中国科学: 信息科学 2020 年 第 50 卷 第 6 期: 813–823
SCIENTIA SINICA Informationis
《中国科学》杂志社
SCIENCE CHINA PRESS
大数据分析专题 · 论文

一种保持语义关系的词向量复用方法

李新春, 詹德川*

计算机软件新技术国家重点实验室 (南京大学), 南京 210023
* 通信作者. E-mail: zhandc@nju.edu.cn

收稿日期: 2019–12–24 ; 修回日期: 2020–03–14; 接受日期: 2020–04–14 ; 网络出版日期: 2020–06–01

国家重点研发计划 (批准号: 2018YFB1004300) 和国家自然科学基金 (批准号: 61773198, 61632004) 资助项目

摘要 在基于深度网络的自然语言处理任务中, 嵌入表示层用词向量刻画词的语义信息, 可以有效地提升模型性能. 词向量可以和当前任务一起端到端地进行学习, 但是从模型参数数量的角度来看, 词向量的训练很容易在小语料库上过拟合. 为了解决这个问题, 通常会使用在大语料库上预训练得到的词向量. 首先, 本文总结了几种常见的复用预训练词向量的方法. 其次, 由于当前任务的变化, 会有一些新词出现, 这些新词的词向量不能通过预训练的词向量获得, 因此本文提出了一种保持语义关系的词向量复用算法 (SrpWer). SrpWer 首先对当前数据集中词语之间的关系进行建模, 然后结合预训练的词向量生成新词对应的词向量. 实验结果验证了 SrpWer 的有效性.

关键词 自然语言处理, 词向量, 模型复用, 新词, 深度学习

1 引言

自然语言处理 [1,2] 的主要研究内容是赋予机器像人一样理解语言的能力, 比如从文字中挖掘出人的情感 [3], 语言翻译 [4] 等. 随着深度学习的发展, 使用深度网络对自然语言进行建模成为了当下最常用并且最有效的方式. 但是, 深度网络处理的是数值型数据, 不能直接用来处理文本. 英文里的单词或者中文里的汉字, 作为文本的基本单位, 往往会在神经网络的最底层进行处理. 一般的做法是将单词映射为一个固定长度的向量, 即词向量 [5,6]. 经过底层的嵌入表示层, 然后就可以使用卷积神经网络 [7,8] 或者循环神经网络 [9,10] 对文本进行建模, 示意图如图 1 所示.

深度网络的训练需要大量的数据, 因为网络中的大量参数会很容易在小数据集上过拟合. 如图 1 所示, 词向量本身也是模型的一部分, 参数数量更为庞大. 举例而言, 假设词库里有 20000 个词, 每个词映射到一个 100 维的向量, 那么嵌入表示层的参数数量就达到了 2000000. 作为参照, 一个隐层大小为 128 的双向循环神经网络再加上几个隐层为 128 的全连接层, 其参数数量在 200000 左右. 当词向

引用格式: 李新春, 詹德川. 一种保持语义关系的词向量复用方法. 中国科学: 信息科学, 2020, 50: 813–823, doi: 10.1360/SSI-2019-0284
Li X C, Zhan D C. A semantic relation preserved word embedding reuse method (in Chinese). Sci Sin Inform, 2020, 50: 813–823, doi: 10.1360/SSI-2019-0284

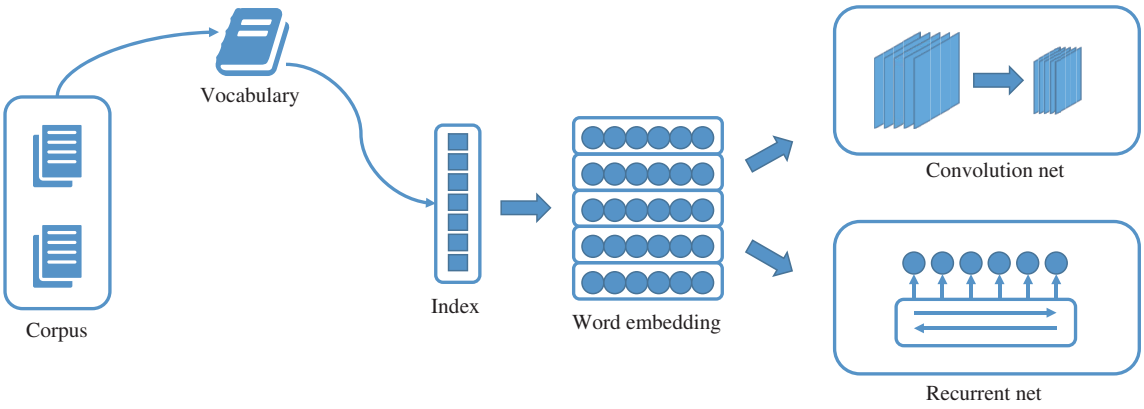


图 1 (网络版彩图) 深度自然语言处理任务框架示意图

Figure 1 (Color online) The illustration of the deep NLP framework

量参数的数量是任务本身数量的 10 倍左右或者更高时, 联合词向量一起端到端进行训练会有一定的困难.

为了解决上文陈述的新任务下深度网络训练的难点, 使用大语料库上预训练的词向量作为当前任务词向量的初始化是首选的方法. 使用不同的语料库, 不同的算法会得到不同的预训练词向量模型. 自从 2013 年以来, 已经有很多训练好的词向量模型可以供新的任务复用. 如何利用这些预训练的词向量辅助当前任务的学习, 和 Zhou [11] 提出的学件 (Learnware) 非常吻合, 二者都是在探讨一种最佳的模型复用方式.

