文章编号:1671-4598(2018)04-0217-04 DOI:10.16526/j.cnki.11—4762/tp.2018.04.057 中图分类号: 文献标识码:A

改进的 HMM 模型在特征抽取上的应用

陈昌浩，范太华
（西南科技大学 计算机科学与技术学院，四川 绵阳 621010）

摘要：目前，情感分类常用的特征抽取方法是基于词典的向量空间模型（VSM），潜在的语义分析（LSA）和基于无监督算法的词嵌入（word2vec），随机词向量法，这些方法都是对单个词语进行处理；通过哈工大词云对采集的豆瓣评论数据集进行语义角色进行的标记以后，采用了改进的隐马尔科夫模型（MHMM）对词对向量进行特征构建，并将其作为一个序列片段作为长短记忆门（LSTM）的输入，最后使用 softmax 函数对动态循环神经网络输出的序列进行分类；实验使用了交叉熵作为优化函数，采用了随机梯度下降法对优化函数进行迭代产生最优解；实验结果证明了该方法对豆瓣影评数据进行情感分类产生了更好的效果。

关键词：向量空间模型；词嵌入；改进的隐马尔科夫模型；情感分析

MHMM Used in Feature Extracting

Chen Changhao, Fan Taihua
(Southwest University of Science and Technology Computer Science and Technology, Mianyang 621010, China)

Abstract: Nowadays, In the process of emotion classification, the method of feature extracting is vector space model (VSM) which is based on dictionary, latent semantic analysis (LSA) and unsupervised algorithm such as word2vec or random words'object are a single word. The collected Douban datasets is processed with semantic role with LTP, and then extract these words pairs as feature by modified Hider Markov Model (MHMM), and next input these features into Long Short-Time Memory (LSTM), and finally use softmax function to recognize the emotional class. In the experiment, cross entropy is used, and stochastic gradient descent method is used for optimization parameters. It turn out that this way has a more effect on emotional classification when Douban datasets is used.

Keywords: VSM; Word2vec; MHMM; sentiment analysis

0 引言

传统的情感分析方式是基于简单统计的情感倾向分类，Tsou 等[1]利用大众对名人的评价语料，全面地统计分析极性元素分布密度和语义强度得到词语的语义倾向。

接着，基于机器学习的文本倾向性研究开始兴起，Pang 等[2-3]利用 bag-of-words 技术并且朴素贝叶斯、最大熵、支持向量机（SVM）分类器方法对电影影评进行情感倾向分析；Whitelaw 等[4]提取文本中形容词和修饰语词组作为特征结合词袋技术形成向量空间模型并采用 SVM 对电影影评分类；Turney 使用一些固定句法模式来抽取基于词性标注的标签。Taboada[5]提出基于词库的方法，用带有一定倾向和强度的情感词及词组的词典采用集约化方法计算每个文本的情感分值。向量空间模型的假设是特征与特征之间是相互独立的（正交假设），这在实际中难以满足。

为了改善向量空间模型的缺陷，LSA（Latent Semantic Analysis）[6]潜在语义分析的方法被提出了，并且在信息检索方面取得了一定的成功。

随着，计算机计算的存储能力和计算性能不断地提高，深度学习的方法再次进入人们的视野，并成为情感分类研究的热点。RNN 具有很强大的抽取文本信息的能力，并且循环神经网络（RNN）在 NLP 里应用广泛，论文[7]证明了 RNN 在文本分类和情感分类上效果很好，但是，RNN 解决不了长期依赖的问题，LSTM 模型能解决任何长度的序列，并且能够捕获长时间的独立性。

随着对 word2vec 的深入，以及谷歌对 word2vec 开源以后，在论文[8]中作者运用的是基于 word2vec 加权的 svm 算法，作者通过计算每一个文档当中词语的 tf-idf 作为权值，最后得出一个比较好的结果。在文章[9]中，作者将连续的三个词作为一个嵌入向量对进行输入，通过神经网络模型，最后测评分类结果，在这篇文章中，所有的词汇在空间上相邻，作为一个输入，这样的做法是 VSM 的扩展，相当于把词对作为一个单元进行处理，忽略了词对之间的关联性。因此，对词对的关联性如果按照论文的处理方式是空间位置的相邻。

