基于 Word2Vec 的一种文档向量表示

唐 明 朱 磊 邹显春
(西南大学计算机与信息科学学院 重庆 400715)

摘 要 在文本分类中,如何运用 word2vec 词向量高效地表达一篇文档一直是一个难点。目前,将 word2vec 模型与聚类算法结合形成的 doc2vec 模型能有效地表达文档信息。但是,这种方法很少考虑单个词对整篇文档的影响力。为了解决这个问题,利用 TF-IDF 算法计算每篇文档中词的权重,并结合 word2vec 词向量生成文档向量,最后将其应用于中文文档分类。在搜狗中文语料库上的实验验证了新方法的有效性。

关键词 TF-IDF,word2vec,doc2vec,文本分类
中图法分类号 TP181 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.6.043

Document Vector Representation Based on Word2Vec

TANG Ming ZHU Lei ZOU Xian-chun
(School of Computer and Information Science, Southwest University, Chongqing 400715, China)

Abstract In text classification issues, it is difficult to express a document efficiently by the word vector of word2vec. At present, doc2vec built on the combination of word2vec and clustering algorithm can express the information of document very well. However, this method rarely considers a single word's influence for the entire document. To solve this problem, in this paper, TF-IDF algorithm was used to calculate the right weight of words in documents, and word2vec was combined to generate document vectors, which were used for Chinese text classification. Experiments on the Sogou Chinese corpus laboratory demonstrate the efficiency of this newly proposed algorithm.

Keywords TF-IDF, Word2vec, Doc2vec, Text classification

1 引言

目前,使用最广泛的文档表示方法几乎都基于词袋法(Bag-of-Word,BOW)[1,2]。词袋法将文档看成是一些词的集合,在该集合中,每个词的出现是相互独立的,且不考虑词的顺序、语法和语义等信息。它将一篇文档表示成与训练词汇集合相同维度的向量,向量中每个位置的值即是该位置所代表的词在文档中出现的次数,并且随着新词汇的增加,文档向量维度也会增加。虽然词袋法在传统分类器上的分类效果不错,比如目前比较成熟的分类技术:回归模型、最近邻分类(KNN)、贝叶斯分类、决策树、RBF 神经网络、支持向量机(SVM)等[3-5],但它依旧存在几个主要问题:1)维度太高,文本向量的维数与训练数据集中出现的所有单词的数目一样多,这样容易出现所谓的"维度灾难"现象,而且如果某一个词汇在训练集中没有出现过,则该词汇在测试集中出现时就无法成为该文本的特征;2)一篇普通文档只有 1000 个词左右,而词向量的维度却能达到 10 万,利用率仅为 1%,所以基于 BOW 表示的文档向量非常稀疏,不利于一些自然语言处理任务;3)词袋法无法很好地表示一篇文档的语义,它假设词与词之间相互独立,并不考虑词与词之间的关系,如"土豆"与"马铃薯"这两个词在用词袋法所表示的文档向量计算相似度时的值为 0,但是我们知道"土豆"与"马铃薯"是同一种食物;4)词袋法很难区分同一个词在不同语境中的意义,如"先生",根据上下文,它可能是对男性的称呼,也可能是古代对老师的称呼,但在词袋法中,其文档向量计算相似度为 1。

随着深度学习的发展[10,11],基于神经网络的自特征抽取的词向量表示方法越来越受工业界和学术界的关注。基于前人的研究,Mikolov 等人[6]在 2013 年提出了 word2vec 模型[7]用于计算词向量(即下文的 Distributed Representation,后面均简称为词向量)。word2vec 模型利用词的上下文信息将一个词转化成一个低维实数向量,越相似的词在向量空间中越相近。将词向量应用于自然语言处理非常成功,已经被广泛应用于中文分词[12,13]、POS Tagging[14]、情感分类[10,11,15]、句法依存分析[10,16]等。