Learnware 强调使用少量数据, 最高效地重用已有的模型, 从而可以适应环境的变化, 辅助当前任务的学习. 在词向量复用过程中会遇到文本领域主题变化的情况. 最简单的一种情况就是新词的出现, 在当前任务的语料库中出现了之前词库里面不曾见到的词, 这些新词对应的预训练的词向量无法获取. 本文提出了一种简单可行的方法来生成新词对应的词向量. 具体地说, 本文提出的方法是一种保持语义关系的词向量复用算法 (SrpWer), 先利用当前语料库建模得到词语之间的语义关系映射, 然后在预训练词向量的基础上施加该映射得到新词的词向量. SrpWer 综合考虑了预训练词向量可以带来的增益效果和当前任务中词语的语义关系, 从而可以获得更好的性能表现.

解决新词出现的问题和增量学习 [12,13]、持续学习 [14] 等研究领域有较高的关联. 增量学习致力于解决目标任务中出现了数据分布不断变化的问题, 即概念漂移问题. 概念漂移经常会导致模型出现灾难性遗忘, 即模型拟合新的数据分布的同时忘记了已经学习到的知识. iCaRL [12] 等方法可以很好地解决遗忘问题. 除了数据分布逐渐变化, 新类的出现也是增量学习需要考虑的问题, IL2M [15] 等方法对新类别进行了检测并建模. 在本课题研究中, 主要关注新词的词向量表示, 与增量学习有不同的侧重点.

本文第 2 节对词向量、词向量复用和新词处理的相关研究工作进行简要介绍. 第 3 节提出并详细阐释了 SrpWer 的算法. 第 4 节在几个具体的文本分类和情感识别任务上进行了相应的实验, 验证了 SrpWer 的有效性. 最后是总结, 对全文以及未来词向量复用的发展方向进行了探讨.

2 相关工作

本节主要介绍词向量的发展历程, 词向量复用的相关研究以及如何处理新词的出现.

2.1 词向量

词向量的本质是把词映射为一个向量, 通过向量之间的操作可以反映词语之间的关系. 比如, 词语的相关程度可以通过两个向量的夹角来衡量, 词向量之间的加减操作具有实验以及理论上的解释 [16].

传统的自然语言处理任务中就出现了词向量的概念, 这些方法经常假设在同一个文档里面共现次数比较多的词语之间具有较高的关联性, 比如潜在语义索引 [17] 使用词 – 文档矩阵分解来获得词和文档的向量表示. 这些方法大多都是基于向量空间模型, 仅仅是将文档看成一个词袋, 没有考虑词语之间的语义关系.

相比较于传统的方法, Bengio 等 [18] 提出了一种基于神经语言概率语言模型 (NPLM) 的方法对句子进行建模, 其中间产物就是词向量. NPLM 利用 N-Gram 进行建模, 使用前 N 个词来预测下一个词出现的概率, 考虑了词语之间的近邻和顺序关系. Mikolov 等 [5, 19] 提出了基于分布式表示的 CBOW 和 Skipgram 模型, 认为上下文相近的两个词具有较强的相关性, 这两种方法是词向量发展的里程碑, 统称为 Word2Vec. 基于词语间的共现关系, Pennington 等 [6] 提出了 Glove 模型, 利用深度网络方法对全局语义关系进行建模.

随着网络结构的多样化和复杂化, ELMO [20], BERT [21] 和 GPT [22] 等依次被提出当做词向量预训练模型. 由于这些模型提供的词向量具有动态性、拟合能力强、支持无监督预训练等优势, 很多自然语言处理任务会使用这些模型提取的特征当做初始化. 但是这些模型需要非常大的计算资源, 通常需要 TPU 或者 GPU 的支持.

2.2 词向量复用和新词处理

在大量语料库上预训练的词向量往往具有较高质量的语义信息. 在预训练的语料库涉及的主题内容足够丰富、词库足够大的情况下, 预训练的词向量是可以直接应用到新任务上的. 但是, 现实场景下经常会出现主题变化、新词出现等问题. 举例而言, 在维基百科上预训练的词向量可能不太适用于某项产品的智能客服对话系统, 尤其是新的产品.

为了综合考虑当前语料库的场景信息, 经常需要对预训练的词向量在当前任务上进行微调. 为了减少微调词向量造成的过拟合问题, 一些方法会限制词向量的更新速度, 比如词向量部分的参数使用一个较小的学习率. 此外, 词向量复用需要考虑动态变化的场景. 第 1 种最常见的场景是机器翻译领域中多个语种的词向量学习, 在保持翻译任务中词语对齐的同时, 学习多个语种中对应单词的向量表示 [23, 24]. 另外一种常见的场景是跨领域的词向量复用, 比如 Yang 等 [25] 提出了一种基于正则化的方法来重用之前预训练的词向量. 词向量的学习也可以通过多任务学习的方式来相互提升表示能力 [26].

随着新的语料库的到来, 一些新的热点词汇会出现. 如何基于原有的词向量模型进行增量学习是一个新的研究方向. Peng 等 [27] 在层级 Softmax 上增量训练词向量, 动态扩充层级分类树的结构. Kaji 等 [28] 基于负采样和 Skip-gram 模型进行增量训练词向量. Kabbach 等 [29] 提出了一种度量上下文信息量的方法, 并据此改进了在线学习的向量加和模型. Hu 等 [30] 采用了元学习 MAML 算法来学习高阶层次的知识, 继而迁移到新的应用场景下, 为新词生成词向量.

