文章编号：1003-0077(2016)05-0101-10

基于属性主题分割的评论短文本词向量构建优化算法

李志宇，梁 循，周小平

（中国人民大学 信息学院，北京 100872）

摘 要：从词向量的训练模式入手，研究了基于语料语句分割(BWP)算法，分隔符分割(BSP)算法以及属性主题分割(BTP)算法三种分割情况下的词向量训练结果的优劣。研究发现，由于评论短文本的自身特征，传统的无分割(NP)训练方法，在词向量训练结果的准确率和相似度等方面与 BWP 算法、BSP 算法以及 BTP 算法具有明显的差异。通过对 0.7 亿条评论短文本进行词向量构建实验对比后发现，该文所提出的 BTP 算法在同义词(属性词)测试任务上获得的结果是最佳的，因此 BTP 算法对于优化评论短文本词向量的训练，评论短文本属性词的抽取以及情感倾向分析等在内的，以词向量为基础的应用研究工作具有较为重要的实践意义。同时，该文在超大规模评论语料集上构建的词向量(开源)对于其他商品评论文本分析的应用任务具有较好可用性。

关键词：在线评论；短文本；词向量；相似度计算

中图分类号：TP **文献标识码**：A

Improving the Word2vec on Short Text by Topic: Partition

LI Zhiyu, LIANG Xun, ZHOU Xiaopin

(School of Information, Renmin University of China, Beijing 100872, China)

Abstract: We propose a method for Word2vec training on the short review textsby a partition according to the topic. We examine three kinds of partition methods, i. e. Based on Whole-review (BWP), Based on sentence-Separator (BSP) and Based on Topic(BTP), to improve the result of Word2vec training. Our findings suggest that there is a big difference on accuracy and similarity rates between the None Partition Model (NP) and BWP, BSP, BTP, due to the characteristic of the review short text. Experiment on various models and vector dimensions demonstrate that the result of word vector trained by Word2vec model has been greatly enhanced by BTP.

Key words: online review; short text; word vector; similarity calculation

1 引言

随着社会化商务的发展，在线评论已经成为了消费者进行网络购物的重要参考决策因素之一[1-2]，同时也成为了包括计算机科学、管理科学以及情报分析等领域研究者在内的重要研究对象之一。通常而言，在线评论包括微博评论、商品评论、点评评论等评论类型，这里我们统称为“评论短文本”。以往关于评论短文本的应用研究主要集中在包括评论效用分析[3]、虚假评论识别[4-5]以及评论观点归纳[6]等方面。然而，这些应用研究往往都基于一个重要的语言模型基础，即统计语言模型。

相对于常规语料而言，如书籍、新闻、论文、维基百科等语料，评论短文本的语言学规范非常弱，省略、转义、缩写等现象非常普遍。如果利用传统的训练或者学习方法对评论短文本进行处理，效果并不理想。但从某种角度上来讲，评论短文本的在文法上的不规范，恰恰是另外一种形式的规范，即评论短文本自身特征的“规范”，由于评论短文本应用的普遍性，因此没有必要非要将评论短文本规约到常规的语料形式上进行处理，反之应该在最大限度上保留评论短文本的语料特征。

对于评论短文本的相关建模主要是从两个角度

收稿日期：2015-06-03 定稿日期：2015-10-15

基金项目：国家自然科学基金(71531012、71271211)；京东商城电子商务研究项目(413313012)；北京市自然科学基金(4132067)；中国人民大学品牌计划(10XNI029)；中国人民大学 2015 年度拔尖创新人才培育资助计划成果

出发：第一，利用 TF-IDF，点互信息、信息增益等，对评论短文本进行建模，从而分析评论之间的相似度或评论的情感倾向等；第二，通过构建“词向量(词袋法)”，将评论文本词语数值化。但这类建模方式往往需要依赖于情感词典、属性词典等人工构造的相关词典，具有较强的领域性，同时可扩展性较差。