然而一篇文档由无数词构成,如何利用词向量有效地表示一篇文档是当前的一个难点。目前在这方面的研究进展缓慢,常见的方法有对一篇文档所包含的所有词向量求平均值[17]、对词向量聚类[18]以及 doc2vec 模型[19]。但这些方法并未重视单个词对整个文档的影响力。针对这个问题,本文在 word2vec 的基础上,利用 TF-IDF 算法[8]对每篇文档中的分词进行加权,并在搜狗中文实验语料库上进行测试,测试结果验证了该方法的有效性。

到稿日期:2016-01-19 **返修日期**:2016-04-20
唐 明(1974-),男,硕士,工程师,主要研究方向为数据挖掘,E-mail:tangming@swu.edu.cn;朱 磊(1992-),男,硕士生,主要研究方向为机器学习;邹显春(1965-),男,硕士,副教授,主要研究方向为数据挖掘、机器学习。

2 相关工作

2.1 词的向量化

词的向量化就是将语言中的词进行数学化，也即把一个词表示成一个向量。词的向量化主要有以下 3 种表达方式。

(1)one-hot representation 方式

这是一种最简单的方式，用一个很长的向量来表示一个词。向量的长度为词典的大小(通常达到 10^5)，向量的分量只有一个 1，其余全为 0，1 的位置对应该词在词典中的位置。比如，“土豆”表示为[0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 …]，而“马铃薯”表示为[0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 …]。这种方式虽然可以简单明了地表达一个词语，但是却无法有效表达它们的语义信息。“土豆”和“马铃薯”虽然是同一种食物，但利用常规的向量距离公式，比如欧几里德距离或者余弦距离公式，都无法有效计算它们的相似度，显然这种方式不能很好地表达词之间的相似性。

(2)Distributed representation(词向量)

这种方式能很好地克服 one-hot representation 方式的缺点，最早由 Hinton[9] 提出，它是将词映射到一个低维、稠密的实数向量空间中(空间大小一般为 100 或者 200)，使得词义越相近的词在空间的距离越近。上面的例子可以类似地表达如下，“土豆”可以表示为：[0.843 −0.125 0.734 −0.345 0.654 …]，而“马铃薯”为[0.923 −0.231 0.698 −0.233 0.743 …]，显然，这种表示方式有利于使用距离向量公式比较词向量之间的相似度。

(3)word2vec 模型训练词向量

通过借鉴 Bengio 提出的 NNLM(Neural Network Language Model)[27] 以及 Hinton 的 Log_Linear 模型[28]，Mikolov 等提出了 word2vec 语言模型[19]。word2vec 可以快速有效地训练词向量。

word2vec 模型有两种，分别是 CBOW 模型(见图 1)以及 Skip-gram 模型(见图 2)。其中 CBOW 模型利用词 $w(t)$ 前后各 c(这里 $c=2$)个词去预测当前词；而 Skip-gram 模型恰好相反，它利用词 $w(t)$ 去预测它前后各 c($c=2$)个词。

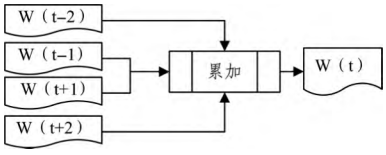

图 1 CBOW 模型

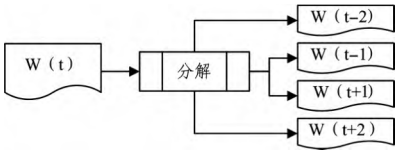

图 2 Skip-gram 模型

由于 CBOW 模型的训练和 Skip-gram 模型的训练类似，这里仅介绍 CBOW 模型的训练过程。其中输入层是词 $w(t)$ 的上下文中的 $2c$ 个词向量，而投影层向量 X_w 是这 $2c$ 个词向量的累加和。输出层是以训练语料库中出现过的词作叶子节点，以各词在语料库中出现的次数作为权值构造出的一棵 Huffman 树。在这棵 Huffman 树中，叶子节点共 $N(=|D|)$ 个，分别对应词典 D 中的词，非叶子节点 $N-1$ 个。通过随机梯度上升算法对 X_w 的结果进行预测，使得 $p(w|context(w))$ 值最大化，$context(w)$ 指词的上下文中的 $2c$ 个词。