本文工作主要基于几种词向量的复用方式进行了实验比较, 并在此基础上提出了一种简单的可以处理新词问题的复用方法. 实验结果验证了方法的有效性.

3 提出的方法

本文针对新的场景下会有新词出现的问题, 提出了一种简单实用的复用方法. 具体而言, 记当前任务的语料库为 $\mathcal{D}$, 语料库里面有多篇文档, 对应的词库为 $V=\{v_j\}_{j=1}^{n}$, 包含 n 个不同的词. 通常来说, 词库的选择是选取语料库中出现频次最高的 n 个词. 记预训练的词向量和词库分别为 $\boldsymbol{W}_{\mathrm{P}}\in\mathbb{R}^{n_p\times d}$ 和 $V_{\mathrm{P}}=\{v_i\}_{i=1}^{n_p}$. 预训练的词库中有 n_{p} 个词, 每个词对应的词向量的维度是 d, 一般 d 会设置为 100 或者 200 左右. 由于新兴热点的出现, 当前语料库的词库和预训练词库存在一些差别, 因此经常会有 $V\neq V_{\mathrm{P}}$. 在新词出现的场景下, 记公共词为 $V_{\mathrm{I}}=V_{\mathrm{P}}\cap V$, 即当前词库和预训练的词库的交集. 同样地, 当前语料库产生的新词则是 $V_{\mathrm{O}}=V\setminus V_{\mathrm{I}}$, 其中 "\" 表示集合减操作.

为了复用之前预训练的词向量, 可以将之前预训练的公共词 V_{I} 对应的部分迁移到当前任务上, 但是对于新词 V_{O} 部分, 只能采用随机初始化. 将预训练的公共部分词向量和随机初始化的新词的词向量拼接在一起之后, 一些传统的方法会被使用, 比如固定住词向量不变或者进行微调. 然而, 本文提出的保持语义关系的词向量复用算法 (SrpWer) 可以根据当前语料库语义关系和预训练的词向量生成新词对应的词向量, 而不仅仅是简单使用随机初始化. 简单来说, SrpWer 可以同时达到 3 个目标:

(1) 考虑当前任务中语料库里的语义关系.

(2) 充分利用之前在大语料库上预训练的词向量.

(3) 为新词生成保持语义关系的词向量, 而不是随机初始化.

首先, 为了建模当前语料库上的语义关系, 可以选择在当前语料库上使用矩阵分解 LSI [17]、神经网络模型 NPLM [18] 或者 Word2Vec [5, 19] 等训练一组词向量 $\boldsymbol{W}\in\mathbb{R}^{n\times d}$. 毫无疑问, $\boldsymbol{W}$ 可以表示当前语料库中词语之间的语义关系, 比如通过向量之间的加减和内积操作等反映词语之间的关联程度. 接下来, 将公共词和新词对应的词向量部分取出, 记为 $\boldsymbol{W}_{\mathrm{I}}\in\mathbb{R}^{n_{\mathrm{I}}\times d}, \boldsymbol{W}_{\mathrm{O}}\in\mathbb{R}^{n_{\mathrm{O}}\times d}$, 其中 n_{I} 和 n_{O} 分别指公共词和新词的数量, 有 $n_{\mathrm{I}}+n_{\mathrm{O}}=n$. SrpWer 会根据 $\boldsymbol{W}_{\mathrm{I}}$ 和 $\boldsymbol{W}_{\mathrm{O}}$ 建模出公共词和新词在当前语料库中的语义关联. 同样地, 记对应的预训练词向量里公共词对应的部分为 $\boldsymbol{W}_{\mathrm{PI}}\in\mathbb{R}^{n_{\mathrm{I}}\times d}$. 那么如何获得新词的词向量就抽象成如下数学形式, 其中 $\boldsymbol{W}_{\mathrm{PO}}\in\mathbb{R}^{n_{\mathrm{O}}\times d}$ 指的是新词的词向量:

$$\boldsymbol{W}_{\mathrm{PO}}=f(\boldsymbol{W}_{\mathrm{I}},\boldsymbol{W}_{\mathrm{O}},\boldsymbol{W}_{\mathrm{PI}}). \tag{1}$$

如何选择一个 f, 既可以维持当前语料库下的语义关系, 又同时尽可能地复用之前预训练的词向量呢? 首先, 为了维持当前语料库的语义信息, 根据词向量之间的线性关系 [16], 假设存在一个映射矩阵 $\boldsymbol{Z}\in\mathbb{R}^{n_{\mathrm{I}}\times n_{\mathrm{O}}}$ 可以将公共词的词向量映射到新词的词向量, 那么最简单直观的是下面的优化目标:

$$\boldsymbol{Z}^{\star}=\arg\min_{\boldsymbol{Z}}\left\|\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{W}_{\mathrm{I}}-\boldsymbol{W}_{\mathrm{O}}\right\|_{\mathrm{F}}^{2}+\lambda\|\boldsymbol{Z}\|_{\mathrm{F}}^{2}, \tag{2}$$