随着自然处理技术的发展，神经网络逐步被引入到相关的文本处理技术中。2013年，谷歌研究团队的开源的 Word2vec 词向量构建工具[7]，引起了词向量应用研究热潮，被称为2013年最为重要的自然语言处理工具之一。随后，Word2vec 作为词向量的转换工具被用于包括短文本情感分析[8-10]以及短文本相似度计算[1, 11]等相关自然语言处理任务。虽然 Word2vec 的应用范围广泛，但是研究者用其建模时，往往直接按照 Word2vec 的模型配置：将每一条短文本语料(可能包含若干短句或长句)作为一个整体行进行输入。通常，在 Word2vec 的参数形式里面只考虑到了输入向量的维度、训练方法以及语料大小对模型造成的影响，却并没有考虑语料的输入形式对 Word2vec 模型训练结果造成的影响。我们研究发现，不同的评论短文本输入形式会对 Word2vec 的词向量训练结果造成明显的差异，因此有必要在 Word2vec 进行词向量训练前考虑输入语料本身的特征，对语料进行预处理后用以提升词向量的训练结果。

根据前文的阐述，对于 Word2vec 训练的预处理需要优化的问题是：对于给定的大规模评论短文本语料库 $C=\{R_1, R_2, R_3, \cdots, R_n\}$(语料库总评论数目为 n)，如何在可接受时间内训练得到一个较为精准的词向量模型。其目标为：对于给定的词查询 $Q=term$，返回与其最相似的前 K 个结果，其中 $term$ 的同义属性词个数为 P，那么，对于查询 Q 的词向量准确度为：$AC=P/K$。为了获得对于给定测试集 $T=\{term_1, term_2, term_3, \cdots, term_m\}$(测试集词条数为 m)的较高准确度，有以下三种基本途径。

1) 通过对词向量的训练算法中的训练层进行改进，采用不同的训练模型或者不同类型的神经网络，来获得更为精准的词向量模型。

2) 通过在训练算法的输入层对语料进行预处理，提高算法训练的准确率和召回率。

3) 通过对词向量的输出层进行后处理，提升应用接口的准确度。

本文中，我们将集中讨论如何通过第二种方式，即在输入层如何对语料进行预处理来提升词向量模型训练的精度，研究包括基于整句分割模式的预处理模式、基于分隔符分割的预处理模式以及基于属性主题分割的预处理模式对于训练模型的影响。在后面小节中，我们将详细阐述这些方案，并重点论述基于属性主题分割模式的预处理算法。

2 相关研究工作与研究背景

2.1 评论短文本的情感分析与属性提取

短文本(Short Text)是指那些长度较短的文本形式。通常情况下，短文本的字符长度不超过400，例如，Twitter/微博短文本、手机信息短文本、在线评论短文本、BBS 回复转帖短文本等[2, 12-13]。由于短文本具有字数少、信息聚合度高以及文本语言不规范等特征，使得针对短文本的分析与研究产生了较大的困难，其中具有代表性的则是针对微博短文本和评论短文本的研究，下面将主要对评论短文本的相关研究进行综述。

随着电子商务的高速发展以及淘宝、京东、大众点评等各类含有评论短文本网站的兴起，评论短文本已经成为消费者在做出购买决策之前的重要参考依据[14]。目前关于评论短文本的研究主要集中在：评论短文本的效用分析、评论短文本的真实性分析、评论短文本的决策影响分析等。但这些研究内容都会涉及两个主题，即：评论短文本的情感分析与评论短文本的属性抽取。

评论情感分析主要是对评论的情感倾向进行分析，包含三个层次：评论对象的属性层次、评论对象的层次以及评论篇章层次。其主要采用的方法是将文本简化为 BOW(Bag of Words)的形式，然后借助情感词典对评论短文本的情感倾向进行分析。其中，Word Net 等情感词典对于评论短文本的情感分析起到了重要的作用。例如，利用 Word Net 中词汇之间的相互关系(距离、语义联系等)来判断词语的情感倾向。但这也带来一个重要问题，即：Word Net 按照同义词集合组织信息，而同义词语不一定具有相同的褒贬倾向，这将导致对词语情感倾向的估计出现偏差[15]。换句话说，目前评论短文本情感分析存在的主要问题是如何针对评论短文本的特征构建情感词之间的数值联系，即词向量的问题。

