计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.200200076

# 基于对抗训练的文本表示和分类算法

张晓辉[1]　于双元[1]　王全新[2]　徐保民[1]
1 北京交通大学计算机与信息技术学院　北京 100044
2 北京交通大学海滨学院计算机与信息技术学院　河北 黄骅 061199
(1765372906@qq.com)

**摘　要**　文本表示和分类是自然语言理解领域的研究热点。目前已有很多文本分类方法,包括卷积网络、递归网络、自注意力机制以及它们的结合。但是,复杂的网络并不能从根本上提高文本分类的性能,好的文本表示才是文本分类的关键。为了获得好的文本表示,提高文本分类性能,构建了基于LSTM的表示学习-文本分类模型,其中表示学习模型利用语言模型为文本分类模型提供初始化的文本表示和网络参数。文中主要采用对抗训练方法训练语言模型,即在词向量上添加扰动构造对抗样本,再利用对抗样本和原始样本一起训练模型,通过提升模型对对抗样本的正确分类能力,提高文本表示的质量,增强模型的泛化性能,从而改善分类模型的分类效果。实验结果表明,基于对抗训练的文本分类方法在基准数据集AGNews,IMDB,DBpedia上分别实现了92.9%,93.2%,98.9%的准确率,证明了该方法能够有效提高文本分类模型的分类性能。

**关键词**:文本分类;文本表示;对抗训练
**中图法分类号**　TP391

## Text Representation and Classification Algorithm Based on Adversarial Training

ZHANG Xiao-hui[1],YU Shuang-yuan[1],WANG Quan-xin[2] and XU Bao-min[1]
1 School of Computer and Information Technology,Beijing Jiaotong University,Beijing 100044,China
2 School of Computer and Information Technology,Beijing Jiaotong University Haibin College,Huanghua,Hebei 061199,China

**Abstract**　Text representation and classification are hot topics in the field of natural language understanding. There are many text classification methods,including convolutional networks,recursive networks,self-attention mechanisms and their combinations. complex networks cannot fundamentally improve the performance of classificationtext representation is the key to text classification. In order to obtain a good text representation and improve the performance of text classification,an LSTM-based representation learning-text classification model is constructed,where the representation learning model uses a language model to provide the text classification model with initialized text representation and network parameters. main work is to adversarial training methods that is,add perturbations to word vectors to construct adversarial samples,and train the original samples. By improving the model's ability adversarial samples,the quality of text representation,and the generalization performance of the model, the classification effect of the classification model. xperimental results show that the method based on adversarial training achieves 92.9%,93.2% and 98.9% on the benchmark datasets AGNews,IMDBDBpedia,that the method can improve the classification effect of the model.

**Keywords**　Text classification,Text representation,Adversarial training

## 1　引言

文本表示即词向量,指将文字或者单词转换成向量的形式,在自然语言处理的许多任务中有着至关重要的作用。好的文本表示能显著提高文本分类、自然语言生成和机器翻译等任务的性能。目前已有很多提高文本表示的方法,最简单的是使用预训练词向量技术,即先在大规模的数据集上预训练模型,然后将生成的词向量应用于具体的自然语言处理任务。常见的预训练词向量技术主要包括Word2Vec,FastText及Glove。另一种提高文本表示的方法是使用预训练的语言模型。比如,Peters等[1]提出的ELMo模型,使用一个双向的LSTM语言模型获得一个上下文相关的词向量表示;Radford等[2]提出的OpenAI GPT模型,使用Transformer语言模型获得通用化的词向量表示,能够更好地捕获长距离语言结构;Devlin等[3]提出的BERT模型,结合了以上两种方法,本质上还是通过训练语言模型提高文本表示的质量。因此,语言模型对于提升文本表示的质量和文本分类的效果至关重要。

Goodfellow等[4]最早提出了对抗训练的概念,并且将其应用在了计算机视觉领域,用于增强模型的鲁棒性。Miyato等[5]首次将对抗训练应用于文本领域,与之前研究不同,作者并没有将对抗扰动应用在输入层的离散one-hot单词向量上,而是定义在了连续的词嵌入上,以作为一种正则化策略,

本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。

基金项目:河北省高等教育科技攻关重点项目(ZD2017304)
This work was supported by the Key Projects of Science and Technology Research of Hebei Province Higher Education(ZD2017304).

