北京大学学报(自然科学版)
Acta Scientiarum Naturalium Universitatis Pekinensis
doi: 10.13209/j.0479-8023.2015.040

基于自动编码器的短文本特征提取及聚类研究

刘勘† 袁蕴英

中南财经政法大学信息与安全工程学院 武汉 430074; †E-mail: liukan@znufe.edu.cn

摘要 针对短文本的特点，提出一种基于深层噪音自动编码器的特征提取及聚类算法。该算法利用深度学习网络，将高维、稀疏的短文本空间向量变换到新的低维、本质特征空间。算法首先在自动编码器的基础上，引入 L1 范式惩罚项来避免模型过分拟合，然后添加噪音项以提高算法的鲁棒性。实验结果表明，将提取的文本特征应用于短文本聚类时显著提高了聚类的效果，有效解决了短文本空间向量的高维、稀疏问题。

关键词 深度学习；自动编码器；特征提取；聚类

中图分类号 TP391

Short Texts Feature Extraction and Clustering Based on Auto-Encoder

LIU Kan†, YUAN Yunying

School of Information and Safety Engineering, Zhongnan University of Economics and Law, Wuhan 430074; † E-mail: liukan@znufe.edu.cn

Abstract According to the characteristics of short texts, the authors propose a feature extraction and clustering algorithm named deep denoise sparse auto-encoder. The algorithm takes the advantage of deep learning, transforming those high-dimensional, sparse vectors into new, low-dimensional, essential ones. Firstly, L1 paradigm is introduced to avoid overfitting, and the noises is added to improve the robustness. Experimental result shows that applying extracted text features can significantly improve the effectiveness of clustering. It is a valid method to solve the high-dimensional, sparse problem in the short text vector.

Key words deep learning; auto-encoder; feature extraction; clustering

互联网已经成为人们日常生活不可或缺的一部分，越来越多的人习惯于通过微博、新闻网站、论坛等浏览热门话题、了解社会动态、参与热点讨论、发布自己的观点[1]。由于网络的高速与便捷，大部分网络信息都是以短文本的形式存在，这些短文本能让读者快速了解主题内容，准确理解用户观点，又不占用过多的阅读时间。因此，以微博为代表的短文本成为网络信息交流的主要载体。但是由于人本身思维的发散性，发布方式的随意性，短文本的结构往往极其不统一。单条短文本提供的信息十分有限，在处理大量短文本时存在着高度稀疏的问题。如何将海量、不规则、稀疏的短文本有效地组织和分析，成为一个具有挑战性的研究热点。本文将针对短文本特征提取及聚类问题，利用深度学习[2]的思想，采用自动编码器处理技术，提取短文本中的隐含特征，以此为基础得到更准确的短文本聚类结果。

1 相关研究

自动编码器是深度学习中一种重要的训练模型，一直以来，在自然语言处理中取得较好的效果[3−5]，也越来越受到研究人员的重视。Glorot 等[6]在自动编码器算法的基础上添加纠正激活函数，实验结果表明，此方法比传统的 sigmoid 或 tangent 激活函数更能提高文本分类的效果。Glorot 等[7]还使用该自动编码器方法，提取出评论的高层抽象特征，

国家社会科学基金(14BXW033)和教育部人文社会科学基金(11YJAZH060)资助
收稿日期: 2014−07−27; 修回日期: 2014−10−21; 网络出版时间: 2014-12-01 10:20

解决了跨领域的文本分类问题。Lu 等[8]利用深度自动编码器算法，为基于词汇的翻译模型提取到了有效的特征集，并在中英文翻译过程中取得很好的效果。Salahutdinov 等[9]在自动编码器的基础上扩展了 LSA 模型，成功发现隐藏在查询和文档中的层次语义结构。张开旭等[10]则将自动编码器算法运用到中文词性标注的问题中。由此可见，依靠深度学习强大的无监督学习特征的能力，自动编码器能较好地提取文本中的隐含特征，并利用这些特征来解决文本的分析与挖掘问题。本文针对短文本的聚类问题，也首先利用自动编码器来完成文本的特征提取。