评论的属性抽取是评论短文本分析的另外一个重要的研究内容，即如何判断和抽取评论中涉及到的商品属性或称对象属性的相互关系。例如，“衣服

手感不错！”和“衣服摸起来不错！”中，词语“手感”和“摸起来”都是同样表达评论者对评价对象（衣服）的质量属性或者感官的判断。因此需要在对评论短文本进行分析时，能够成功地发现和评价这类属性的相互关系。评论短文本属性的抽取对于评论属性情感分析和评论总结都具有重要的作用。

总而言之，评论短文本的分析需要依赖于对评论短文本的形式化（数学化）建模，通常而言，需要在原有文本分析技术的基础上，结合短文本的自身特征进行改进，设计出有效的短文本语言模型的建模方法，以提高应用的效率和准确率。

2.2　词向量和 Word2vec

语言模型是自然语言处理（Nature Language Processing，NLP）领域的一个重要的基础问题之一，它在句法分析、词性标注、信息检索以及机器翻译等子领域的相关任务中都有重要的作用。在传统语言模型中，统计语言模型具有非常广泛的应用，其核心思想是利用概率来对语言形式进行预测[16]。通常而言，统计语言模型都基于相应的领域语料来进行分析工作。一般的，用以简化统计语言模型的相关方法包括：N-gram 模型、马尔科夫模型、条件随机场模型、决策树模型等。

随着深度学习相关研究的逐步深入，神经网络的应用领域逐渐由图像、音频等扩展到了自然语言处理领域，即神经网络语言模型（Neural Network Language Model，NNLM），NNLM 可以看作传统统计语言模型的扩展与提升，并于近年在 ACL、COLING 等相关顶级会议上取得系列进展。NNLM 具有代表意义的系统研究由 Bengio 于 2003 年在*A Neural Probabilistic Language Model* 一文中提出[17]，在该模型中作者将每一个词汇表示为一个固定维度的浮点向量，即词向量（*Word Vector*）。然而，NNLM 中的词向量（记为 NWV）和传统统计语言模型中的 One-Hot Representation（OHR）有着本质上的差异，主要体现在以下三点。

1）OHR 中的向量元素采用 0，1 表示，词向量中所有的分量只有一个数值为 1，其余分量全部为 0，而 NWV 的分量由浮点数构成，其取值为连续值。

2）OHR 的向量维数不固定，通常根据词典的大小而发生改变，并且一般较为庞大，容易造成维数灾难[17]，而 NWV 的维度通常根据具体的应用固定在 50～1 000 左右，具有可接受的时间复杂度。

3）OHR 的词向量元素并不包含统计语义或语法信息，通过 NNLM 的研究发现，NWV 通过向量间的相互计算，可以进一步拓展或表达出相应的语义和语法特征。

词向量是 NNLM 实现后的关键产物，在 Bengio 的工作之后，出现了一系列关于词向量的实现与构建的相关工作，包括 Tomas Mikolov[18-19]、Google 的 Word2Vec[7] 等。其中 Google 于 2013 年开源的 Word2vec 作为重要的词向量训练工具，在情感分析、属性抽取等领域，取得了一系列的应用成果[11, 20-21]，同时，词向量训练的好坏对于提升应用成果的性能具有重要的意义。但通常情况下，即使采用相同的 Word2vec 工具，不同类型或大小的语料库以及不同的向量维度都会对词向量的训练结果好坏造成影响。

因此，本文主要从探讨 Word2vec 训练词向量的优化方式入手，重点研究了不同的中文语料的预处理策略对于词向量训练结果的优化程度，特别的是对中文评论短文本——这一类重要的自然语言处理语料。本文主要贡献在于：首先，我们提出基于属性主题分割的短文本评论语料预处理算法，对比实验结果表明，该算法对于改善词向量的训练结果具有明显的提升效果；其次，我们获取了 0.7 亿条评论短文本数据，通过词向量模型的训练，并优化后得到了具备较高精度的词向量库（开源），该词向量对于其他与在线商品评论相关的（例如，评论情感分析、评论属性抽取等）自然语言处理任务具有重要的参考意义；最后，我们给其他领域关于词向量的训练优化研究提供了一定的参考思路：即针对特定的处理语料设计相关的预处理策略或许能够显著提升词向量的训练效果。