1 特征抽取模型的设计

1.1 概率图模型

设 $\boldsymbol{X}=\{x_1, \cdots, x_k, \cdots, x_l\}$ 表示的是训练集，$x_k=(x_{k_1}), \cdots, x_{k_i}), \cdots, x_{k_p})$，其中 $k\in\{1, 2, \cdots, l\}$，$i\in\{1, 2, \cdots, p\}$，x_k 表示的是第 k 个样本，x_{k_i} 表示的是第 k 个样本中的第 i 个观测值，l 表示的是训练集的样本容量，p 表示的是是组成样本 x_k 的序列长度。

集合 $S=\{s_1, s_2, \cdots, s_n\}$ 为训练集当中所有的状态集合，集合 $O=\{o_1, o_2, \cdots, o_m\}$ 为观测值的集合（其中 m 为观测集合的长度，n 为状态值的长度）。那么隐马尔科夫模型所涉及概率图模型就如图 1。

图 1 表示的是训练集中任意一个序列对应的状态和观测值之间的关系。从上面的概率图模型中我们可以看出三个基本的变量，状态集，观测矩阵，状态转移矩阵。因此对于隐马尔科

收稿日期:2017-12-06； **修回日期**:2017-12-28。
基金项目:四川省教育厅资助项目(14ZB0113)。
作者简介:陈昌浩(1991-),男,四川遂宁市人,硕士研究生,主要从事自然语言处理方向的研究。
范太华(1962-),男,四川成都市人,副教授,研究生导师,主要从事知识工程方向的研究。

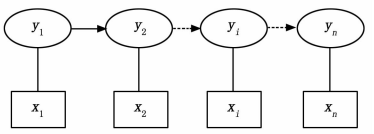

图 1 概率图模型

夫模型，我们定义三个最基本的矩阵和向量。状态转移矩阵 $\boldsymbol{A}=[a_{ij}]$（其中 i，$j\in\{1,2,\cdots,n\}$），$\boldsymbol{B}=[b_{ij}]$，其中 $i\in\{1,2,\cdots,n\}$，$j\in\{1,2,\cdots,m\}$。设 $\boldsymbol{\pi}=[\pi_1,\pi_2,\cdots,\pi_n]^T$ 表示状态向量，用来表示每个状态的权重。

其中对于以上的 $a_{ij}=P\{S=s_j\mid S=s_i\}$ 表示从 s_i 状态转移到 s_j 状态的一步转移概率，A 表示的是一步转移概率矩阵，$b_{ij}=P\{S=s_i\mid O=o_j\}$ 表示观测值对应的某一个状态的概率值。$\pi_i=P\{S=s_i\}$ 表示某一个状态对应状态概率。

而对于训练集要获得上面三个参数 $\Phi=(\boldsymbol{A},\boldsymbol{B},\boldsymbol{\pi})$。针对自然语言的特殊要求，在构建模型之前，我们定义一种运算如公式（1）：

$$\boldsymbol{\alpha}\otimes\boldsymbol{\beta}=\boldsymbol{\lambda} \tag{1}$$

在公式（1）中 $\boldsymbol{\alpha}$、$\boldsymbol{\beta}$、$\boldsymbol{\lambda}$ 都是 n 维向量，对于它们的任意分量都有 $\lambda_i=\alpha_i\times\beta_i$。

计算某一个观测值对应的状态表示值用公式（2）进行计算：

$$\boldsymbol{\gamma}=\boldsymbol{A}^T\cdot\boldsymbol{\beta}\otimes\boldsymbol{\pi} \tag{2}$$

这里 $\boldsymbol{A}$ 表示的是状态转移矩阵，$\boldsymbol{\beta}$ 表示某观测矩阵 $\boldsymbol{B}$ 中的某一个观测值对应的观测向量，$\boldsymbol{\pi}$ 表示的是状态向量。