当神经网络训练完成时，即可求出所有词的词向量 w。有趣的是，当利用词向量表示一个词时，可以发现类似这样的规律：“king”−“man”+“woman”=“queen”[22]，可以看出词向量非常有利于表达词的语义特征。

2.2 文档的向量化

(1)BOW 模型

文档的向量化就是将一篇文档表示为一个向量，主要是基于词的向量化。将文档向量化之后，就可以利用常规的距离向量公式比较两篇文档之间的相似度。传统的 BOW 可以看作是词的 one-hot 表示向量的叠加，比如 2.1 节中“土豆”的词向量为[0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 …]，“马铃薯”的词向量为[0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 …]，而一篇仅包含“土豆”和“马铃薯”这两个词的文本就可以表示为[0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 …]。显然这种表达方式存在的问题与 one-hot 一样，由于其特征向量的高维性和稀疏性，很难利用常规的向量距离公式有效地计算两篇文档之间的相似度。当然传统的 BOW 也有许多优化的方法，利用 TF-IDF 加权是其中一种，即将文本向量中出现非“0”的值替换为 TF-IDF 权值，这样的特征向量比传统 BOW 在文本分类方面更有效。

(2)doc2vec 模型

doc2vec 模型[18,19] 的训练与 word2vec 模型类似，在利用词的上下文对当前词进行预测的训练过程中添加了一个文档特征向量，其预测模型和训练过程分别如图 3 和图 4 所示。

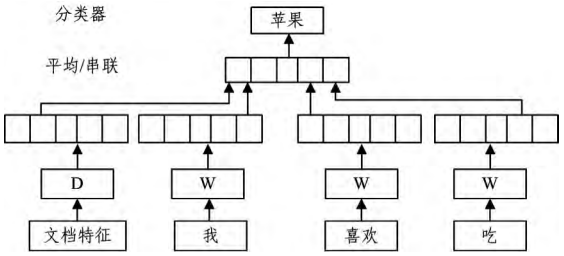

图 3 doc2vec 预测模型

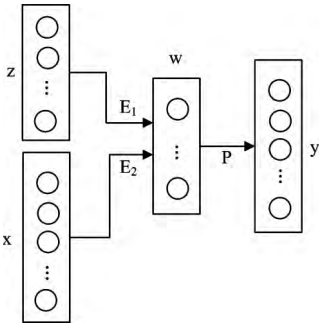

图 4 doc2vec 的训练过程

从图 3 可以看出，该模型是利用前几个词 W 去预测当前的词，比如用“我”、“喜欢”、“吃”、“苹果”这个句子中的前 3 个词去预测第 4 个词，而且在训练过程中添加了一个文档特征

向量 D，其余部分与 word2vec 预测模型类似，可以将该文档特征向量看作是一个表示当前文档中的其余部分信息或者主题信息的向量。通常根据文档特征向量的维度不同，将该文档的特征向量与词向量相结合的方法包括取平均值和串联两种方式。图 4 中的训练过程是利用串联的方式对文档向量进行训练。其中 z 代表文档向量，x 代表词向量，w 代表隐含层权值，y 代表输出层的值。与 word2vec 类似，y 中的每一维即代表词典中的一个词，E_1、E_2 和 P 分别代表它们之间的连接矩阵。其训练过程与 word2vec 模型类似，并利用 BP 算法调节参数。

(3)基于 word2vec 的方法

目前基于 word2vec 的文档表示方法主要是一些聚类的方法或者取平均值的方法，而表现较优的是 doc2vec 模型，因为它在训练的时候加入了表达文档的特征向量，所以用于文档分类等任务时，一般表现比 word2vec 好。

综合考虑词的向量化方式和文档的向量化方法，结合 word2vec 和 TF-IDF 算法，提出了一种基于 word2vec 的文档向量表示方法。

3 基于 TF-IDF 算法的 word2vec 改进方法

在 word2vec 词向量的基础上，结合 TF-IDF 算法提出了文档向量的表示方法。

3.1 word2vec 与 TF-IDF 结合

对于包含 M 个文档的集合 D，其中 $D_i(i=1,2,\cdots,M)$ 已经采用分词工具 ANSJ 对中文文档进行分词，将其通过 word2vec 模型训练，得到每个分词对应的 N 维词向量 w，其中 $w=(v_1,v_2,\cdots,v_N)$。