通信作者:王全新(qxwang@bjtuhbxy.edu.cn)

提高了文本分类模型的泛化性能。但是该模型使用了很多预训练技术，先生成文本表示的词向量，再在分类模型中使用对抗训练方法。在没有预训练技术的前提下，模型效果显著下降。因此，好的文本表示才是文本分类模型性能提升的关键。

本文提出了基于对抗训练的表示学习-文本分类模型(LM-AdvT)，即将文本分类任务分为两个阶段。表示学习阶段主要产生高质量的文本表示，并为文本分类模型初始化部分网络参数；文本分类阶段主要实现最终的分类任务。该模型使用基于 LSTM 的循环神经网络，表示学习阶段采用神经语言模型，并利用对抗训练方法，提高了文本表示的质量，从而提升了文本分类阶段的分类效果。Wang 等[6]仅仅使用对抗训练技术作为正则化语言模型的方法，目的是增强模型的鲁棒性，与本文出发点不同。通过产生的文本表示，本文方法可以应用到任何自然语言处理任务，而不仅仅是文本分类；同时我们通过文本表示模型，也保留了循环神经网络的参数信息，从而加快了后续文本分类模型的训练速度。

## 2 相关工作

文本表示不仅仅要方便分类模型的训练，还要充分涵盖语义信息。文本表示方法分为传统的文本表示方法和基于神经网络的方法。传统的文本表示方法主要指基于向量空间模型和基于主题模型的方法。向量空间模型即通常所指的词袋模型，即将文本表示成实数值组成的向量，通常使用 TF-IDF 的计算方式。这种模型的优点是计算简单，容易理解，但是存在着明显的缺点，比如随着词表的增大，向量空间急剧膨胀，向量维度高度稀疏，而且无法表达词语的多义性，也不能体现句子内在的结构信息和语义信息。为了提高文本的语义表示能力，主题模型应运而生，即通过在文档中学习潜在的主题，计算出每篇文章的主题分布。主题模型主要有(Latent Semantic Analysis，LSA)和(Latent Dirichlet Allocation，LDA)。LSA 主要通过奇异值分解，实现对原始矩阵的降维，得到文档向量和词向量。但是，LSA 没有假设主题的先验分布，所以文档的数目越大，训练参数越多。LDA 引入了狄利克雷先验分布，解决了 LSA 中存在的问题。但是主题模型也存在着参数空间大、训练时间长、性能不稳定、建模局限性等问题。

目前，使用最广泛的文本表示方法是基于神经网络的方法。这其中包括以 Word2Vec，Glove，FastText 为代表的基于词向量合成的模型，以循环神经网络和卷积神经网络为代表的基于 RNN/CNN 的深度学习模型，以及层级注意力和自注意力的基于注意力机制的模型。Word2Vec 为 Bengio 等[7]于 2003 年提出的神经网络语言模型的简化版本，本质上是通过对神经语言模型使用一些优化技巧使其专注于词向量的产生，得到词的低维、稠密的向量表示，即词向量或词嵌入。但是，word2vec 产生的词向量属于静态词向量，并不能体现复杂语境下的语义信息和词义信息。为了能够合理表达出自然语言的语义和语法信息，研究者们提出了神经网络结构 LSTM 和 CNN。2014 年，Kim[8]提出了基于卷积神经网络的文本分类网络模型，其主要特点是能够利用大小不同的卷积核提取重要的局部特征，但是在有限的窗口下提取句子特征，无法对长距离的信息进行建模，也无法体现单词间的词序信息。针对 CNN 的不足之处，出现了使用 LSTM 和 GRU 等擅长捕捉长距离信息的循环神经网络，以及两种网络的结合网络及各种变体。在大多情况下，对于长文本分类或者文档分类任务，某些特定的单词或短语对于类别有着极高的价值，甚至直接决定了类别，如果能够从大量的信息中快速筛选出这些价值高的内容，则能极大地提高任务处理的效率和准确率。因此，基于注意力机制的模型也被应用在了文本分类领域，其本质是在注意力的作用下依然可以较好地捕捉到有效的特征信息，忽略无意义的输入。