1.2 获取 HMM 的三要素的方法

在实验过程中，按照 Baum－Welch 算法对构成中文的语料库进行训练。获得隐马尔科夫模型的三要素 $\boldsymbol{\Phi}=(\boldsymbol{A},\boldsymbol{B},\boldsymbol{\pi})$，根据哈工大语言云得到结果，其中的隐含变量是词对的语义标注，观测向量是观测词汇，可以获得马尔科夫模型的初始化，随后，根据 Baum－Welch 或者 EM 算法进行迭代获得马尔科夫模型的三要素，其流程如图 2 所示。

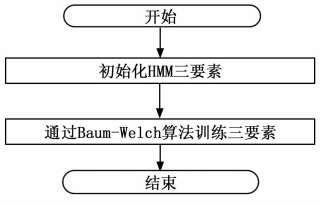

图 2 HMM 训练过程

1.3 MHMM 表示词对向量

根据马尔科夫链的一步转移概率，可以很清楚的知道某一个词对出现的概率。状态矩阵表示的是一步转移概率，而词汇的观测矩阵表示的词对在每个状态下的观测概率。因此，用公式（2）表示的是词对在语义特征上的抽取概率和。接下来用马尔科夫链的性质，说明算法的合理性。

定义 1：转移概率在离散序列马尔科夫链 $\{X_n\}$ 中，其具有有限或者无限的状态 $S=\{s_1,s_2,\cdots,s_n\}$，假设 $x=x(t_n)$ 表示序列 $t=t_n$ 时的状态，则条件转移矩阵表示的是在所有的观测序列当中从上一步转移到下一个状态的概率统计值，用其对应的频率来估计。即是说转移概率矩阵为 $\boldsymbol{P}=[p_{ij}]=\boldsymbol{P}\{x(t_{n+1})=s_j\mid x(t_n)=s_i\}$，其中 s_j，$s_i\in S$。

定义 2：观测向量表示的是，在整个观测当中，某一个观测值 o 在某一个状态上的概率值。假设观测向量用 β 表示，β 是一个 m 维的向量。其中假设：

$$\boldsymbol{\beta}=\{b_1,b_2,\cdots b_i,\cdots,b_m\}$$

其中：对于 bi 的解释表示如下。

$$b_i=P\{S=s_i\mid O=o\}$$

上式表示的是某一个观测词对在 s_i 状态下的概率。

用马尔科夫链的知识，很容易知道在整个马尔科夫过程中，条件转移概率用的是 n 个状态，而每个状态对应的概率为 b_i，那么一次词对向量对应的每个状态的输出就是：

$$\boldsymbol{P}=\boldsymbol{A}^T\boldsymbol{\beta} \tag{3}$$

如果再乘以每个状态对应的权重，上面是用 $\boldsymbol{\pi}$ 来表示的，那么就可以看成是这个词汇对在出现的每个对应的权重概率值。在公式（3）中，这个权重用 $\boldsymbol{\pi}$ 表示，这里的 $\boldsymbol{\pi}$ 表示的是每个状态在训练集中对应的状态的概率。

而且根据马尔科夫链的一步转移矩阵，以及在每个状态下的概率分布，很容易算出这个观测变量在每个状态下的概率。因此，用这个方法来表示一个语义词对，具有一定的科学性。

2 算法有效性验证

2.1 数据获取与预处理

本文的数据集通过网络爬虫，采集了豆瓣的影评数据，对采集的数据进行后续处理：第一部是将整个数据集进行过滤，把影评数据里面重复的字段删除；第二部是将单个测评数据当中连续几个重复的词条进行过滤；第三，去除里面的停顿词。第四部是将评分替换成差评和好评，标准是低于 6 分的判定是差评，高于 6 分的判定是好评。

对中文分词，从分词效果上来看，哈工大的自然语言处理工具的作用效果更好，分词的正确率较高，而且，本文考虑了隐含变量的运用，因此，在进行数据预处理的时候，选择了哈工大词云的语义角色标注。

在哈工大的词云上，根据词对的语义特征，将语义标注分成了：施事关系，当事关系，等等大约 100 种关系，而这种关系可以用这样的一个序列表示（x_1，x_2，…，x_n，r）其中 x_i 表示的是某一个词语，前面的 x_1 至 x_n 表示的是在语义标注里面存在的词汇，r 表示这 n 个词语之间的语义关系。