其中 λ 是正则化项权重. 上面的优化目标的直观含义就是在当前语料库上寻找到最优的映射矩阵, 建模当前语料库下公共词和新词的语义关系. 然后我们假设这种语义映射也可以施加到预训练的词向量上, 使用该映射矩阵对预训练的公共词的词向量部分进行转换得到

$$\boldsymbol{W}_{\mathrm{PO}}=\boldsymbol{Z}^{\star\mathrm{T}}\boldsymbol{W}_{\mathrm{PI}}. \tag{3}$$

通过上面的代数运算, $\boldsymbol{W}_{\mathrm{PO}}$ 既可以将当前语料库的语义信息考虑进建模过程, 又可以复用预训练的词向量, 最终可以生成新词对应的词向量, 是一种非常简单直观的方法. 最后将 $\boldsymbol{W}_{\mathrm{PI}}, \boldsymbol{W}_{\mathrm{PO}}$ 两部分词向量拼接就可以得到适应当前语料库的预训练词向量. 拼接方法为

$$\hat{\boldsymbol{W}}=[\boldsymbol{W}_{\mathrm{PI}};\boldsymbol{W}_{\mathrm{PO}}]\in\mathbb{R}^{(n_{\mathrm{I}}+n_{\mathrm{O}})\times d}. \tag{4}$$

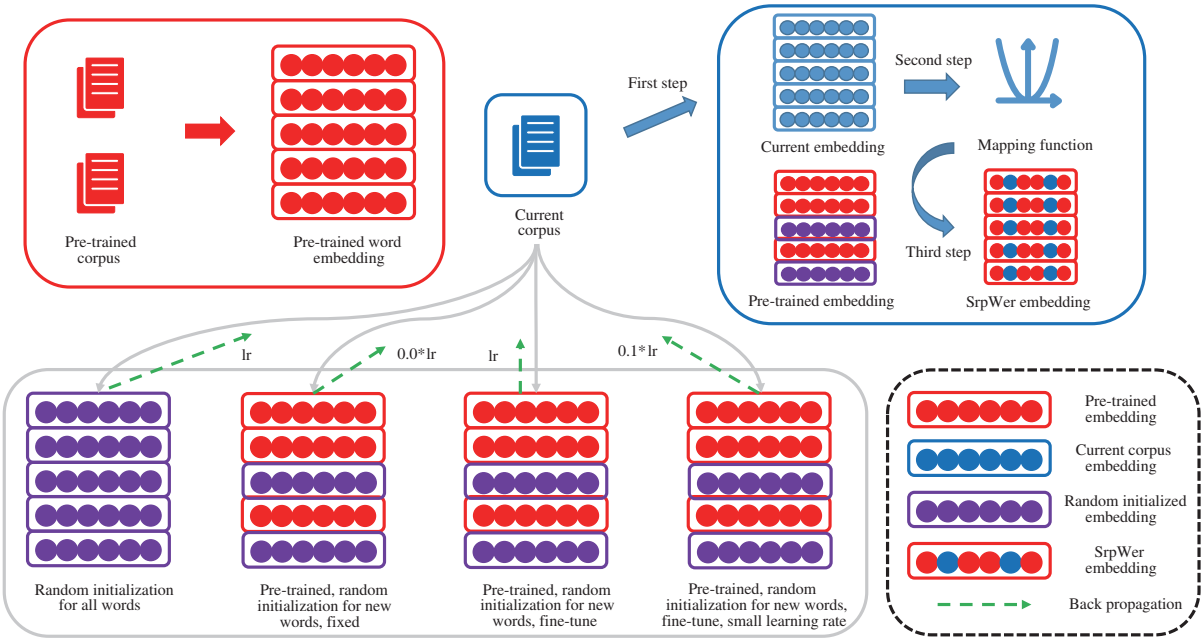


图 2 (网络版彩图) SrpWer 和传统方法对比示意图

Figure 2 (Color online) The illustration of the comparison between SrpWer and other conventional methods

这里值得一提的是, 为什么不直接使用 $\boldsymbol{W}_{\mathrm{PO}} = \boldsymbol{W}_{\mathrm{O}}$ 呢? 因为我们假设当前语料库不足以训练出一个较好的词向量模型, 并且通常情况下, 新词在当前语料库中出现的频次比较低, 那么其对应的词向量质量更难保证. 但是当前语料库里面的语义关系是相对容易得到的. 举例而言, 对于 “5G”, 作为一个新的词汇, 只在少量有限的场景中才会出现, 因此想得到针对其本身的一个向量描述很困难. 但是, 我们可以发现 “5G” 经常和 “网络”、“自动驾驶” 等词一起出现, 和 “生物”、“体育” 等共现频次很低. 因此我们可以通过这种简单的共现关系, 使用预训练的 “网络”、“自动驾驶”、“生物”、“体育” 等词的词向量以线性加权的方式推测出 “5G” 的词向量表示.

SrpWer 的示意图可以参考图 2, 图中红色框部分表示了大语料库上预训练词向量的过程, 一行红色实心圆代表一条预训练的词向量. 灰色框列举了几种复用词向量的方法, 第 1 种不使用预训练的词向量, 使用随机初始化的词向量, 用紫色表示; 其余 3 种表示对于公共词使用预训练的词向量 (红色部分), 新词采用随机初始化 (紫色部分), 但是所使用的学习率方法不同 (图中绿色虚线箭头代表反向传播). 右上角蓝色框里面介绍了 SrpWer 的算法流程示意图, 具体的算法细节可以参考算法 1.