由于短文本的词频过低，建立的空间向量往往是高维且稀疏的，为相似度计算带来较大的困难，使文本分析的效果较差。目前的解决方法主要集中在扩充信息方面。如 Fan 等[11]借助搜索引擎扩充文本的信息。Banerjee 等[12]则利用维基百科中的词条信息丰富文本信息。邱云飞等[13]根据文本中包含的 3 种特殊符号对短文本进行特征扩展。Jin 等[14]借助与聚类短文本内容相似的长文本内容，实现短文本的高效聚类。Tang 等[15]等通过机器翻译从其他语言中抽取特征来扩充短文本的特征值。虽然单条短文本的信息较少，仅反映某个小方面的内容，但大量相同主题的短文本聚集在一起，还是能够体现出该类短文本所具有的共性，这可以作为提取关键特征，降低向量维度的另一种思路。杨婉霞等[16]基于该思想提出一种语义和统计特征相结合的短文本聚类算法，其核心是引入语义词典，将相似度较高的词汇进行合并处理，提高了短文本的聚类效率，但这种方法对语义词典的依赖性较大，词典的内容在很大程度上决定了聚类的效果。

本文将延续同类短文本自身包含潜在共性的思路，来解决短文本向量高维、稀疏的问题。与杨婉霞等的词合并方法不同，本文利用的自动编码器算法可以模仿人脑机制，通过非线性组合高维底层特征学习得到低维抽象特征的特性。结合短文本向量的特点，通过添加 L1 范式以避免算法的过度拟合，通过对输入数据进行加噪处理以提高模型的鲁棒性，从而完成从大规模无标注短文本中提取低维有效特征的任务。这样得到的结果受外部因素的影响较小，能够提高聚类的准确度，还能保证计算的高效性。

2 算法流程

2.1 基本思路

基于噪音稀疏的自动编码(Denoise Sparse Auto-Encoder，DSAE)文本聚类算法的基本思想是利用深度学习的自动编码过程，将短文本的高维稀疏向量转化为低维向量，而且学习过程使低维向量包含了文本信息的本质特征，去除了高维中不必要的干扰部分，这样得到的结果用于聚类分析将提高最终的聚类效果。算法分为 5 个过程，首先对短文本进行预处理，构建向量空间模型，每条短文本都会转化成空间中的一个向量；然后将这些高维稀疏向量输入到构造好的深层噪音稀疏自动编码器中学习，经过逐层抽象，提取得到低维抽象的特征向量，在这一部分中还包括正则化过程和加噪过程。最后利用聚类算法得到短文本聚簇结果。具体的流程如图 1 所示。

短文本的预处理包括清洗、分词等处理，得到构成这些短文本的词袋。词袋中的每个词都可以表示为短文本特征向量中的一个度量。如果短文本中出现了该词，就记为 1，否则记为 0。由此，每条短文本都可以表示为空间中的一个向量 x，其表现方式如下所示:

$$\boldsymbol{x}=(t_1,t_2,t_3,...,t_i...,t_m), \tag{1}$$

其中 m 代表词袋中词的总数，t_i 代表该短文本是否包含第 i 个词，如果包含该词，t_1=1，否则

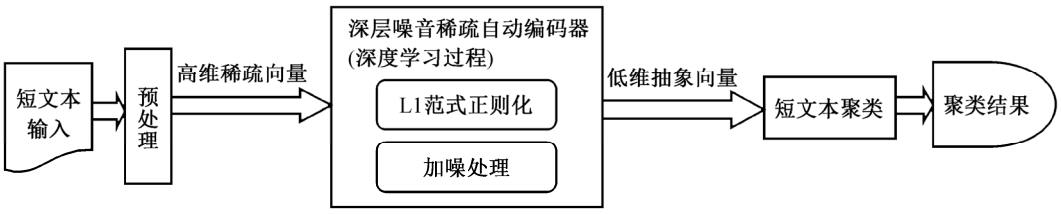

图 1　算法的基本流程图

Fig. 1　Framework for basic idea

t_1=0。

2.2 基本自动编码器

基本的自动编码器接受一个输入向量 $\boldsymbol{x}$ 后, 首先对其进行线性变化, 在激活函数的作用下得到一个编码结果 $\boldsymbol{y}$。选取 sigmoid 函数作为激活函数, 计算如式(2)所示。然后该编码结果 $\boldsymbol{y}$ 会在解码器的作用下, 得到重构的向量 $\boldsymbol{z}$, 计算公式见式(3)。