2.2 特征抽取

2.2.1 RNN

RNN 是循环神经网络，它的结构单元如图 3 所示。

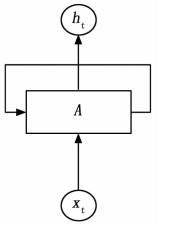

图 3 RNN 单元

RNN 的计算过程如下，假设输入序列是：

其中 $\boldsymbol{x_t}$ 表示的是一个 n 维的向量。

$$\boldsymbol{x_t}=[x_1,x_2,\cdots,x_n]$$

记忆单元的初始值为：

$$\boldsymbol{C}=[c_1,c_2,\cdots,c_n]$$

RNN 的激活函数为线性激活函数，输入权值矩阵为 $\boldsymbol{w_{in}}$，

输出权值矩阵为 w_{out}。根据前向算法，很容易得到的下面的算法：

$$C_i = w_{in}^T \cdot [x_i, C_{i-1}] \tag{4}$$

式中，$[x_1, C_{i-1}]$ 中表示将两个向量拼接在一起。那么输出就为：

$$o_i = w_{out} \cdot C_i \tag{5}$$

2.2.2　LSTM

同样的道理，LSTM 单元如图 4 所示。

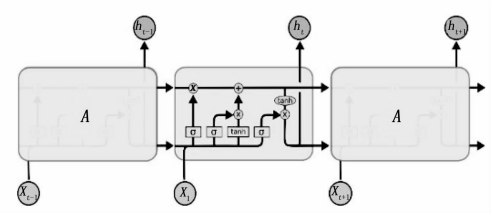

图 4　LSTM 结构示意图

可以从图中看出 LSTM 由：输入门、输出门和遗忘门，三个门进行控制其输出以及在细胞单元里面的输入值和输出值的变化，而且这三个门的权重值都是通过 LSTM 本身学到的。

按照上面的要求，可以得到如下的步骤（以下的 $\sigma(\cdot)$ 表示的是 sigmoid 函数）：

第一步：决定单元状态保留的信息，是通过遗忘门来实现的。对应的是图中的 f_t，其计算如下：

$$f_t = \sigma(w_f \cdot [x_t, h_{t-1}] + b_f) \tag{6}$$

第二步：通过输入门，来确定哪些值会被更新。此时，更新的值对应图中的 i_t 和$\tilde{C}_t$它们的计算公式如下：

$$i_t = \sigma(w_i \cdot [x_t, h_{t-1}]) + b_i \tag{7}$$

$$\tilde{C}_t = tanh(w_c \cdot [x_t, h_{t-1}] + b_c) \tag{8}$$

第三步：更新记忆状态，对应图中的 C_t，其计算过程如下。

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C} \tag{9}$$

第四步：最后输出 O_t 和 h_t，它们的计算过程如下：

$$O_t = \sigma(w_o \cdot [x_t, h_{t-1}]) \tag{10}$$

$$h_t = o_t \times tanh \cdot (C_t) \tag{11}$$

运用 LSTM 神经单元的多个层次对输入序列进行迭代会产生很多个输出，然后在实验过程中取出序列的最后一个输出作为句子向量。

2.2.3　句子特征抽取方式

进行特征抽取以前，通过哈工大语言云对原始数据进行语义分析，并将训练数据存储为 json 数据，作为训练数据，然后根据里面的词对训练 HMM 的精确参数（$\boldsymbol{A}$，$\boldsymbol{B}$，$\boldsymbol{\pi}$），其中 A 表示状态矩阵，B 表示观测矩阵，π 表示状态向量。训练 HMM 的过程如 2.2 介绍的那样。在进行特征抽取的时候采用的是 MHMM 模型，用改进的隐马尔科夫模型对一个语义词对进行表示。并且按照 MHMM 模型，获得句子当中出现的某一个词对出现的特征按照如图 5 所示的过程去进行词对特征抽取，并将其输入到 LSTM 神经元中，并且将最后的输出作为句子特征向量。