3　拆分词嵌入的评论短文本分割模式

3.1　基于完整句的分割模式（Based on Whole for Partition，BWP）

完整句子是指以句号、感叹号、省略号、问号以及分号分割后组成的句子形式[22-23]。通常情况下，我们认为一个句子的结束是一种观点、态度和说明内容的结束。对于评论短文本而言，一条评论通常包含几个带有完整句分隔符的句子，这些句子表达的观点既有可能相似，也有可能不同。换句话说，这些句子之间既有可能存在逻辑之间的联系性，也有可能是相互独立的。因此，当这些句子在语法上或观点上是相互独立，甚至截然相反时，如果将这些句

子作为一个整体输入，用以词向量的训练，将会给训练模型带来较大的误差。

基于完整句的分割模式是指利用以句号、感叹号、省略号、问号以及分号作为完整句的指示分割符，对一条评论中的句子进行拆分。同时考虑到评论文本的统计信息（表1），当不含完整句分隔符的评论语句的连续字符长度达到23时，我们将进行人工截断，自动将该句划分为一个整句。

3.2 基于分隔符的分割模式（Based on Separator for Partition，BSP）

相比于传统的文本语料，评论短文本在句点符号的使用上更加的随意，内容上也更加丰富和自由，包括含有各种表情符号、缩写、拼写错误以及不规范的断句符等。如图1所示，该图为淘宝商城某商品评论页面的截图。可以看到，对于评论短文本而言，其观点句通常在一个分隔符之内进行表达，并且长度更"短"，同时在语法规范上也表现得尤为不足。

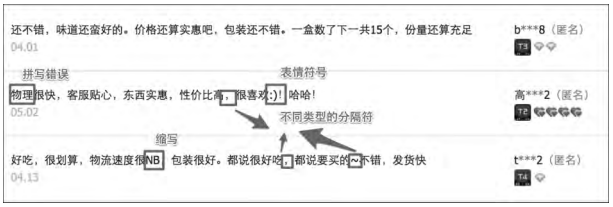

图1 评论短文本案例截图

如图2所示，对于评论"宝贝做工不错，物流速度马马虎虎！客服态度很好！"而言，这是一种典型的评论短文本的出现方式，即：观点句1＋观点句2＋…＋观点句n。但观点句之间很有可能存在修饰词不兼容（即观点句1的修饰词不能用于观点句2的情况）以及观点句情感极性相反的情况。如果采用前文所述的BWP分割方式，由于消费者撰写评论时使用符号的不规范，极有可能造成不同的意义、类型和观点的语句被分割到同一个训练语句中，从而增大模型训练的误差。因此，这也就是我们在实验对照中使用第二种分割方法，即分隔符分割法。

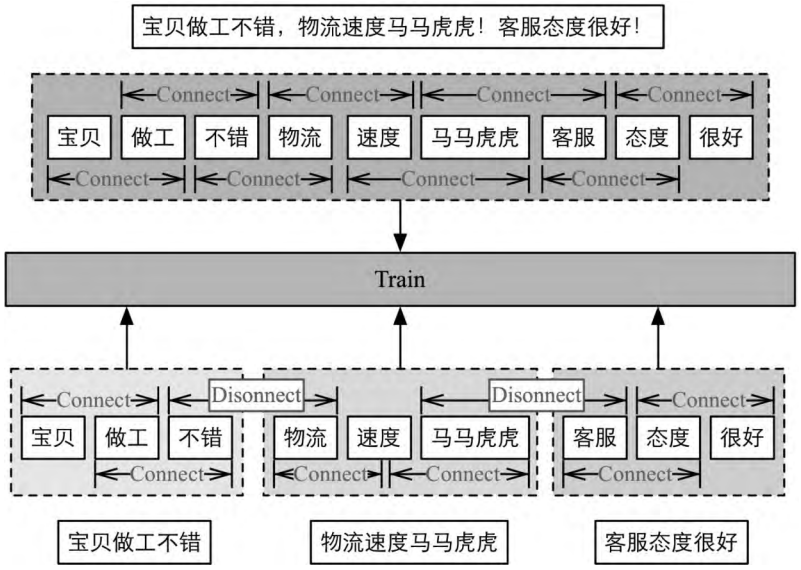

图2 基于分隔符分割模式与原始非分割方法的对比例图

基于分隔符分割的策略，目的是将这些观点句利用分隔符进行拆分。通常而言，评论短文本中的分隔符包括：（。）、（，）、（；）、（、）、（空格）、（！）、（～）、（＃）、（…）、（＊）、（：）、（-）、（？）、（"）、（"）、（＋）、、（-）以及（常见表情符号）等，同时，如果以上符号在评论短文本中存在西文格式，将同样认为是评论文本分隔符。