$$\boldsymbol{y} = f_{\boldsymbol{\theta}}(\boldsymbol{x}) = s(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}), \tag{2}$$

$$z = g_{\boldsymbol{\theta}'}(\boldsymbol{y}) = s(\boldsymbol{W}'\boldsymbol{y} + \boldsymbol{b}')。 \tag{3}$$

编码参数是 $\boldsymbol{\theta}=\{\boldsymbol{W}, \boldsymbol{b}\}$, 解码参数是 $\boldsymbol{\theta}'=\{\boldsymbol{W}', \boldsymbol{b}'\}$。其中 $\boldsymbol{W}$ 是一个 $d' \times d$ 的权重矩阵, $\boldsymbol{W}'$是 $\boldsymbol{W}$ 的转置, 即 $\boldsymbol{W}'=\boldsymbol{W}^{\mathrm{T}}$, $\boldsymbol{b}$ 和 $\boldsymbol{b}'$是偏倚向量。

自动编码器的学习过程是无监督的, 优化的目标是使重构后的向量 $\boldsymbol{z}$ 尽量还原输入向量 $\boldsymbol{x}$, 即最小化重构带来的损失, 得到最优参数$\boldsymbol{\theta}^*$和$\boldsymbol{\theta}'^*$, 见式(4)。本文使用的损失函数为 Kullback-Leibler 散度, 如式(5)。

$$\boldsymbol{\theta}^*, \boldsymbol{\theta}'^* = \arg\min_{\boldsymbol{\theta}, \boldsymbol{\theta}'} L(\boldsymbol{x}, \boldsymbol{z}) = \arg\min_{\boldsymbol{\theta}, \boldsymbol{\theta}'} L(x, g_{\boldsymbol{\theta}'}(f_{\boldsymbol{\theta}}(\boldsymbol{x}))) \tag{4}$$

$$L(\boldsymbol{x}, \boldsymbol{z}) = KL(\boldsymbol{x} \| \boldsymbol{z})。 \tag{5}$$

自动编码器采用经典随机梯度下降算法进行训练, 在每个迭代过程中, 使用式(6)更新权重矩阵:

$$\boldsymbol{W} \leftarrow \boldsymbol{W} - l \times \frac{\partial L(\boldsymbol{x}, \boldsymbol{z})}{\partial \boldsymbol{W}}, \tag{6}$$

其中 l 为学习率, $\boldsymbol{b}$ 和 $\boldsymbol{b}'$采用相同的方式更新。自动编码器的结构如图 2 所示, 编码和解码的过程完成了文本信息的特征提取, 学习过程和误差控制保证了输出结果能反映输入文本的主要特征。

2.3 L1 范式正则化

自动编码器强大的非线性表达能力使得它会经常性地出现对输入数据的过度拟合, 即对个别对象特有的特征也进行充分描述。短文本的结构差异较大, 特有的特征较多。如果直接运用自动编码器算法, 将导致最终抽取的特征向量不能反映短文本的公有分布性特点, 训练出来的模型泛化能力比较差, 无法推广运用到其他短文本。因此本文对自动编码器的学习能力进行了约束。

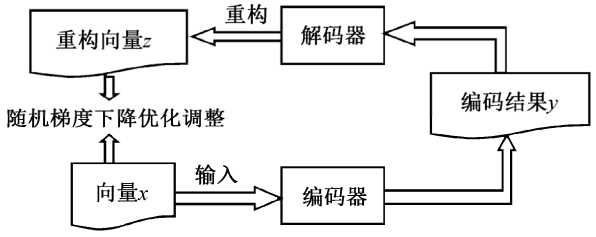

图 2 基本的自动编码器结构
Fig. 2 Structure for auto-encoder

L1 范式正则化是一种常用的变量选择方法, 被广泛运用于模型的改进。本文采用这种思想, 利用绝对值函数作为惩罚项来压缩自动编码器的系数, 使绝对值较小的系数自动压缩为 0, 从而保证算法中各项参数的稀疏性, 避免过分学习短文本中的非显著特性。具体地, 是将前面的式(5)调整为式(7)和(8)来计算。