Szegedy 等[9]于 2013 年首次提出了对抗样本的概念，即通过对输入进行微小的扰动而创建的样本，但是这个样本足以使机器学习模型判断错误，从而增加模型的损失。对抗训练在图像领域作为一种防御方式，通过添加扰动构造对抗样本，训练模型以正确分类未修改样本和对抗样本的过程，从而增强模型在遇到对抗样本时的鲁棒性；其同时作为一种正则化策略，也能一定程度地提高模型的泛化能力。Miyato[5]将对抗训练应用在了自然语言处理领域，与图像领域不同作者将对抗扰动应用在了分类模型的 word embedding 层作为一种正则化技术，在模型的训练过程中使用对抗训练改善模型的分类效果，使得对抗训练不再是为了防御基于梯度的恶意攻击，反而更多地是作为一种正则化方法，提高了模型的泛化能力。但是，文本分类性能的好坏不仅与网络模型有关，好的文本表示才是关键。因此，本文方法专注于基于对抗训练生成好的文本表示。通过在神经语言模型上应用对抗训练方法提高文本表示的质量，增强文本表示的多样性，再通过基于 LSTM 的文本分类模型提高文本分类模型的分类性能。

## 3 基于对抗训练的文本表示和分类算法

基于对抗训练的文本表示和分类算法主要分为两个阶段，分别为表示学习阶段和文本分类阶段。模型架构如图 1 所示。

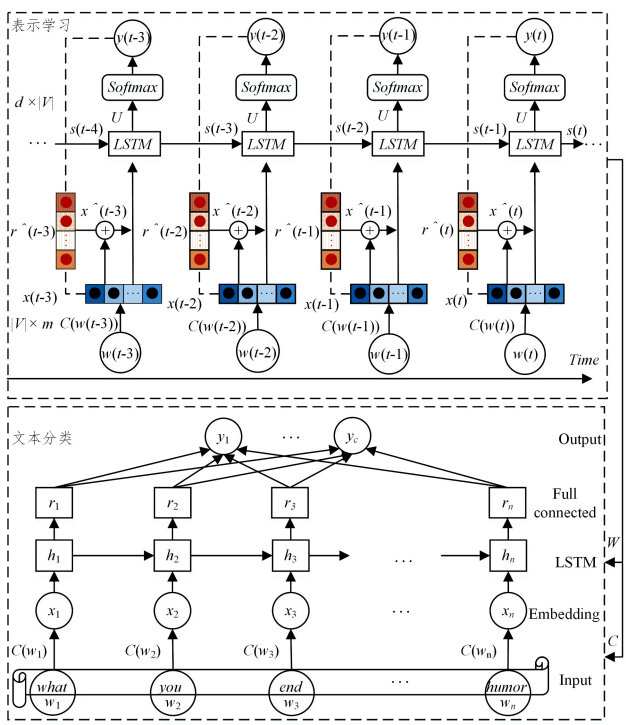



图 1 所提模型的架构图

Fig. 1 Architecture of proposed model

表示学习阶段使用对抗训练方法提高文本表示的质量，

为下一步的文本分类生成好的文本表示即词向量，同时初始化文本分类模型的部分权重参数。

在表示学习模型中，主要采用神经网络语言模型，通过构建神经网络的方式来建模自然语言内在的依赖关系，同时采用对抗训练的方法，通过构建对抗样本，训练模型正确分类的泛化性能，从而提高文本表示的质量。