对于每类文档集中的每个文档里的每个分词，利用 TF-IDF 算法计算其在该文档中的权重值 $K(t,D_i)$，其表示为词 t 在文档 $D_i(i=1,2,\cdots,M)$ 中的权重。TF-IDF 综合考虑了词在单个文档中出现的概率 tf 以及该词在整个文档集中的权重 idf。

词 t 的 idf 计算公式如下：

$$idf(t)=\log(M/n_t+0.01) \tag{1}$$

其中，M 为训练文档的总数；n_t 为训练文档集中出现词 t 的文档数。

TF-IDF 的计算公式如下：

$$K(t,D_i)=\frac{tf(t,D_i)\times idf(t)}{\sqrt{\sum_{t\in D_i}[tf(t,D_i)\times idf(t)]^2}} \tag{2}$$

其中，$tf(t,D_i)$ 为词 t 在第 i 篇文档中的词频，分母为归一化因子。

对于每篇文档 $D_i(i=1,2,\cdots,M)$，其文档向量可以表示为如下形式：

$$d_i=\sum_{t\in D_i} w_t K(t,D_i) \tag{3}$$

其中，w_t 表示分词 t 的词向量，所以文档向量 d 也是一个 N 维的实数向量。

3.2 文本分类的工作流程

对未知文档的分类过程如图 5 所示。由于对于单篇未知的文档，单独计算其 idf 值并无意义，因此在计算其 TF-IDF 值时，idf 值(如果是新词，则 idf 取 0.01)依然选取该分词在训练阶段的值。

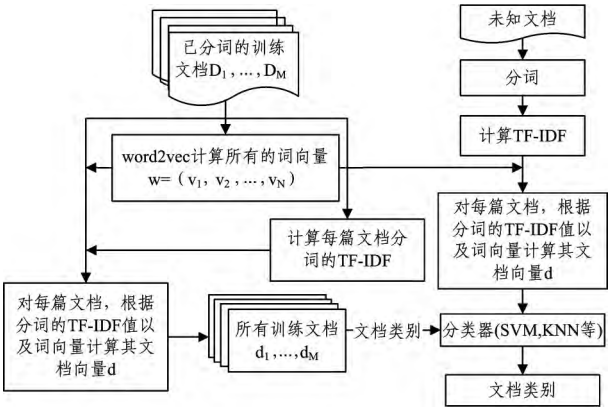

图 5 文本分类的工作流程

4 实验

4.1 文本分类模型评估

采用的评估指标包括准确率、召回率、F_1 指标以及宏平均。其中准确率 p 是指文本分类正确的样本数与所有分类文本数的比例：

$$p=\frac{a}{a+b}(\text{如果 } a+b=0,\text{则 } p=1) \tag{4}$$

召回率 r 是文本分类正确的样本数与该类的实际文本数的比例：

$$r=\frac{a}{a+c}(\text{如果 } a+c=0,\text{则 } r=1) \tag{5}$$

其中，a 是分类正确的样本数据，b 是错误地划分到该类别的样本数，c 是属于该类但未被区分出来的样本数。

F_1 指标是将准确率与召回率同时考虑的一种指标：

$$F_1=\frac{2pr}{p+r} \tag{6}$$

准确率、召回率以及 F_1 是对单独类别的分类性能进行评估；宏平均是对所有类别的 p，r 以及 F_1 的平均值，用来评估系统的总体分类性能。

4.2 实验结果与实验分析

实验选用的测试数据是搜狗实验室的中文文本分类语料库[25]，共包含文本 17910 篇，分为 9 类，分别是财经、IT、健康、体育、旅游、教育、招聘、文化和军事，其中每类有 1990 篇文本。在预处理过程采用 ANSJ 分词工具对文本进行分词，去掉标点符号、停用词、助词等后得到词条 247263 个。