3.3 基于属性主题的分割模式（Based on Topic for Partition，BTP）

在研究中我们发现，虽然基于BSP分割能够将含有不同修饰符和不同属性观点的评论语句进行分割，以保证训练算法在这类评论上的稳定性，但BSP分割法却无法对评论中存在相互联系的，甚至是同类的评论语句进行合并。因此，在BSP的基础上，我们提出了基于属性主题的分割算法。

如图 3 所示，BTP 算法在 BSP 的基础上，考虑了一条评论中，被分隔符分割的评论句子之间的在主题上的相互联系。采用 BSP 对评论文本进行预处理后，利用词向量训练算法进行训练，得到初始的词向量模型，然后利用该初始词向量模型对 BSP 分割进行重构，合并属性主题相关的句子，在保证不同类型观点句得到有效分割的同时，保证了同类型观点句的关联性，具体算法流程如算法 1 所示。

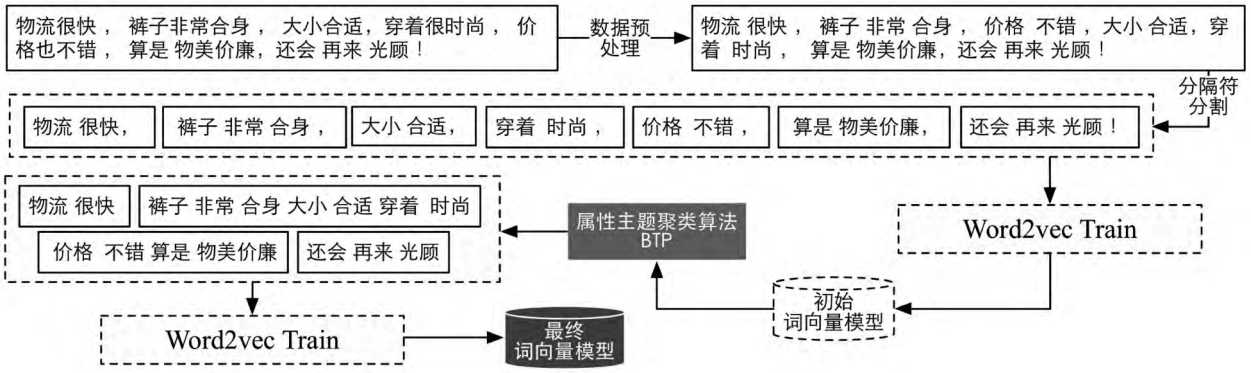

图 3　基于属性主题的词嵌入分割模型

算法 1 的核心思想：首先通过分隔符对评论进行整体拆分，然后利用 BSP 训练得到的词向量来计算相邻的每个最短分割候选句之间的属性相关度。其中，一条最短分割候选句的属性特征由短句中的名词词向量（或者数个名词词向量的均值）替代，如果候选短句不包含名词，则用形容词替代。最后，接着使用类似层次聚类的方式，对最短候选句进行逐项合并，直至满足退出要求，然后返回分割结果进行 BTP 模型的词向量训练。

算法 1：基于属性主题切割的词嵌入训练算法（BTP）

输入：$M_s=\{(W_x,V_x)\}$，$C=\{R_1,R_2,R_3,\cdots,R_i\}$，$R_i=\{P_1,P_2,P_3,\cdots,P_j\}$，$P_j=\{W_1,W_2,W_3,\cdots,W_x\}$
/＊M_s：基于分隔（S）符切割训练的词向量结果，W_x 为词语，V_x 为该词语对应的词向量；C：已经经过预处理的评论语料库；R_i：对于每一条已处理评论，由 j 个分隔句组成；P_j：对于每个分隔句，由 x 个词语组成；＊/