4 实验部分

本节详细比较和分析使用不同的预训练词向量, 在不同任务场景下几种模型复用方法的性能表现.

4.1 实验设置

由于不同的语料库, 不同的算法会生成不同的预训练词向量. 因此实验选取了两种预训练的词向量: WIKI-Glove 和 IMDB-SG. 详细描述如下:

算法 1 保持语义关系的词向量复用算法 (SrpWer)

输入:

预训练词向量 $\boldsymbol{W}_{\mathrm{P}} \in \mathbb{R}^{n_{\mathrm{P}} \times d}$;
预训练语料的词库 $V_{\mathrm{P}} = \{v_i\}_{i=1}^{n_p}$;
当前语料库 $\mathcal{D}$;
当前词库 $V = \{v_j\}_{j=1}^{n}$;
正则化项系数 λ.

主迭代:

Skipgram$(\mathcal{D}) \rightarrow \boldsymbol{W} \in \mathbb{R}^{n \times d}$;
$V_{\mathrm{I}} = V_{\mathrm{P}} \cap V$, $V_{\mathrm{O}} = V \setminus V_{\mathrm{I}}$;
$\boldsymbol{W}_{\mathrm{I}} = \mathrm{lookup}(\boldsymbol{W}, V_{\mathrm{I}})$, $\boldsymbol{W}_{\mathrm{O}} = \mathrm{lookup}(\boldsymbol{W}, V_{\mathrm{O}})$;
$\boldsymbol{W}_{\mathrm{PI}} = \mathrm{lookup}(\boldsymbol{W}_{\mathrm{P}}, V_{\mathrm{I}})$;
$\boldsymbol{Z}^{\star} = \arg\min_{\boldsymbol{Z}} \left\| \boldsymbol{Z}^{\mathrm{T}} \boldsymbol{W}_{\mathrm{I}} - \boldsymbol{W}_{\mathrm{O}} \right\|_{\mathrm{F}}^2 + \lambda \|\boldsymbol{Z}\|_{\mathrm{F}}^2$;
$\boldsymbol{W}_{\mathrm{PO}} = \boldsymbol{Z}^{\star \mathrm{T}} \boldsymbol{W}_{\mathrm{PI}}$;
$\hat{\boldsymbol{W}} = [\boldsymbol{W}_{\mathrm{PI}}; \boldsymbol{W}_{\mathrm{PO}}]$.

输出: $\hat{\boldsymbol{W}} \in \mathbb{R}^{n \times d}$.

• WIKI-Glove [6] 是使用 Glove 方法在 Wikipedia 2014 和 Gigaword 5 上训练的词向量, 词向量维度大小是 100 维.

• IMDB-SG [31] 是在 IMDB 数据集上使用 Skip-gram 模型预训练的词向量, 词向量维度大小是 200 维.

实验选择了文本分类和情感识别两种任务, 包括 3 个数据集: News20, Yelp13 和 IMDB. 详细描述如下 (文本最大长度是指将文本截断或填充到同等长度):

• News20 是新闻文本分类的数据集, 类别数目为 20, 每个文本最大长度设置为 1000 个词.

• Yelp13 是情感识别数据集, 情感等级数目为 5, 每个文本最大长度设置为 400 个词.

• IMDB 是情感识别数据集, 情感等级数目为 10, 每个文本最大长度设置为 400 个词.

根据图 1 所示, 实验选择了两种经典的网络框架: CNN 和 GRU. 详细设置如下:

• CNN 采用三层一维卷积, 卷积核大小是 3, 步长都设置为 1; 每层卷积核后面接一层最大池化层; 中间层通道数目都设置为 128; 激活函数采用 ReLU.

• GRU 采用隐层大小为 128 的一层双向循环神经网络, 最后得到的双层表示拼接在一起.

对于一段文本, 先做一些去除特殊符号、小写转换等预处理步骤, 然后进行分词, 并根据最大文本长度截断或填充得到词的列表. 对于不在词库中的词, 统一设置为 "UNK"; 对于填充的词, 统一设置为 "PAD". 在具体的深度网络前向过程中, 先通过嵌入表示层得到文本中所有词对应的向量表示, 然后经过 CNN 或者 GRU 得到相应的输出, 采用最大化聚合的方式获得一个最终向量, 然后接一个两层全连接进行分类. 分类损失采用交叉熵损失. 学习率设置为 0.01, 学习率每隔 2 遍训练数据集衰减为之前的 0.5 倍, 批大小设置为 128.

对于具体的词向量复用方式, 有以下 7 种方法, 方法示意图可以参照图 2. 详细介绍如下:

(1) NoPT 不使用预训练词向量.

(2) PT-NoFT 使用预训练词向量, 但是不微调.

(3) PT-FT 使用预训练词向量, 并微调.

(4) PT-FT-Mu 使用预训练词向量, 微调, 但是词向量对应的学习率会乘以 0.1.

(5) SrpWer-NoFT 使用 SrpWer 得到的词向量, 但是不微调.

(6) SrpWer-FT 使用 SrpWer 得到的词向量, 并微调.