$$L(\boldsymbol{x}, \boldsymbol{z}) = KL(\boldsymbol{x} \| \boldsymbol{z}) + \mathrm{Lasso}(\boldsymbol{\theta}), \tag{7}$$

$$\mathrm{Lasso}(\boldsymbol{\theta}) = \lambda \sum_{j=0}^{|\boldsymbol{\theta}|} |\boldsymbol{\theta}_j|。 \tag{8}$$

其中 λ 是 L1 范式的参数, 其值越大, 惩罚力度越大, 训练得到的结果会越稀疏, 其取值需要根据实际数据进行多次调试, 帮助模型达到拟合能力和泛化能力的均衡。

2.4 加噪处理

根据 Bingio 等[17]的研究可知, 自动编码器在输出层维度大于或等于输入层维度时, 可以得到比较好的特征提取效果。但由于短文本构成的输入向量十分稀疏, 在输出层维度较高的情况下, 自动编码器算法中的编码器极有可能不会进行任何非线性的变换学习, 而直接复制稀疏的输入向量, 将其输出到解码器中, 无法达到提取短文本中抽象特征的目的。

此外, 网络短文本的输入随意性很高, 大量的网民会在发布的文字中添加一些个性化的符号和语言, 或者由于输入太匆忙, 多输、漏输甚至错输一些文字, 给基于短文本的特征提取提出更高的要求, 训练出来的模型必须具有较强的鲁棒性。

针对这些问题, 本文采取的方法是先在短文本向量中添加一定噪音, 再将其输入到编码器中进行训练。与 Vincent 等[18]直接选取一定比例的数据强制变为 0 的方法不同, 本文一方面选取部分数据强制为 0, 另一方面也随机挑选一定比例的数据, 强制变为 1。前者是考虑到高维的输入向量中可能存在一些数据缺失, 训练出来的自动编码器应该能够还原这些缺失的特征; 后者是考虑到网络短文本输入的不规范性, 保证模型避免受到个性化或者无关输入的影响。加入噪音后, 输入向量 x 变成了 $\tilde{x}$, 随机梯度下降算法优化的计算如下所示:

$$\theta^*, \theta'^* = \arg\min_{\theta, \theta'} L(x, z) = \arg\min_{\theta, \theta'} L(x, g_{\theta'}(f_{\theta}(\tilde{x})))。 \tag{9}$$

将多个噪音稀疏自动编码器叠加起来就形成深度学习网络。在训练的过程中，K 层网络的输入是 K–1 层网络中编码器输出的短文本向量，K 层网络通过最小化损失函数，不断调整参数，使其输入与解码器重构后的结果尽量接近。达到最优解后，K 层网络丢弃解码器，将编码器输出的经过抽象后的低维特征向量作为 K+1 层的输入，继续进行下一层训练。如此循环，逐层训练，就构成基于深度噪音稀疏编码器的短文本特征提取模型，其结构如图 3 所示。

2.5 短文本聚类

经过上述自动编码器处理，并通过正则化和加噪过程就得到短文本低维特征向量，为进一步的文本挖掘打下基础。本文将以上计算得到的结果，应用于短文本的聚类分析，探讨其对聚类效果的影响。

K-means 算法作为一个简单高效的数据聚类算法，在文本聚类中应用广泛。本文主要采用 K-means 算法，从训练得到的低维短文本特征向量中随机选取 K 个短文本向量作为初始簇中心。根据与簇中心的距离，其他每个短文本向量都被分配到最近的一个簇，然后重新计算每个簇的均值，再使用这些新簇中心，重新分配每个短文本向量，直到短文本向量的分配不再发生变化。这样就得到短文本的最终聚类结果。