将提出的 TF-IDF 加权的 word2vec 模型与 TF-IDF 加权的 BOW 模型(即将 BOW 向量中分词相应位置的值替换成 TF-IDF 权值)、均值 word2vec 模型以及 doc2vec 模型的分类效果进行对比，其中分类器分别采用了 KNN($K=30$)、libSVM 以及 RBF 神经网络。TF-IDF 加权的 BOW 模型在构建词向量时，特征提取采用了文档频率选择(Document Frequency，DF)与信息增益选择(Information Gain，IG)结合[23,24]的方法，且将词条在整个所属类别文本中出现次数为 5 以下的过滤掉，最终只选取了 67756 个词条来构建词向量。均值 word2vec 模型是对一篇文本计算其所有 word2vec 所得词向量的均值。其中，word2vec 以及 doc2vec 的计算采用 gensim 开源软件实现[26]。所有实验采用五分交叉验证，即把数据集随机划分成 5 份，每次取其中 4 份进行训练，1 份进行测试，然后把 5 次分类结果的平均值作为最终结果。测试结果用正

确率(p)、召回率(r)、F_1 指标进行评测，测试结果如表 1－表 4 所列。其中类别 $C1$、$C2$、$C3$、$C4$、$C5$、$C5$、$C7$、$C8$、$C9$ 分别代表财经类、IT 类、健康类、体育类、旅游类、教育类、招聘类、文化类以及军事类；avg 代表 $C1$－$C9$ 的宏平均值。

表 1　TF-IDF 加权 BOW 模型(%)

类别	SVM			KNN			RBF		
	p	r	F_1	p	r	F_1	p	r	F_1
C1	85.12	72.58	78.35	82.38	70.75	76.12	73.21	80.11	76.50
C2	90.78	78.39	84.13	92.60	75.38	83.11	70.93	69.27	70.09
C3	81.64	68.61	74.56	80.62	76.08	78.28	84.50	60.76	70.69
C4	95.47	71.29	81.63	97.71	61.46	75.46	96.67	79.28	87.11
C5	66.98	66.27	66.62	88.89	56.08	68.77	91.35	63.13	74.66
C6	67.01	73.82	70.25	45.01	83.42	58.47	87.59	43.21	57.87
C7	72.39	70.81	71.59	61.87	65.63	63.69	51.33	88.43	64.96
C8	77.14	75.32	76.22	57.03	71.41	63.42	53.22	75.39	62.39
C9	84.95	80.93	82.89	89.65	79.45	84.24	84.39	82.76	83.57
avg	80.16	73.11	76.25	77.31	71.07	72.40	77.02	71.37	71.98

表 2　均值 word2vec 模型(%)

类别	SVM			KNN			RBF		
	p	r	F_1	p	r	F_1	p	r	F_1
C1	87.67	83.85	85.71	90.98	85.38	88.1	76.59	82.61	79.45
C2	82.6	81.54	82.06	86.98	80.51	83.62	73.08	74.37	73.71
C3	84.05	79.74	81.84	82.54	88.46	85.4	87.11	54.67	67.08
C4	98.71	98.46	98.59	99.2	95.9	97.52	99.50	81.11	89.34
C5	86.63	83.08	84.82	85.22	82.82	84.01	90.49	61.11	72.88
C6	87.86	77.95	82.61	90.20	82.56	86.21	92.45	46.13	61.49
C7	80.05	86.41	83.11	81.86	83.33	82.59	49.07	91.46	63.87
C8	65.05	72.05	68.37	68.03	80.77	73.86	58.00	82.81	68.20
C9	86.43	93.08	89.63	92.86	93.33	93.09	88.92	87.84	88.34
avg	84.34	84.01	84.08	86.43	85.9	86.04	79.47	73.57	73.82

表 3　doc2vec 模型(%)