考虑到整个过程当中，每个词对形成的词对向量存在一定的稀疏性，因此，在对整个数据输入到 LSTM 之前，运用 softmax 函数对数据进行归一化处理。

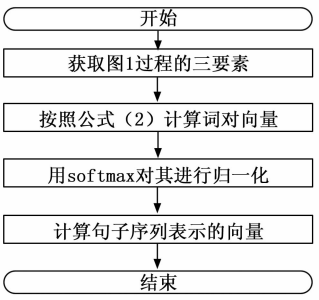

图 5　词对向量的训练过程

接下来，将特征抽取以后的每个词对向量组成的序列，放入 LSTM 单元当中，用动态 RNN 对输入的词对序列迭代处理，神经单元的最后一个输出向量作为评论样本表示的特征向量。

2.3　情感分类模型建立

在构建模型之前，对模型输入的词对或者词汇进行了预训练，获得句子向量，然后根据句子向量和类别标签产生分类训练器。在实验过程中，建立了如下的分类器模型：

1）基于 word2vec 的 SVM：先将所有的词汇用 word2vec 进行训练产生了词向量，对每一条评论分词产生的序列进行遍历获得词汇向量后，用每个分量的其平均值表示句子向量，用这个句子向量输入到 SVM 中进行模型训练。

2）标准的 LSTM 算法：通过 word2vec 对词汇进行训练以后，把词汇按照分词顺序进行排列并按照这个词序在 word2vec 的模型中找到对应的向量，输入到 LSTM 的神经元中，获得其句子的向量，最后按照这个向量进行 3000 次的迭代产生，使用交叉熵作为优化器，采用随机梯度下降法进行优化求得最优值。

3）基于词对的 LSTM 算法：通过 word2vec 对词汇进行训练以后，将三个在语序上相邻的词汇放在一起，求其平均值，然后将这些平均值作为输入，输入至 LSTM 单元中进行计算，其训练器的优化过程同上。

4）基于 MHMM 的 LSTM 算法：通过 3.2.3 的过程进行特征抽取，然后输入到 LSTM 的输入单元中，其优化过程和迭代过程同上。

5）基于随机向量的 LSTM 算法：对词向量的初始化采用的是随机向量，输入到 LSTM 单元当中，其训练和优化的过程与第二种方式基本相同。

2.4　MHMM 用于情感分类

基于 MHMM 的 LSTM 实验做法是将评论集通过训练马尔科夫模型的三要素，得到每个单词对的词向量，将它们 LSTM 的一个输入，再将序列按照顺序逐个输入到 LSTM 神经元中，其处理结构如图 6 所示。