3 实验及结果分析

3.1 实验数据

网络短文本主要有微博、评论、即时消息、在线问答等形式。本文选取比较有代表性的微博数据作为分析对象，利用基于深层噪音稀疏自动编码器的短文本聚类算法，进行无监督学习，然后与人工标注的结果及已有的相似研究进行对比，从而验证该算法的有效性。数据来源于大数据共享平台——数据堂上的微博分类语料库。该数据集经过人工标注，分为 IT、财经、健康三大类。本文中每类各取 1500 条微博。

3.2 评价指标

本文采用 Entropy 和 Precision 两种衡量聚类效果的指标[19]。

Entropy 衡量一个聚类结果的纯度，其值越小，代表聚类的纯度越高。Entropy 的计算公式如式(10)和(11)所示。

$$\text{Entropy}=-\sum_{k=1}^{G'}\frac{|A_k|}{N}\sum_{j=1}^{G}p_{jk}\times\log(p_{jk}), \tag{10}$$

$$p_{jk}=\frac{1}{|A_k|}|\{d_i\,|\,\text{label}(d_i=c_j)\}|, \tag{11}$$

其中 G 表示通过算法得到的聚类个数；G'表示实际的类别个数，A_i 表示聚类中的某一个簇类，其中每条微博 $d_i\in A$，i=1, …, $|A|$的实际标注为 label(d_i)，其值等于标准的类标记 C_j(j=1, …, G)。

Precision 衡量聚类结果的准确度，其值越大，代表聚类的质量越高。它假设每个簇类中数量最多的实际类标识就是该簇类的标识，所以每个簇类的 Precision 等于最大的实际类标识所占的比例，如式(12):

$$\text{Precision}(A_i)=\frac{1}{|A|}\max(|\{d_i\,|\,\text{label}(d_i)=c_j\}|), \tag{12}$$

聚类结果总体的 Precision 是所有聚类 Precision 的加权平均:

$$\text{Precision}=\sum_{k=1}^{G'}\frac{|A_k|}{N}\text{Precision}(A_k)。 \tag{13}$$

3.3 实验过程

对微博分类语料库中的数据进行预处理：首先去除“$LOTOzf$”和“转发(330)评论(75)12 月 1 日 15:00 来自时光机”等与微博内容无关的内容；其次微博中的链接都是自动生成的短链接，对分析微博内容没有帮助，但机器用户的微博往往刻意添加更多的无关链接，因而去除短链接的字符串，只保留“http”关键词；再次，利用常见中文错别字大全，对

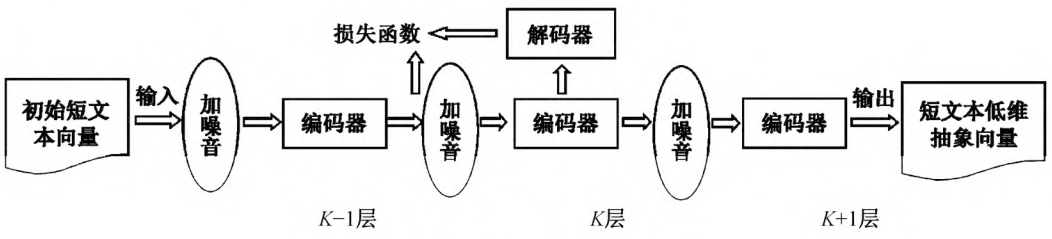

图 3 短文本深度学习结构图

Fig. 3 Structue for deep learning in short text

微博中输入错误的词语进行替换; 最后使用NLPIR2014 汉语(中文)分词系统对清理好的微博进行分词, 并结合微博内容的特点, 建立停止词词表, 去除分词结果中的停止词, 得到微博数据词袋。

实验共分成 4 部分: 对文本特征向量直接进行聚类; 利用维基百科词条内容扩展文本特征向量后再进行聚类; 在引入语义词典合并相似词的基础上构建特征向量后, 再进行聚类; 采用本文的方法(即经过深层噪音稀疏自动编码器处理)后再进行聚类。聚类方法均采用 K-Means 算法。