表示学习模型在 $t$ 时刻的输入是 $w(t)$，$w\in V$。$w(t)=w_1,w_2\cdots w_n$ 表示在 $t$ 时刻输入的长度为 $n$ 的文本序列，$V$ 为训练数据集的词汇表，其长度根据训练数据集的不同而变化，设置最大值为 10001。将 $t$ 时刻输入的文本序列 $w(t)$ 转换成索引之后，通过矩阵 $C$ 映射为文本序列的分布式特征表示，即特征向量。这里的 $C$ 是一个所有文本序列共享的 $|V|\times m$ 的自由参数矩阵，其中 $m$ 为自定义参数，通常设置为 256 或者 512，即词汇表 $V$ 中的任意单词 $w_i$ 都能通过 $C$ 映射为实向量 $x_i\in\mathcal{R}^m$。矩阵 $C$ 的第 $i$ 行即为单词 $w_i$ 的文本表示或者特征向量，维度为 $m$。在 $t$ 时刻，语言模型的输入向量为 $x(t)$。$x(t)=C(w(t))=C(w_1(t)),C(w_2(t)),\cdots,C(w_n(t))$ 即 $t$ 时刻输入的单词的文本表示的级联。同时，为了获得更好的文本表示，采用对抗训练的方法来训练语言模型，将 $t$ 时刻产生的对抗样本 $\hat{x}(t)$ 和原始输入向量 $x(t)$ 一同作为隐藏层的输入 $\tilde{x}(t)$，以训练模型正确分类的能力。具体的对抗训练方法将在下一节介绍。表示学习部分选择基于 LSTM 的神经网络语言模型，在 $t$ 时刻，LSTM 的输入 $\hat{x}(t)$ 有两个：当前时刻网络的输入值 $\tilde{x}(t)$，上一时刻 LSTM 的隐藏状态 $s(t-1)$，即：

$$\hat{x}(t)=\tilde{x}(t)+s(t-1)$$

在 $t$ 时刻，LSTM 的隐藏状态 $s(t)$ 包括上一时刻 LSTM 的输出值 $h(t-1)$ 及上一时刻的单元状态 $c(t-1)$，即：

$$s(t)=h(t-1)+c(t-1)$$

具体的 LSTM 单元内的计算规则不再详细介绍。对于语言模型的输出层结果预测，采用一个全连接层进行特征空间转换，即：

$$y(t)=f(U^{\mathrm{T}}h(t)+b)$$

该层的输入，是在 $t$ 时刻 LSTM 层的输出值 $h(t)\in\mathcal{R}^{d_h}$，其中 $d_h$ 是 LSTM 隐藏层的维度"输出是 $t$ 时刻，该输入文本序列的下一个单词的预测值 $y(t)\in\mathcal{R}^d$，其中 $d$ 是目标类别数，$U\in\mathcal{R}^{d\times|V|}$，$b\in\mathcal{R}^{|V|}$ 是学习可得到的权重和偏置，$f(z)$ 是一个 softmax 函数，将预测结果归一化为属于每个类别的概率。

表示学习阶段训练结束后，就进入了文本分类阶段。文本分类模型主要分为 5 个部分，分别为输入层（Input Layer）、词嵌入层（Embedding Layer）、循环神经网络层（LSTM Layer）、全连接层（Full connected Layer）和输出层（Output Layer），并且采用在表示学习阶段训练好的文本表示 $C\in\mathcal{R}^{|V|\times m}$ 初始化词嵌入层，采用表示学习阶段 LSTM 网络的参数 $W$ 初始化文本分类阶段的 LSTM 网络。输入层输入的是文本序列 $w_i=w_1,w_2\cdots w_n$，$n$ 为文本序列的长度，利用词嵌入层将原始输入文本序列 $w_i(i\in n)$ 通过词向量矩阵 $C\in\mathcal{R}^{|V|\times m}$ 逐个映射成包含语义语法信息的分布式特征表示 $x_i(i\in n)$，即：

$$x=x_1,x_2,\cdots,x_i$$

其中，$x_i=C(w_i),i\in n$。

将得到的文本表示向量 $x$ 输入 LSTM 循环网络，然后使用两个全连接层将文本从特征空间映射到类别标签。其中，第一个全连接层的单元数量 $k$ 为输入的文本序列的维度，激活函数为 Relu；第二个全连接层的单元数量 $c$ 根据分类任务的不同而不同，如果为二分类，单元数量为 1，如果是多分类，单元数量为类别数。最后，输出层使用 sigmoid 或者 softmax 分类器对 $y$ 进行归一化，得到预测的属于每个类别的概率 $p\in\mathcal{R}^C$，其中 $p$ 中的第 $i$ 个分量 $p_i$ 的计算公式如下：