表 1 预训练的词库与当前任务词库交叠情况统计表

Table 1 The statistics among vocabularies of pretrained corpus and current tasks

	IMDB		News20		Yelp13	
	n_{I}	n_{O}	n_{I}	n_{O}	n_{I}	n_{O}
WIKI-Glove	19911	89	17902	2038	18965	1035
IMDB-SG	19976	24	13616	6384	18856	1144

表 2 基于 WIKI-Glove 词向量的 3 种任务使用不同复用方法的效果对比表

Table 2 Performance comparisons of different usages on three NLP tasks based on WIKI-Glove embeddings

	CNN			GRU		
	IMDB	News20	Yelp13	IMDB	News20	Yelp13
NoPT	0.318	0.416	0.508	0.429	0.684	0.596
PT-NoFT	0.309	0.444	0.553	0.476	0.777	0.609
PT-FT	0.336	**0.741**	0.587	0.473	**0.816**	0.603
PT-FT-Mu	0.321	0.665	0.565	0.471	0.809	0.615
SrpWer-NoFT	0.322	0.648	0.598	0.472	0.806	0.623
SrpWer-FT	**0.369**	0.719	**0.626**	0.469	0.805	0.612
SrpWer-FT-Mu	0.350	0.677	0.624	**0.480**	0.809	**0.631**
Improve	+0.033	−0.022	+0.039	+0.004	−0.007	+0.016

表 3 基于 IMDB-SG 词向量的 3 种任务使用不同复用方法的效果对比表

Table 3 Performance comparisons of different usages on three NLP tasks based on IMDB-SG embeddings

	CNN			GRU		
	IMDB	News20	Yelp13	IMDB	News20	Yelp13
NoPT	0.293	0.553	0.541	0.450	0.703	0.614
PT-NoFT	0.330	0.578	0.551	0.499	0.734	0.628
PT-FT	0.338	**0.745**	0.578	0.466	0.819	0.605
PT-FT-Mu	0.340	0.652	0.574	0.485	0.812	0.618
SrpWer-NoFT	0.353	0.566	0.598	0.481	0.786	0.613
SrpWer-FT	0.350	0.686	0.595	0.469	**0.819**	**0.642**
SrpWer-FT-Mu	**0.373**	0.652	**0.634**	**0.503**	0.802	0.641
Improve	+0.033	−0.059	+0.056	+0.004	+0.000	+0.014

(7) SrpWer-FT-Mu 使用 SrpWer 得到的词向量, 微调, 但是词向量对应的学习率会乘以 0.1.

具体的, 针对当前任务中出现的新词, 前 4 种方法会设置新词对应的词向量为随机初始化的向量, 后面 3 种则使用 SrpWer 得到的新词词向量. 当前任务上的词库选择是默认选择频次最高的 20000 个词. 3 个任务的词库与两个预训练词向量词库的交叠情况见统计表 1, 其中 n_{I} 和 n_{O} 分别指公共词和新词的数目.

4.2 实验结果和分析

基于 WIKI-Glove 和 IMDB-SG 两种预训练的词向量得到的实验结果见表 2 和 3. 表格的每一行

表 4　基于 IMDB-SG 词向量在不同新词比例的 News20 上使用 GRU 复用性能对比

Table 4　Performance comparisons of varying new words' ratios on News20 task based on IMDB-SG embeddings and GRU network

	0.01	0.05	0.10	0.15	0.20	0.30	0.40
SrpWer-NoFT	0.811	0.807	**0.816**	0.812	0.803	0.798	0.794
SrpWer-FT	0.822	**0.832**	0.828	0.817	0.819	0.812	0.817
SrpWer-FT-Mu	0.816	0.821	**0.826**	0.815	0.812	0.802	0.805

代表了一种复用方法在不同当前任务上的结果, 最后一行代表了 SrpWer 相较于对比方法最大的性能变化. 每个表格分为两栏, 分别表示网络架构为 CNN 和 GRU 的结果. 具体到每一列则是每个任务数据集上的性能表现. 从以上表格的实验结果分析, 不使用预训练词向量 NoPT 的性能表现会很差, 验证了联合词向量端到端进行训练会比较困难. 使用 SrpWer-FT 和 PT-FT 会得到较优的结果, 说明了在当前任务上使用预训练词向量并微调的方式会带来比较好的性能表现. 同时, 对于预训练的词向量使用较小的学习率有时也会取得更好的结果, 比如使用 IMDB-SG 时 SrpWer-FT-Mu 一般会取得最好的性能提升效果.

此外, 从使用的预训练词向量角度来看, 使用 IMDB-SG 的效果会比 WIKI-Glove 高, 可能是因为语料库、算法或者词向量维度大小的影响. 但是本实验中主要是算法的影响, 因为 WIKI 语料库比 IMDB 大很多, 但是 Skipgram 算法更能把握词语的分布式关系和局部语义信息, 因此 IMDB-SG 的效果会稍微显著.

使用 SrpWer 得到的词向量一般会有更好的性能, 因为 SrpWer 在新词词向量上不仅仅是采用随机初始化, 而是考虑了当前任务的语义关系. 但是, SrpWer 在 News20 上性能表现提升不是很明显, 结合表 1 的统计结果, 分析原因可能是因为新词数目太多, 导致当前语料库中语义关系难以准确建模.