输出：$M_T=\{(W_x,V_x)\}$　　/＊基于属性主题（T）切割训练的词向量结果＊/

```
for Ri in C do:
    Sentence = [], Vector = []          /* 初始化分割结果，词向量临查询结果列表 */
    m=0,n=0                             /* 始化指针 */
    for Pj in Ri do:
        for Wx in Pj do:
            if Wx is Noun then:
                Vector[m][n]=[Wx find_vec(Ms)]   /* 查询该词对应 Ms 模型中对应的向量 */
                n+=1
            else:
                Continue
        end
        Sentence[m]=Pj                  /* 将查询得到的词对应的分隔句存入结果列表 */
        m+=1
    end
```

```
15      while  Merge[index]in Merge > 0.5 && Len(Merge) >3 do:
               /*只要已被处理的分隔句矩阵中存在任一两行的属性主题相似性的概率大于0.5,同
               时剩下有待被合并的行数大于3组,则合并计算继续进行*/
16             for index1=0; index1++; index1<m do Vector[m]=Mean(Vector[m])
               /*计算每一行所有名词向量的均值向量,作为该分隔句的属性特征向量*/
17             Merge=Similarity(Vector[m])
               /*利用余弦相似度依次计算相邻行分隔句之间的相似度,获得可能的合并概率*/
18             if Merge[index2] is the Max in Merge && Merge[index2]>0.5 then:
19                 Vector[index2]=Vector[index2]+Vector[index2-1]
                   /*合并相似的属性主题的特征向量*/
20                 Delete(Vector[index2-1])
21                 Sentence[index2] = Sentence[index2] + Sentence[index2-1]
                   /*合并相似属性主题的分隔句*/
22                 Delete(Sentence[index2-1])
23         end
24         Word2vec_Train(Sentence)   /*将分隔完成的主题相似性句子传入词向量训练模型*/
25  end
26  return M_T={(W_x,V_x)}            /*返回训练结果*/
```

4 实验数据

4.1 数据描述

本文的实验数据集来自天猫商城的评论短文本数据,主要字段包括:商品ID、评论者昵称、初次评论内容、初次评论时间、追加评论内容、追加评论时间、评论相对位置、评论者信誉、评论商品ID、评论商家ID以及商家回复。其中文本内容包括消费者的初次评论数据、追加评论数据以及商家的回复数据三个部分,总计评论数目为72 152 543条,约40GB。主要涉及领域包含:服装、食品、美妆、母婴、数码、箱包、家电、运户,共计八大领域的82个子领域。数据集的相关基本统计信息如表1所示。

表1 数据集基本信息

数据类型	基本统计信息	说明
初次评论平均字数	23.243	字数输入范围:[0,400]
追加评论平均字数	24.052	字数输入范围:[0,+1000)
商家回复平均字数	97.492	字数输入范围:[0,+1000)
消费者平均信誉值	T1.643	信誉取值范围:{T1,T2,T3,T4}

4.2 数据清洗

由于数据量巨大,因此数据清洗是本次实验的重要工作之一。本次实验过程中,为了提高数据的读取和操作性能,我们将评论数据存储在当前流行的非结构化数据库之一的Mongodb[24]中,其性能为普通SQL数据库性能的十倍以上,大大地缩短了实验的时间消耗。其中,数据清洗的核心步骤包括重复评论/无关评论的删除、分词、停用词的删除以及繁简体的合并操作。