$$p_i=\frac{y_i}{\sum_{c=1}^{C}y_c}$$

### 3.1 语言模型

简单地说，语言模型可以看作是，给定一个序列的前提下，预测下一个词出现的条件概率 $p(w_{1:n})$：

$$p(w_{1:n})=\prod_{i=1}^{n}p(w_i\mid w_{1:i-1})$$

其中，$w_{1:n}=[w_1,\cdots,w_n]$，表示输入序列的长度为 $n$，其中 $w_i\in V$ 表示序列中的第 $i$ 个单词。本文中基于 LSTM 的循环神经语言模型的输出 $y(t)$ 即为在 $t$ 时刻该条件概率的值，根据 LSTM 的隐藏层及 softmax 层得到。

$$y(t)=softmax\{h(t),u\}$$

其中，$h(t)$ 表示当前 $t$ 时刻，隐藏层的输出值；$u=\{u_i\}\subset\mathcal{R}^U$ 表示该 softmax 层的训练参数。

$$h(t)=f\{x(t),s(t-1);w,c\}$$

其中，$f$ 是 LSTM 单元的非线性映射；$x(t)$ 是当前 $t$ 时刻，LSTM 层的输入值，由分布式词向量表示；$s(t-1)$ 是 $t-1$ 时刻，LSTM 单元的隐藏层向量值；$w=\{w_i\}\subset\mathcal{R}^W$ 表示训练参数 LSTM 单元的权重；$c=\{c_i\}\subset\mathcal{R}^C$ 表示词嵌入向量。因此，为了得到最优的参数 $\theta=(u,w,c)$，本文的目标变成了以下优化问题：

$$\arg\min_{\theta}\{L_{lm}(x_{1:\ell},\theta)\}$$

其中，$x_{1:\ell}$ 是输入文本序列的分布式词向量表示，序列长度为 $\ell$。对于语言模型的损失，我们使用交叉熵表示：

$$L_{lm}(x_{1:\ell},\theta)=-\frac{1}{\ell}\sum_{i=1}^{\ell}\log p((x_i\mid x_{1:i-1});\theta)$$

### 3.2 对抗训练

为了提高文本表示即分布式词向量的质量，在语言模型词嵌入层添加对抗扰动，生成对抗样本，并通过对抗训练方法训练模型正确分类对抗样本和原始样本，从而提升文本表示的质量。

对抗训练最初由 Goodfellow[4] 提出，是应用于图像领域的一种防御恶意攻击的方法，用来提高模型的鲁棒性；之后，Miyato[5] 将对抗训练应用在自然语言处理领域的文本分类任务中作为一种正则化方法，用以提高模型的分类性能。我们将对抗训练方法和语言模型相结合，以提高文本表示的质量。

最优的对抗扰动，应该是在最坏情况下的扰动，这个扰动能使模型的损失最大化，即：

$$\hat{r}=\arg\max_{r,\|r\|\leqslant\varepsilon}\{L_{\mathrm{adv}}(\hat{x}_{1:\ell},\theta',r)\}$$

其中，$\varepsilon$ 是一个标量的超参数，用来对扰动的大小进行约束，不同任务的，最优 $\varepsilon$ 可能不一样，需要仔细调试。其中 $r=\{r_i\}\subset\mathcal{R}^V$ 即对于原始文本序列输入值 $x=\{x_i\}\subset\mathcal{R}^V$ 的分量 $x_i$ 都存在一个相应的扰动 $r_i$。另外：

$$\hat{x}_{1:\ell}=x_{1:\ell}+r_{1:\ell}$$

在输入值上增加扰动产生的样本，即为对抗样本。$\theta'=(u,w,c)$ 表示当前网络的参数，之所以使用 $\theta'$ 而不是 $\theta$，是因为对抗损失并不进行反向传播。$L_{\mathrm{adv}}$ 表示对抗样本的损失，其定义与 $L_{lm}$ 类似。然而，在深度神经网络中并不能准确计算出这个