类别	SVM			KNN			RBF		
	p	r	F_1	p	r	F_1	p	r	F_1
C1	85.21	86.37	85.79	88.95	87.43	88.18	81.56	84.37	82.94
C2	86.59	82.35	84.42	87.56	82.48	84.94	91.02	67.38	77.44
C3	87.25	85.33	86.28	83.89	85.96	84.91	90.12	65.39	75.79
C4	96.53	97.80	97.16	97.85	98.21	98.03	96.52	96.34	96.43
C5	88.47	89.68	89.07	85.32	91.25	88.19	81.25	85.36	83.25
C6	89.51	81.28	85.2	85.98	80.23	83.01	90.24	58.56	71.03
C7	81.98	85.63	83.77	81.95	84.89	83.39	42.36	92.65	58.14
C8	82.15	81.33	81.73	79.56	77.85	78.70	85.89	52.33	65.04
C9	87.69	92.87	90.21	91.59	92.87	92.23	91.24	87.41	89.28
avg	87.26	86.96	87.06	86.96	86.80	86.84	83.36	76.64	77.70

表 4　TF-IDF 加权 word2vec(%)

类别	SVM			KNN			RBF		
	p	r	F_1	p	r	F_1	p	r	F_1
C1	86.64	87.64	87.07	89.56	87.41	88.45	80.17	85.85	82.89
C2	86.45	82.16	84.21	86.22	81.71	83.87	90.22	65.73	75.99
C3	86.80	84.40	85.55	83.86	86.71	85.22	89.33	67.01	76.53
C4	98.25	96.83	97.53	98.16	97.64	97.90	96.37	96.36	96.36
C5	87.93	89.20	88.54	84.33	92.09	88.02	80.60	84.10	82.28
C6	88.66	79.95	84.07	87.31	79.55	83.21	91.15	58.52	71.24
C7	80.54	86.96	83.57	82.21	85.78	83.93	43.30	94.87	59.45
C8	79.10	79.87	79.44	79.37	78.24	78.77	86.71	51.71	64.71
C9	87.86	93.64	90.61	92.08	93.12	92.58	90.85	88.37	89.57
avg	86.91	86.74	86.73	87.01	86.92	86.89	83.19	76.95	77.67

由表 1－表 4 可以发现，均值 word2vec 模型在 SVM、KNN 以及 RBF 分类器上的宏平均准确率、召回率以及 F_1 值比 TF-IDF 加权 BOW 模型的有不少的提升，比如宏平均正确率在 KNN 分类器上由 77.31%提升到了 86.43%，在另外 2 个分类器上也分别提升了 4.18%(SVM)和 2.45%(RBF)。宏平均召回率在 SVM、KNN 以及 RBF 分类器上分别提升了 10.9%，14.83%以及 2.2%。而宏平均 F_1 分别提升了7.83%，13.64%以及 1.84%。可以看出，均值 word2vec 模型所生成的词向量比传统的 BOW 所生成的词向量能更有效地表示一篇文档的特征。而本文提出的基于 TF-IDF 加权的 word2vec 模型相比均值 word2vec 模型又有一些提升，在 SVM、KNN 以及 RBF 分类器上，宏平均正确率分别提升了2.57%，0.58%以及 3.72%；宏平均召回率分别提升了2.73%，1.02%以及 3.38%；宏平均 F_1 分别提升了 2.65%，0.85%以及 3.85%。而 TF-IDF 加权的 word2vec 模型与 doc2vec 模型的宏平均值效果相差不大，在几组分类器上的效果相当，可见提出的方法的有效性，其可以作为另外一种用于文档分类的有效方法。由图 6 也可以清楚地看出，无论采用何种分类器，基于 TF-IDF 加权的 word2vec 模型在宏平均上均有不错的表现，验证了所提出的生成文档向量的方法在文档分类方面的有效性。