为了验证新词数目是否会影响到当前语料库语义关系的建模, 对使用 IMDG-SG 表示的 News20 新词的数目进行控制, 具体操作为: 根据表 1 中统计的 6384 个新词在 News20 中出现的频次进行排序, 按照频次从大到小筛选出固定比例 $R = \frac{n_{\mathrm{O}}}{n_{\mathrm{I}}} \in \{0.01, 0.05, 0.10, 0.15, 0.20, 0.30, 0.40\}$ 的词语, 其余低频的新词语舍弃掉, 验证所提算法的有效性. 实验结果如表 4. 通过表格可以发现, 当新词数目和已有词数目的比例在 0.05 到 0.15 之间时, 所提算法性能具有较好的性能表现, 说明了新词比例对 SrpWer 的确有影响. 这项实验提供了一个经验性的结论: 当新词数目过多时, SrpWer 在本地语料库上使用简单的线性关系建模出的新词词向量并不是很准确, 在下游任务中, 模型的学习和训练比较困难. 但是当新词数目较少 (比如本实验中新词比例 0.1 左右) 时可以得到比较好的新词词向量初始化, 可以更好地提升模型性能.

总的来说, 本文提出的 SrpWer 可以在新词数目不是太多的情况下解决词向量复用过程中出现的新词问题, 给词向量复用提供了一种新的、简单易用的途径. 另外, 由于使用 SrpWer 在 Yelp13 上性能提升比较显著, 将其学习过程中训练集和测试集上的损失曲线绘制成图 3. 从图中可以看出, NoPT 在测试集上的损失在上升, 这说明训练过程中存在着过拟合现象, 和之前的分析一致. 此外, 从这两组损失变化曲线图中可以看出, 使用 SrpWer 复用的损失一般会在训练集上更快地下降, 这说明 SrpWer 可以更好地辅助当前任务的学习, 加快收敛速度. 总之, 使用 SrpWer 的损失可以在测试集上更低, 也验证了 SrpWer 的有效性.

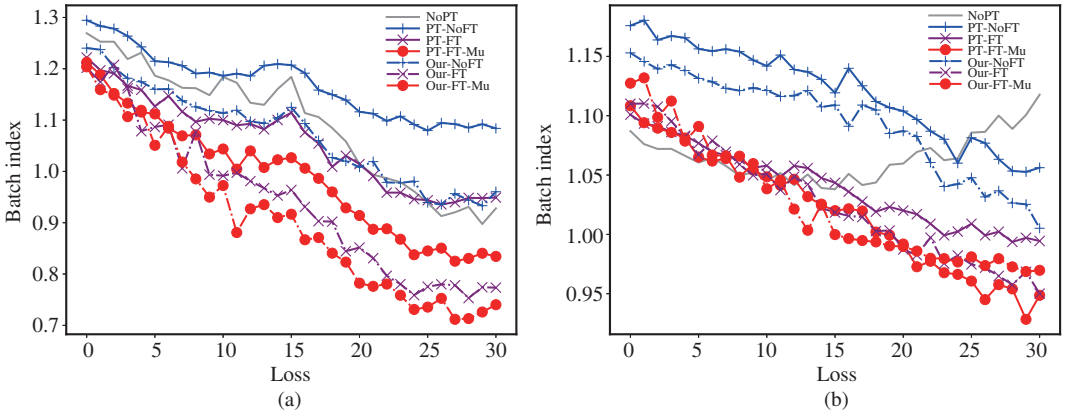


图 3 (网络版彩图) 基于 **WIKI** 词向量的 **Yelp13** 的学习曲线

Figure 3 (Color online) Learning curves for different usages based on Yelp13 task and WIKI embeddings. (a) Training loss; (b) test loss

5 总结

本文基于深度自然语言处理任务中词向量的复用方式展开探讨. 首先总结并对比了几种词向量复用方法, 然后针对新任务中出现新词的问题提出了一种简单易用的解决方法 SrpWer. SrpWer 可以同时考虑当前任务的语义信息和预训练的词向量进行建模, 为新词生成语义信息更为丰富的词向量, 而不仅仅是随机初始化. 在多个数据集上的实验效果验证了 SrpWer 的有效性.

本文的实验结果中也有一些有趣的现象, 比如在使用不同的预训练词向量时, 在同一任务上使用某一特定的词向量复用方法也会带来差别很大的性能表现. 这其实是词向量复用方法的核心难点: 如何基于不同的预训练词向量、不同的网络架构和当前任务的类型, 自适应地选择最合适的词向量复用方式. 一方面, 这个难点也是 Learnware 要解决的核心问题之一, 如何为当前任务适配一个最优的模型. 另一方面, 可以基于元学习或者可迁移性度量选择合适的词向量, 继而确定一种最佳的词向量复用方式. 探索这些可行性方案是未来的研究工作.

参考文献

1 Otter D W, Medina J R, Kalita J K. A survey of the usages of deep learning in natural language processing. 2018. ArXiv: 1807.10854

2 Young T, Hazarika D, Poria S, et al. Recent trends in deep learning based natural language processing. 2017. ArXiv: 1708.02709

3 Zhang L, Wang S, Liu B. Deep learning for sentiment analysis: a survey. Wiley Interdiscip Rev Data Min Knowl Discov, 2018, 8: e1253

4 Zhu M, Pan P, Chen W, et al. DM-GAN: dynamic memory generative adversarial networks for text-to-image synthesis. In: Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, 2019. 5802–5810

5 Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space. In: Proceedings of the 1st International Conference on Learning Representations, Scottsdale, 2013

6 Pennington J, Socher R, Manning C D. Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, 2014. 1532–1543

7 Kim Y. Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, 2014. 1746–1751

8 Gu X, Gu Y, Wu H. Cascaded convolutional neural networks for aspect-based opinion summary. Neural Process Lett, 2017, 46: 581–594