扰动的值。Goodfellow[4]提出了一种近似算法，即$L_{adv}$围绕$\hat{x}_{1:\ell}$线性化，这种方法可以得到$\hat{r}$的非迭代解，即

$$\hat{r}_i=\varepsilon\frac{g_i}{\|g\|_2},g_i=\nabla_{x_i}L_{lm}(x_{1:\ell},\theta')$$

其中，$\|\cdot\|_2$表示 L2 正则，即扰动最坏的方向就是损失在输入值的梯度的正方向。因此，对抗损失定义为：

$$L_{adv}(\hat{x}_{1:\ell},\theta)=-\frac{1}{\ell}\sum_{i=1}^{\ell}\log p((\hat{x}_i|\hat{x}_{1:i-1});\theta';\hat{r})$$

对抗训练实际上是使最坏情况下的错误率最小化，即最小化对抗损失$L_{adv}(\hat{x}_{1:\ell},\theta)$，所以我们模型的最终优化目标为：

$$\arg\min_{\theta}\{L_{lm}(x_{1:\ell},\theta)+\lambda L_{adv}(\hat{x}_{1:\ell},\theta)\}$$

其中，$\lambda$是一个标量超参数，用来控制两个损失函数的平衡。

我们使用的对抗训练方法由于使用非迭代求解方法，也没有进行反向传播训练，因此并没有增加额外的计算开销。对抗训练方法的具体步骤如算法 1 所示。

**算法 1** 对抗训练算法

Input: Training data x, model parameter $\theta$ and the loss of language model $L_{lm}$ at the time of t

While not converge do
  For bitch data set of x(t)
  Calculate the gradient $g=\nabla_x L_{lm}$
  Adversarial perturbation $\hat{r}=\varepsilon\frac{g}{\|g\|_2}$
  Building the adversarial samples $\hat{x}=\hat{r}\oplus x$
  Calculate adversarial losses $L_{adv}$ through the lstm and softmax layers in order
  Minimizing loss function $L_{total}\leftarrow(L_{lm}+\lambda L_{adv})$
  Update parameters $\theta$ using gradient descent method
  End for
End while

Remark: $\theta$, $\lambda$ are hyperparamenters, $\oplus$ representation of vector stitching

## 4 实验对比与结果分析

### 4.1 数据集

在表 1 所列 3 个基准数据集上进行实验。其中，Dataset 对应使用的数据集名称；*Class* 对应目标类别个数；*Train* 对应训练样本数量；*Test* 对应测试样本数量。

表 1 基准数据集表

Table 1 Benchmark dataset

| Dataset | *Class* | *Train* | *Test* |
|---|---|---|---|
| IMDB | 2 | 25 000 | 25 000 |
| DBpedia | 14 | 558 663 | 69 853 |
| AGNews | 4 | 120 000 | 7 600 |

除了 IMDB 数据集，在其他数据集上都选择训练数据的 10%作为验证集，IMDB 选择 15%作为验证集。另外，本实验将数据集中出现的所有标点符号都作为空格处理，并将数据集中的大写单词都变为小写，删除了数据集中仅出现一次的单词。

### 4.2 实验设置

词向量：设置 IMDB 数据集上的数据维度为 256 维，DBpedia 和 AGNews 为 512 维。

表示学习：LSTM 层设置 1 024 个隐藏单元。

文本分类：LSTM 层设置 1 024 个隐藏单元；全连接层在 DBpedia 为 128 维，在 IMDB 和 AGNews 为 30 维。

优化参数设置如下。

(1)表示学习优化：初始学习率设置为 0.001，每个训练步的学习率指数衰减因子设置为 0.999 9；除了词向量维度，对其他的参数设置了范数为 1.0 的梯度剪裁，对每一个序列的开端设置了 400 个单词的截断反向传播。