输入层：经过 LTP 产生的语义词对。

MHMM：经过 MHMM 对产生的词对进行处理，得到每个词对在语义上产生的概率分布，然后用 softmax 进行归一化处理，产生的输出。

Z：经过 MHMM 产生的输出，经过归一化处理，用来训练神经网络。

LSTM：LSTM 用来作为特征提取的一个工具，每个序列都会产生一个对应的输出。

Softmax 层：用 softmax 层作为分类的依据。

根据以上的说明，MHMM 用于 LSTM 模型的情感分类

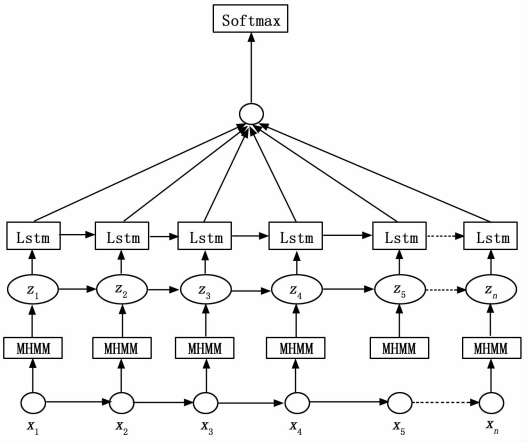

图 6 基于 MHMM 的 LSTM 情感分类图

算法如算法 1 所示。

算法 1：

输入：无标签的数据 D，训练集 D－train，测试集 D－test

输出：测试集的情感标签

1)将训练集和测试集中的数据进行预处理

2)获取隐马尔科夫参数(A，B，π)，并且用来得到词汇对的向量表示

3)初始化 LSTM－RNN 的参数，训练模型

4)For Sentences s in D－train：

a)对于 s 中的每个词汇，找到其对应的词向量，放入输入层

b)通过 LSTM－RNN 产生输出，用输出的最后一个向量作为 softmax 的输入。

c)通过 softmax 层产生分类的依据

d)通过反向传播调节参数获得最后的模型

End for

5)导出模型，用于测试集的情感分类

6)For Sentences s in D－test：

a)对于 s 中的每个词汇，找到其对应的词对向量，放入输入层

b)通过 LSTM－RNN 产生输出，用输出的最后一个向量作为 softmax 输入。

c)通过 softmax 层产生分类

End for

3 实验结果分析

3.1 实验测评参数的定义

本文采用正确率，召回率和 f－measure 对分类产生的结果进行测评，在进行测评之前首先对几个符号进行定义：

TP：通过分类算法，将原本的正类预测成为正类的数目；

FN：通过分类算法，将原本的正类预测成为负类的数目；

TN：通过分类算法，将原本的负类预测成为负类的数目；

FP：通过分类算法，将原本的负类预测成为正类的数目；

那么，可以定义以下的公式进行分类的测评。

正确率：

$$accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} \tag{11}$$

召回率：

$$recall = \frac{TP}{TP + TN} \tag{12}$$

$f-measure$：

$$f - score = \frac{2TP + TN}{2TP + FP + FN} \tag{13}$$

3.2 实验结果及实验结论

3.2.1 训练阶段

在爬取的豆瓣影评数据集中抽取四万条左右的评论数据，经过哈工大词云进行语义角色标注，获得词对表示的语义标签，把词对作为观测矩阵、把语义标签作为潜在的隐含特征，通过训练 HMM 的三要素获得了词对的表示，并且将这些语义用来训练 HMM 模型。对这四万多条的数据进行统计其评论情感极性见表 1。

表 1 训练集情感极性统计结果表

训练集	
正例	负例
21300	21050

并且将这些特征输入到 LSTM－RNN 模型中，并且抽取最后的一个词汇特征作为神经网络的输入，再进行分类，在训练的时候，选择的优化器是交叉熵，使用随机梯度下降法进行学习，其学习率设置为 0.0001，在训练模型的时候，每次用于训练模型的评论集的 batch 大小设置为 512 条，经过大约 2 千万次的训练，各个模型－算法正确率见表 2。

表 2 数据集情感极性统计结果表

特征选取－算法	正确率/%
Word2vec－SVM	84.76
词对向量－LSTM	96.48
MHMM－LSTM	97.40
随机向量－LSTM	86.70
Word2vec－LSTM	92.53

从表中，可以很容易得出的是，运用 MHMM 进行特征抽取的效果比词对向量进行特征抽取的效果要高 1 个百分点，用词对向量进行特征抽取比标准的 word2vec 进行词向量的表示要 11 个高百分点，因此 MHMM 的在特征的抽取上有较好的作用效果。

3.2.2 测试阶段

通过对测试集的情感极性分析获得了数据统计情况见表 3。

表 3 测试集情感极性统计结果表

测试集	
正例	负例
15000	15000

通过对数据的测试，获取了的测评数据包括：模型的正确率，模型的召回率和模型的 $f-score$，其详细情况见表 4。

从表 4 中可以看出来，使用三个模型进行特征抽取的时候，MHMM 的正确率比词对向量高出 1 个百分点，比标准的 word2vec 进行特征抽取词向量的模型高 0.04 百分点。召回率最大的是运用 word2vec 进行特征抽取的模型，其比词对向量高 2 个百分点，比 MHMM－LSTM 词向量进行特征抽取 3 个高百分点。$f-score$ 值最高的是使用 word2vec 进行特征抽取的，其比其他两个模型高出的百分点依次是：1.5 个百分点和 1.4 个百分点。

运用 MHMM 产生的词向量和空间相邻的词语产生的词向

(下转第 224 页)