9 Irsoy O, Cardie C. Opinion mining with deep recurrent neural networks. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, 2014. 720–728

10 Zhang X, Lapata M. Chinese poetry generation with recurrent neural networks. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, 2014. 670–680

11 Zhou Z H. Learnware: on the future of machine learning. Front Comput Sci, 2016, 10: 589–590

12 Sylvestre-Alvise R, Alexander K, Georg S, et al. iCaRL: incremental classifier and representation learning. In: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, 2017. 5533–5542

13 Kibok L, Kimin L, Jinwoo S, et al. Overcoming catastrophic forgetting with unlabeled data in the wild. In: Proceedings of 2019 International Conference on Computer Vision, Seoul, 2019. 312–321

14 Matthias D L, Rahaf A, Marc M, et al. Continual learning: a comparative study on how to defy forgetting in classification tasks. 2019. ArXiv: 1909.08383

15 Eden B, Adrian P. IL2M: class incremental learning with dual memory. In: Proceedings of the 2019 International Conference on Computer Vision, Seoul, 2019. 583–592

16 Ethayarajh K, Duvenaud D, Hirst G. Towards understanding linear word analogies. In: Proceedings of the 57th Conference of the Association for Computational Linguistics, Florence, 2019. 3253–3262

17 Papadimitriou C H, Raghavan P, Tamaki H, et al. Latent semantic indexing: a probabilistic analysis. J Comput Syst Sci, 2000, 61: 217–235

18 Bengio Y, Ducharme R, Vincent P, et al. A neural probabilistic language model. Mach Learn, 2003, 3: 1137–1155

19 Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality. In: Proceedings of the 27th Annual Conference on Neural Information Processing Systems, Lake Tahoe, 2013. 3111–3119

20 Peters M E, Neumann M, Iyyer M, et al. Deep contextualized word representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Louisiana, 2018. 2227–2237

21 Devlin J, Chang M W, Lee K, et al. BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019. 4171–4186

22 Radford A, Narasimhan K, Salimans T, et al. Improving language understanding by generative pre-training. 2018. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf

23 Chen X, Cardie C. Unsupervised multilingual word embeddings. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, 2018. 261–270

24 Alaux J, Grave E, Cuturi M, et al. Unsupervised hyper-alignment for multilingual word embeddings. In: Proceedings of the 7th International Conference on Learning Representations, 2019

25 Yang W, Lu W, Zheng V W. A simple regularization-based algorithm for learning cross-domain word embeddings. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, 2017. 2898–2904

26 Neill J O, Bollegala D. Semi-supervised multi-task word embeddings. 2018. ArXiv: 1809.05886

27 Peng H, Li J, Song Y, et al. Incrementally learning the hierarchical softmax function for neural language models. In: Proceedings of the 31st AAAI Conference on Artificial Intelligence, 2017. 3267–3273

28 Kaji N, Kobayashi H. Incremental skip-gram model with negative sampling. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, 2017. 363–371

29 Kabbach A, Gulordava K, Herbelot A. Towards incremental learning of word embeddings using context informativeness. In: Proceedings of the 57th Conference of the Association for Computational Linguistics, Florence, 2019. 162–168

30 Hu Z, Chen T, Chang K W, et al. Few-shot representation learning for out-of-vocabulary words. In: Proceedings of the 57th Conference of the Association for Computational Linguistics, Florence, 2019. 4102–4112

31 Tang D, Qin B, Liu T. Document modeling with gated recurrent neural network for sentiment classification. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, 2015. 1422–1432

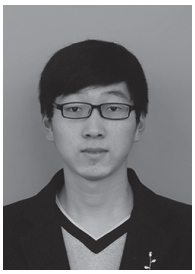
A semantic relation preserved word embedding reuse method

Xinchun LI & Dechuan ZHAN*

National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023*, China*
* Corresponding author. E-mail: zhandc@nju.edu.cn

Abstract When deep learning is applied to natural language processing, a word embedding layer can improve task performance significantly due to the semantic information expressed in word vectors. Word embeddings can be optimized end-to-end with the whole framework. However, considering the number of parameters in a word embedding layer, in tasks with a small corpus, the training set can easily be overfitted. To solve this problem, pretrained embeddings obtained from a much larger corpus will be utilized to boost the current model performance. This paper summarizes several methods to reuse pretrained word embeddings. In addition, as corpus topics change, new words will appear for a given task, and their corresponding embeddings cannot be obtained from pretrained vectors. Therefore, to reuse word embeddings, we propose a semantic relation preserved word embedding reuse method. The proposed method first learns word relations from the current corpus. Then, pretrained word embeddings are utilized to help generate embeddings for new observed words. Experimental results verify the effectiveness of the proposed method.

Keywords natural language processing, word embeddings, model reuse, new words, deep learning



Xinchun LI was born in 1997. He received his B.S. degree from the School of Information Management, Nanjing University, in 2018. He was admitted to study for an M.S. degree at Nanjing University without an entrance examination in 2018. His research interests include machine learning and data mining.

Dechuan ZHAN was born in 1982. He received a Ph.D. degree in computer science from Nanjing University, China, in 2010. In the same year, he became a faculty member in the Department of Computer Science and Technology at Nanjing University, China. He is currently a professor in the School of Artificial Intelligence, Nanjing University. His research interests are mainly in machine learning, data mining, and mobile intelligence.