(2)文本分类优化：除了初始学习率设置为 0.005，指数衰减因子设置为 0.9998 以外，其他设置与表示学习模型相同。

本实验中采用基于梯度下降的 Adam 优化方法来训练模型。

模型优化：在词嵌入层使用了 0.5 的 dropout；softmax 层采用了负采样技术，其中候选样本设置为 1024；设置扰动 $\varepsilon=5$；使用归一化的词向量 $\bar{x}_i$表示当前词向量 $x_i$，即：

$$\bar{x}_i=\frac{x_i-E(x)}{\sqrt{Var(x)}}$$

$$E(x)=\sum_{j=1}^{V}f_j x_j$$

$$Var(x)=\sum_{j=1}^{V}f_j(x_j-E(x))^2$$

其中，$f_i$表示词汇表中第 $i$ 个单词出现的频率。

### 4.3 实验结果与分析

实验结果如表 2 所列，其中 LM-AdvT 为文中所提方法。该表比较了 LM-AdvT 方法和目前最先进的文本分类方法在基准数据集 AGNews，IMDB，DBpedia 上的准确率。实验结果表明，LM-AdvT 方法在 3 个基准数据集上取得了最高的准确率。

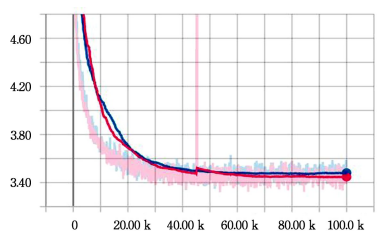
表 2 模型在测试数据集上的准确率

Table 2 Model accuracy on test dataset

(单位：%)

| Model | AGNews | IMDB | DBpedia |
|---|---|---|---|
| LM-AdvT | **92.9** | **93.2** | **98.9** |
| LSTM-adv(2016)[5] | 92.8 | 93.1 | **98.9** |
| LM-LSTM(2015)[10] | 92.61 | 92.36 | 98.5 |
| SA-LSTM(2015)[10] | — | 92.76 | 98.81 |
| Region-emb(2018)[11] | 92.8 | — | 98.9 |
| SANet(big)(2018)[12] | 92.6 | — | 98.8 |
| DRNN(2018)[13] | **92.9** | — | **98.9** |
| Encoder1-CNN-S(2019)[14] | 92.5 | — | 98.8 |
| FastText(2016)[15] | 92.5 | — | 98.6 |
| WSEM(2018)[16] | 92.66 | — | 98.57 |
| VVD(2019)[17] | 91.6 | — | 98.5 |

图 2 显示了 LM-AdvT 与 LSTM-adv 模型在 IMDB 数据集的学习曲线。LSTM-adv 模型与 LM-AdvT 模型的主要区别是，LSTM-adv 模型只在分类阶段使用了对抗训练方法。由于 LM-AdvT 使用对抗训练方法训练语言模型，因此 LM-AdvT 的函数曲线略有波动，但是总体上仍比 LSTM-adv 的损失函数值低。




注：粉色曲线代表 LM-AdvT，蓝色曲线代表 LSTM-adv

图 2 LM-AdvT 和 LSTM-adv 在 IMDB 数据集上的损失(电子版为彩色)

Fig. 2 LM-AdvT loss and LSTM-adv loss on IMDB

实验对比了 LM-AdvT 和 LSTM-adv 在表示阶段学习到的文本表示即词向量的情况，随机选择了 movies，love，poor 3 个词，并根据余弦相似度列出了与这 3 个词最相近的 5 个词，如表 3 所列。对于单词 movie，LM-AdvT 学习到的相近词都是名词且与 movie 指代相近，而 LSTM-adv 学习到的最近词却是词性和词义都不接近的 arrives；对于单词 love，LM-AdvT 给出的相近词中包括名词和动词，因为 love 既能当动词使用，又能当名词使用；但是 LSTM-adv 的相近词中却只有动词。由以上结果分析可以得出，LM-AdvT 能够有效提升文本表示的质量。