对于直接进行 K-Means 聚类的方法, 由于分词结果共 17171 个不同的词汇, 进行词频统计发现, 其中 6698 个词汇仅出现了一次, 对分析短文本之间的相似度意义不大。结合 Glorot 等[6−7]的研究结果, 前 5000 个词就能较全面地体现短文本的内容, 更多的词并不会对实验结果造成太大的影响, 因而本实验也选取前 5000 个词作为每条微博的特征集, 按照前文叙述的方法, 建立对应的空间向量, 然后利用 K-Means 算法直接进行聚类。

对于先利用维基百科词条内容扩展文本特征向量的方法, 从维基百科下载 2014 年 8 月 23 日发布的中文版维基百科库, 并导入 Mysql 数据库, 然后利用 Lucene 创建维基百科内容的索引。与 Banerjee 等[12]在短文本聚类中使用的英文不同, 中文的词与词之间没有空格作为自然分界符, 如果直接将整段短文本在维基百科中查询, 往往得不到任何查询结果。参考 Su 等[20]的研究成果, 将每条微博词袋中的每个词都作为关键词进行搜索, 得到用于扩充信息的维基百科文档。将该文档的分词结果与原微博词袋结合在一起, 利用信息增益值指标进行排序, 取前 5000 个词作为每条微博的特征集, 建立空间向量并利用 K-Means 进行聚类。

对于语义词典合并相似词的方法, 实验采用的方法与杨婉霞等[16]的方法相同, 首先下载哈尔滨工业大学社会计算与信息检索研究中心发布的《同义词词林扩展版》, 统计词频时, 将表达同一个概念的多个同义词进行词频合并处理, 每条微博抽取前 20 个词作为关键词, 构成该微博的特征向量, 最后进行 K-Means 聚类。

对于本文提出的深层噪音稀疏自动编码器方法, 采用基于 Python 的 Theano 库来实现。随着深度学习隐藏层数量的增加, 训练所花费的时间会出现较快的增长, 经过多次调试, 本实验选取拥有三层隐藏层的深度学习结构, 结点个数分别为 5000, 3000 和 1000。当置 0 和置 1 参数过大时, 自动编码器会由于信息缺失过多而出现非常高的误差。此外, 通过实验发现, 置 0 和置 1 逐层递减时能够取得较好的效果, 且置 1 参数不宜过大。最终本实验选择的置 0 噪音的添加概率分别是 0.3, 0.2 和 0.1, 置 1 噪音的添加概率分别是 0.03, 0.02 和 0.01。实验测试过的 L1 范式正则化的惩罚项系数有 10^{-1}, 10^{-2}, …, 10^{-8}, 其中 10^{-4} 表现最好。训练模型时, 随机梯度下降算法学习率的选取也十分重要, 学习率选取过大, 极易导致模型收敛于局部最优解, 过小又会使得训练的时间过长。本实验同样测试了 10^{-1}, 10^{-2}, …, 10^{-8} 等 8 个不同的参数, 最终发现学习率为 10^{-2} 结果最优。

3.4 实验结果和分析

上述 4 种方法得到的聚类结果如表 1~4 所示, 对应的信息熵和准确度如图 4 和 5 所示, 其中 K-means 代表没有经过处理的结果, Wiki+K-Means 代表经维基百科扩展后的结果, Cilin+K-Means 代表引入同义词词林后的结果, DSAE+K-means 代表经过深层噪音稀疏自动编码器处理过的结果。综合信息熵和准确度两种衡量标准可知, 4 种方法的效果由低到高排列, 依次为 K-means, Wiki+K-Means, Cilin+K-Means 和 DSAE+K-means。

从实验结果可以看出, 直接使用 K-Means 的效果最差, 综合信息熵为 0.457, 加权准确率也仅有 61.4%, 说明短文本高维、稀疏的特点对传统空间向量方法影响很大, 使其基本失去了实际应用的价值。Wiki+K-Means 的方法仅比 K-Means 略好一