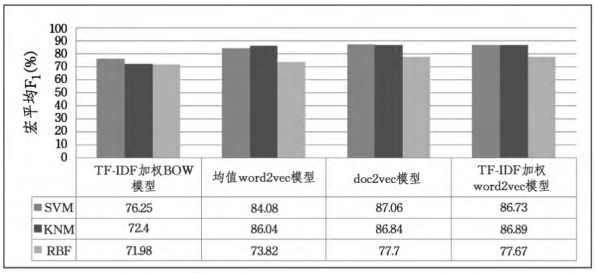

图 6　4 种文档向量分类效果比较

结束语　针对当前文本向量表示方法的不足，借助 word2vec 的优点，将 word2vec 和 TF-IDF 结合，提出了一种基于 word2vec 的 TF-IDF 加权计算文档向量算法。在搜狗中文实验语料库上的实验表明，相较于 TF-IDF 加权的 BOW 模型以及均值 word2vec 模型，本算法有更好的文本分类效果。

参 考 文 献

[1] Baeza-Yates R, Ribeiro-Neto B. Modern Information Retrieval [M]. New York: ACM press, 1999

[2] Manning C D, Schütze H. Foundations of Statistical Natural Language Processing [M]. Cambridge: MIT press, 1999

[3] Hwang M, Choi C, Youn B, et al. Word Sense Disambiguation Based on Relation Structure[C]// International Conference on Advanced Language Processing and Web Information Technology. 2008: 15-20

[4] Wang X, Mccallum A, Wei X. Topical N-Grams: Phrase and Topic Discovery, with an Application to Information Retrieval [C]// IEEE International Conference on Data Mining. IEEE Computer Society, 2007: 697-702

[5] Haruechaiyasak C, Jitkrittum W, Sangkeettrakarn C, et al. Implementing News Article Category Browsing Based on Text Categorization Technique [C]// 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology. IEEE Computer Society, 2008: 143-146

[6] Mikolov T, Sutskever I, Chen K, et al. Distributed Representations of Words and Phrases and their Compositionality [J]. Advances in Neural Information Processing Systems, 2013, 26: 3111-3119

(下转第 269 页)

[10] Gullo F, Ponti G, Tagarelli A. Clustering uncertain data via k-medoids[M]// Scalable Uncertainty Management. Springer Berlin Heidelberg, 2008: 229-242

[11] Xie Xiao-lu, Li Lei. Research on Multi-attribute Group Decision Under Interval Number Information[J]. Computer Engineering, 2014, 40(10): 210-213(in Chinese)
谢小璐, 李磊. 区间数信息下的多属性群决策研究[J]. 计算机工程, 2014, 40(10): 210-213

[12] Reynolds P A, Richards G J, Rayward-smith V. The Application of K-Medoids and PAM to the Clustering of Rules[J]. Lecture Notes in Computer Science, 2004, 3177: 173-178

[13] Aggarwal C C. Yu P S. A survey of uncertain dataalgorithms and applieations[J]. IEEE Transactions Onknowledge and Data Engineering, 2009, 21(5): 609-623

[14] Lu Zhi-mao, Feng Jin-gong, Fan Dong-mei, et al. New clustering algorithms for large data processing[J]. System Engineering and Electronics, 2014(5): 1010-1015(in Chinese)
卢志茂, 冯进玫, 范冬梅, 等. 面向大数据处理的划分聚类新方法[J]. 系统工程与电子技术, 2014(5): 1010-1015

[15] Zhou Shi-bing, Xu Zhen-yuan, Tang Xu-qing. New method for determining optimal number of clusters in K-means clustering algorithm[J]. Computer Engineering and Applications, 2010, 46(16): 27-31(in Chinese)
周世兵, 徐振源, 唐旭清. 新的 K-均值算法最佳聚类数确定方法[J]. 计算机工程与应用, 2010, 46(16): 27-31

[16] Yu Jian, Cheng Qian-sheng. Search range of the Optimal clustering number in fuzzy clustering algorithms[J]. Science in China: Series E, 2002, 32(2): 274-280(in Chinese)
于剑, 程乾生. 模糊聚类方法中的最佳聚类数的搜索范围[J]. 中国科学: E 辑, 2002, 32(2): 274-280

[17] Dudoit S, Fridlyand J. A prediction-based resampling method for estimating the number of clusters in a dataset[J]. Genome Biology, 2002, 3(7): 1-21

