] MIKLOS R, MICHAELl S, MIKLSS R. John von Neumann and the foundations of quantum physics[M]. Berlin: Springer, 2003

[18] MAAS A, DALY R E, PHAM P T, et al. Learning word vectors for sentiment analysis[C]//Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 2011: 142-150

[19] PANG B, LEE L, VAITHYANATHAN S. Thumbs up? Sentiment classification using machine learning techniques[C]//Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2002: 79-86

[20] SOCHER R, PERELYGIN A, WU J, et al. Recursive deep models for semantic compositionality over a sentiment treebank[C]

//Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013: 1631-1642

[21] ZHANG X, ZHAO J, LECUN Y. Character-level convolutional networks for text classification[C]//International Conference on Neural Information Processing Systems, 2015

[22] WANG Y, SUN A, HAN J, et al. Sentiment analysis by capsules[C]//Proceedings of the 2018 World Wide Web Conference, 2018: 1165-1174

[23] ALBAWI S, MOHAMMED T A, AL-ZAWI S. Understanding of a convolutional neural network[C]//2017 International Conference on Engineering and Technology, 2017: 1-6

A framework of variable-length sequence data preprocessing based on semantic perception

WANG Xiaodong[1], WANG Jiwei[1], ZHONG Zhihao[2], YANG Huan[1], YAO Hongjing[3], GUO Yangming[3]

(1.School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China;
2.School of Software, Northwestern Polytechnical University, Xi'an 710072, China;
3.School of Cybersecurity, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: Deep learning frameworks generally adopt padding or truncation operations toward variable-length sequences in order to use efficient yet intensive batch training. However, padding leads to intensive memory consumption, and truncation inevitably loses the original semantic information. To address this dilemma, a variable-length sequence preprocessing framework based on semantic perception is proposed, which leverages a typical unsupervised learning method to reduce the different dimensionality to the exact size and minimize information loss. Under the theoretical umbrella of minimizing information loss, information entropy is adopted to measure the semantic richness, weights to variable-length representations is assigned, and the semantic richness is used to fuse them. Extensive experiments show that the information loss of the present strategy is less than the truncated embeddings, and the apparent superiority of the present method in gaining more information capability and achieving promising performance on several text classification datasets.

Keywords: variable-length sequence; data preprocessing; padding; truncation; semantic information; maximizing information

引用格式:王晓东, 王继维, 钟智昊, 等. 基于语义感知的变长序列数据预处理框架[J]. 西北工业大学学报, 2025, 43(2): 388-397

WANG Xiaodong, WANG Jiwei, ZHONG Zhihao, et al. A framework of variable-length sequence data preprocessing based on semantic perception[J]. Journal of Northwestern Polytechnical University, 2025, 43(2): 388-397 (in Chinese)