表 1 K-means 聚类结果

Table 1 Clustering results by K-means

聚类结果	IT	财经	健康
簇类 1	777	301	286
簇类 2	364	1072	348
簇类 3	359	127	866

表 2 Wiki+K-means 聚类结果

Table 2 Clustering results by Wiki+ K-means

聚类结果	IT	财经	健康
簇类 1	926	233	237
簇类 2	384	1026	167
簇类 3	190	241	1096

表 3　Cilin+K-means 聚类结果

Table 3　Clustering results by Cilin+K-means

聚类结果	IT	财经	健康
簇类 1	1120	192	194
簇类 2	134	1278	207
簇类 3	246	30	1099

表 4　DSAE+K-means 聚类结果

Table 4　Clustering results by DSAE+K-means

聚类结果	IT	财经	健康
簇类 1	1317	64	103
簇类 2	23	1433	225
簇类 3	160	3	1172

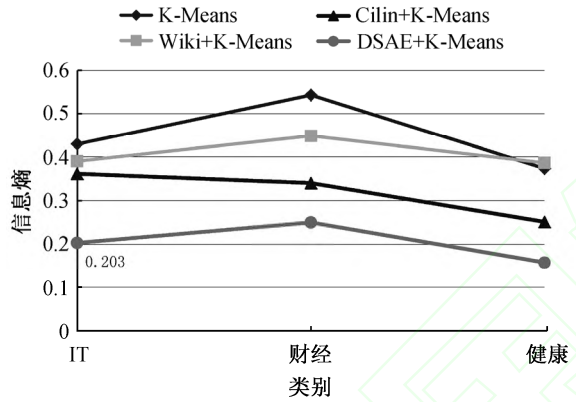

图 4　聚类的信息熵

Fig. 4　Entropy of the three clusters

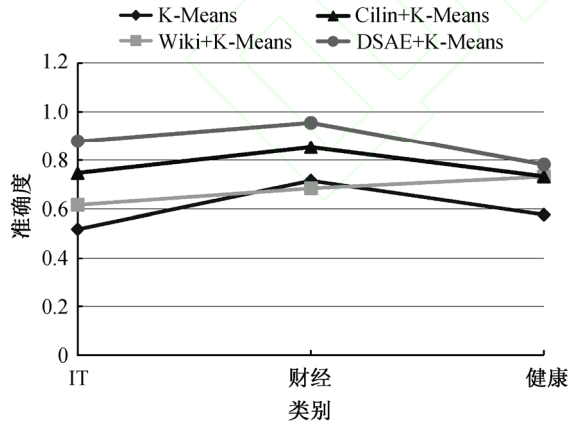

图 5　聚类的准确度

Fig. 5 Precision for the results of clustering

些，这主要是因为之前实验使用的短文本数据都是新闻类的，而使用词汇时，微博比新闻更为随意、多变。利用维基百科对微博中的词进行信息扩充时，部分词由于能够准确地找到相应的词条，比如健康类的词汇与其他词混淆的较少，信息得到了正确地扩充，准确率能够明显提升。但对于其他两类微博，使用维基百科扩展的方式反而可能会因为找到的部分词条信息与作者要反映的内容相差较大，无法提高聚类的效果。Cilin+K-Means 的聚类整体效果还不错，说明利用同义词进行词频合并以实现短文本向量降维的方法是较为有效的。但一方面此方法最终使用的特征向量维数较少，丢失了较多有价值的信息，使得聚类效果无法达到最好；另一方面它对同义词词典的要求比较高，未登录词会对该方法造成较大的影响。DSAE+K-Means 在几种方法中取得最好的成绩，综合信息熵为 0.207，综合准确度为 87.8%，聚类结果的纯度最高，准确度也最好，说明该深层噪音稀疏自动编码器能够利用自己非线性的特性，从高维底层特征学习得到低维抽象特征，挖掘出短文本空间向量中的本质，显著提升了聚类的性能。

4　结语

本文针对网络短文本空间向量高维、稀疏的特点，提出了深度噪音稀疏自动编码器算法，在基本的自动编码器模型中，增加避免过度拟合、控制稀疏性的 L1 范式正则化，又在输入数据中添加了置 0 和置 1 噪音，在降低短文本空间向量维度的同时，有效地提取出了数据中的本质特征，并在短文本的聚类分析中取得良好的聚类效果。