[18] Kao B, Lee S, Lee F, et al. Clustering Uncertain Data Using Voronoi Diagrams and R-Tree Index. [J]. Knowledge & Data Engineering IEEE Transactions on, 2010, 22(9): 1219-1233

[19] Eredm A, İmre GüNDEM T. M-FDBSCAN: A multicore density-based uncertain data clustering algorithm[J]. Turkish Journal of Electrical Engineering & Computer Sciences, 2014, 22(1): 143-154

(**上接第 217 页**)

[7] Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space [C]// ICLR 2013. 2013

[8] Joachims T. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization [M]. Springer US, 1997: 143-151

[9] Hinton G E. Learning distributed representations of concepts [C]// Proceedings of CogSci. 1986: 1-12

[10] Socher R, Bauer J, Manning C D, et al. Parsing with Compositional Vector Grammars [C]// Meeting of the Association for Computational Linguistics. 2013: 455-465

[11] Socher R, Perelygin A, Wu J Y, et al. Recursive Deep Models for Semantic Compositionality over a Sentiment Treebank [C]// Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP). 2013: 1631-1642

[12] Sun Y, Lin L, Yang N, et al. Radical-Enhanced Chinese Character Embedding [J]. Lecture Notes in Computer Science, 2014, 8835: 279-286

[13] Mansur M, Pei W, Chang B. Feature-based Neural Language Model and Chinese Word Segmentation [C]// IJCNLP. 2013: 1271-1277

[14] Zheng X, Chen H, Xu T. Deep Learning for Chinese Word Segmentation and POS Tagging [C]// EMNLP. 2013: 647-657

[15] Tang D, Wei F, Yang N, et al. Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification [C]// ACL. 2014: 1555-1565

[16] Zhang M, Zhang Y, Che W, et al. Chinese Parsing Exploiting Characters [C]// ACL. 2013: 125-134

[17] Xing C, Wang D, Zhang X, et al. Document Classification with Distributions of Word Vectors [C]// 2014 Annual Summit and Conference Asia-Pacific Signal and Information Processing Association (APSIPA). IEEE, 2014: 1-5

[18] Kim H K, Kim H, Cho S. Bag-of-Concepts: Comprehending Document Representation through Clustering Words in Distributed Representation[OL]. http://dm. snu. ac. kr/static/docs/TR/SNUDM-TR-2015-05. pdf

[19] Le Q V, Mikolov T. Distributed Representations of Sentences and Documents [J]. Eprint Arxiv, 2014, 4: 1188-1196

[20] Morin F, Bengio Y. Hierarchical Probabilistic Neural Network Language Model [J]. Aistats. 2005, 5: 246-252

[21] Mnih A, Hinton G E. A Scalable Hierarchical Distributed Language Model [C]// Advances in Neural Information Processing Systems. 2009: 1081-1088

[22] Mikolov T, Yih W, Zweig G. Linguistic Regularities in Continuous Space Word Representations [C]// HLT-NAACL. 2013: 746-751

[23] Santana L E A, De Oliveira D F, Canuto A M P, et al. A Comparative Analysis of Feature Selection Mmethods for Ensembles with Different Combination Methods [C]// International Joint Conference on Neural Networks, 2007 (IJCNN 2007). IEEE, 2007: 643-648

[24] Forman G. An Extensive Empirical Study of Feature Selection Metrics for Text Classification [J]. The Journal of Machine Learning Research, 2003, 3: 1289-1305

[25] **搜狗. 文本分类语料库**[OL]. http://www. sogou. com/labs/dl/c. html

[26] Gensim. Topic Modelling for Humans [OL]. http://radimrehurek. com/gensim

[27] Bengio Y, Schwenk H, Senécal J S, et al. Neural Probabilistic Language Models [M]// Innovations in Machine Learning. Springer Berlin Heidelberg, 2006

[28] Mnih A, Hinton G. Three New Graphical Models for Statistical Language Modelling [C]// Proceedings of the 24th International Conference on Machine Learning. ACM, 2007: 641-648