表 3 模型表示阶段词向量的近似单词

Table 3 Model phase word vector

| Model | movie | love | poor |
|---|---|---|---|
| LSTM-adv | arrives | holocaust | pathetic |
| | comedies | loved | horrible |
| | Flicks | enjoy | inept |
| | movie | hate | stolen |
| | films | loves | bad |
| LM-AdvT | stories | fool | awful |
| | performers | friendship | pretentious |
| | comedies | hate | terrible |
| | Flicks | loved | horrible |
| | films | loves | bad |

**结束语** 本文的 LM-AdvT 方法采用了两个模型，即表示学习模型和文本分类模型，它们都采用基本网络结构——LSTM 循环神经网络，分别用来学习文本表示和预测分类结果。表示学习模型使用对抗训练的方法，将对抗扰动应用在词嵌入层作为一种正则化技术，提高了文本表示的质量。将训练得到的词向量和 LSTM 神经网络参数用于接下来的文本分类模型，从而提高文本分类模型的分类效果。实验结果显示该模型确实能够提高文本分类的效果，更证明了好的文本表示是提高模型分类效果的关键。在接下来的研究中，打算使用序列化模型代替语言模型训练词向量；对于生成的词向量，除了将其应用于文本分类，还可以应用于任何自然语言处理领域，比如机器翻译、情感分析等。

## 参 考 文 献


[1] PETERS M E, NEUMANN M, IYYER M, et al. Deep contextualized word representations[J]. arXiv: 18020.05365, 2018.

[2] RADFORD A, NARASIMHAN K, SALIMANS T, et al. Improving language understanding by generative pre-training[J/OL]. https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/languageunsupervised/language understanding paper.pdf.

[3] DEVLIN J, CHANG M W, LEE K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv: 1810.04805, 2018.

[4] GOODFELLOW I J, SHLENS J, SZEGEDY C. Explaining and harnessing adversarial examples[J]. arXiv: 1412.6572, 2014.

[5] MIYATO T, DAI A M, GOODFELLOW I. Adversarial training methods for semi-supervised text classification[J]. arXiv: 1605.07725, 2016.

[6] WANG D, GONG C, LIU Q. Improving Neural Language Modeling via Adversarial Training[C]// International Conference on Machine Learning. 2019.

[7] BENGIO Y, DUCHARME R, VINCENT P, et al. Aneural probabilistic language model[J]. Journal of Machine Learning Research, 2003, 3(Feb): 1137-1155.

[8] KIM Y. Convolutional neural networks for sentence classification[J]. arXiv: 1408.5882, 2014.

[9] SZEGEDY C, ZAREMBA W, SUTSKEVER I, et al. Intriguing properties of neural networks[J]. arXiv: 1312.6199, 2013.

[10] DAI A M, LE Q V. Semi-supervised sequence learning[C]// Advances in Neural Information Processing Systems. 2015: 3079-3087.

[11] QIAO C, HUANG B, NIU G, et al. A New Method of Region Embedding for Text Classification[C]// ICLR. 2018.

[12] LETARTE G, PARADIS F, GIGUÈRE P, et al. Importance of self-attention for sentiment analysis[C]// Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP. 2018: 267-275.

[13] WANG B. Disconnected recurrent neural networks for text categorization[C]// Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2018: 2311-2320.

[14] NIU G, XU H, HE B, et al. Enhancing Local Feature Extraction with Global Representation for Neural Text Classification[C]// Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 2019: 496-506.

[15] JOULIN A, GRAVE E, BOJANOWSKI P, et al. Bag of tricks for efficient text classification[J]. arXiv: 1607.01759, 2016.

[16] SHEN D, WANG G, WANG W, et al. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms[J]. arXiv: 1805.09843, 2018.

[17] CHEN W, SU Y, SHEN Y, et al. How large avocabulary does text classification need? a variational approach to vocabulary selection[J]. arXiv: 1902.10339, 2019.


**ZHANG Xiao-hui**, born in 1995, postgraduate, is a member of China Computer Federation. Her main research interest includes natural language processing.

**WANG Quan-xin**, born in 1984, master's degree, lecturer. Her main research interests include Java and database.