这种通过深度学习网络，利用大数据自身来学习特征的方式，比人工提取的形式更能有效地保证数据的本质特征，其结果对于许多自然语言处理任务，比如观点提取、文本分类等也能起到较好的扩展应用。此外，下一步的工作将在短文本聚类算法上进行语义分析，提高对特征提取和聚类结果的可解释性。

参考文献

[1] 中国互联网网络信息中心.第 33 次中国互联网发展状况调查统计报告[EB/OL].(2014－03－05)[2014－07－01]. http://www.cnnic.net.cn/hlwfzyj/hlwxzbg/hlwtjbg/201403/t20140305_46240.htm

[2] Hinto G, Salakhutdinov R. Reducing the dimensionality of data with neural networks. Science, 2006, 313: 504–507

[3] Bengio Y, Lamblin P, Popovici D, et al. Greedy layer-wise training of deep networks. Advances in Neural Information Processing Systems, 2007(19): 153–160

[4] Vincent P, Larochelle H, Lajoie I, et al. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. Journal of Machine Learning Research, 2010, 11(12): 3371–3408

[5] Bengio Y, Yao L, Alain G, et al. Generalized denoising auto-encoders as generative models. Advances in Neural Information Processing Systems, 2013(26): 899–907

[6] Glorot X, Bordes A, Bengio Y. Deep sparse rectifier networks // Proceedings of the 14th International Conference on Artificial Intelligence and Statistics. Fort Lauderdale, FL, 2011: 315–323

[7] Glorot X, Bordes A, Bengio Y. Domain adaptation for large-scale sentiment classification: a deep learning approach // Proceedings of the 28th International Conference on Machine Learning (ICML-11). Washington, 2011: 513–520

[8] Lu Shixiang, Chen Zhenbiao, Xu Bo. Learning new semi-supervised deep auto-encoder features for statistical machine translation // Proceeding of the 52nd Annual Meeting of the Association for Computational Linguistics. Maryland, 2014: 122–132

[9] Salakhutdinov R, Hinton G. Semantic hashing. International Journal of Approximate Reasoning, 2009, 50(7): 969–978

[10] 张开旭, 周昌乐. 基于自动编码器的中文词汇特征无监督学习. 中文信息学报, 2013, 27(5): 1–7

[11] Fan X, Yu H, Wang D. Utilizing the relations between entities to recognize organization name in Chinese short text. Journal of Computational Information Systems, 2010, 6(1): 121–129

[12] Banerjee S, Ramanathan K, Gupta A. Clustering short texts using Wikipedia // ACM SIGIR conference on Research and Development in Information Retrieval. Amsterdam, 2007: 787–788

[13] 邱云飞, 王琳颖, 邵良杉, 等. 基于微博短文本的用户兴趣建模方法. 计算机工程, 2014, 40(2): 279

[14] Jin O, Liu N N, Zhao K, et al. Transferring topical knowledge from auxiliary long texts for short text clustering // Proceedings of the 20th ACM International Conference on Information and Knowledge Management. Glasgow, UK, 2011: 775–784

[15] Tang J, Wang X, Gao H, et al. Enriching short text representation in microblog for clustering. Frontiers of Computer Science, 2012, 6(1): 88–101

[16] 杨婉霞, 孙理和, 黄永峰. 结合语义与统计的特征降维短文本聚类. 计算机工程, 2012, 38(11): 171–175

[17] Bengio Y, Lamblin P, Popovici D, et al. Greedy layer-wise training of deep networks // Advances in Neural Information Processing Systems 19 (NIPS'06). Cambridge: MIT Press, 2007: 153–160

[18] Vincent L H, Bengio Y, Manzagol P A. Extracting and composing robust features with denoising autoencoders // Proceedings of the Twenty-fifth ACM International Conference on Machine Learning. Helsinki, 2008: 1096–1103

[19] Tao L, Shengping L, Zheng C, et al. An evaluation on feature selection for text clustering // Proceedings of the Twentieth International Conference on Machine Learning. Washington, 2003: 488–495

[20] Su Chen, Yanne P, Zhang Y. Text classification using Wikipedia knowledge. ICIC Express Letters Part B: Applications, 2012, 3(5): 1251–1257