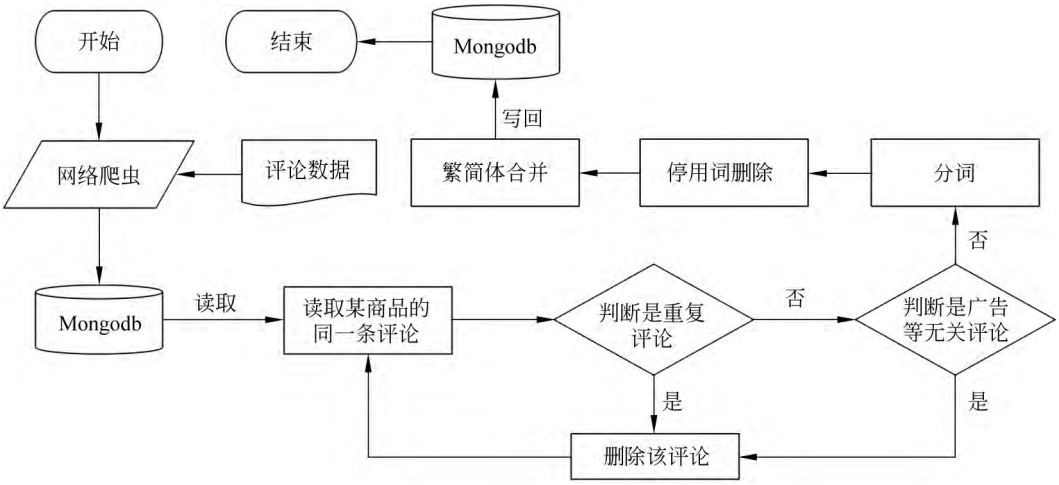

图 4　数据清洗流程图

5　实验与分析

5.1　性能评估

5.1.1　标准测试集

为了能够有效的测试三类预处理优化训练方案与原始语料(非优化)训练方案之间的差异，需要构建标准的同义(类)词测试集对四种不同类型的训练结果进行评价。具体的构建步骤为：选取待测试词组 100 个，即 $Q=\{Q_1,Q_2,Q_3,\cdots,Q_{100}\}$，查询每个词在每类已经训练好模型上的前 50 个相似词，组成 200 个备选词组。首先删除重复词汇，然后人工对测试词查询得到的 200 备选词进行筛选，得出所有的测试词的同义(类)词组，构成 $St=(Q_i\mid\{sim_1,sim_2,sim_3,\cdots,sim_n\})$ 标准测试词对集，其中，对不同的查询词 Q_i 其 n 值可能不同。

5.1.2　评价指标

在信息检索，模式识别，机器翻译等领域，有两类最为常用的算法评价指标，即：准确率(Precision Rate)和召回率(Recall Rate)。本文将参考准确率和召回率的评价方式，构建模型的评价指标，为便于说明，做出如下假设：

• 评价指标 1：平均相似度(S)

对于标准测试词对 St 中的查询词 Q_i，用其相似词构建评价词对为：

$$\{(Q_i\mid sim_1),(Q_i\mid sim_2),(Q_i\mid sim_3),\cdots,(Q_i\mid sim_n)\}$$

那么，对于一个标准查询词对 $(Q_i\mid sim_j)$，其在模型 X 中的向量组为 $(V_{Q_i}\mid V_{sim_j})$ 的相似度指标如式(1)所示。

$$S_i=\frac{\sum_{j=1}^{n}\frac{V_{Q_i}\cdot V_{sim_j}}{\|V_{Q_i}\|\ \|V_{sim_j}\|}}{n}\tag{1}$$

特例，如果查询词 sim_j 在模型 X 中不存在，那么对于查询对 $(Q_i\mid sim_j)$ 而言，其在模型 X 中的相似度为 -1。

则，对于测试集 $Q=\{Q_1,Q_2,Q_3,\cdots,Q_{100}\}$，模型 X 的平均相似度(%)指标如式(2)所示。

$$S=\sum_{i=1}^{100}S_i\tag{2}$$

• 评价指标 2：平均召回率

标准测试词对集 $S=(Q_i\mid\{sim_1,sim_2,sim_3,\cdots,sim_n\})$，查询词 Q_i 在模型 X 中的前 n 个最相似结果为：$T=(Q_i\mid\{Tsim_1,Tsim_2,Tsim_3,\cdots,Tsim_n\})$，那么对于查询词 Q_i，模型 X 的召回率如式(3)所示。

$$R_i=\frac{S\cap T}{n}\tag{3}$$

则，对于测试集 $Q=\{Q_1,Q_2,Q_3,\cdots,Q_{100}\}$，模型 X 的平均召回率(%)指标如式(4)所示。

$$R=\sum_{i=1}^{100}R_i\tag{4}$$

5.2　结果分析

为了验证和对比实验结果，本文的实验基于 MAC OS X 10.10.4 操作系统，Intel Core i7 4850Q 处理器(四核八线程)，16GB 内存，512GB SSD 存储系统，并采用 Python 语言进行实现。由于 Word2vec 的基础模型包含 Skip-Gram 以及 CBOW 两类，因此本文所有对比实验同时在这两种类型的基础模型上进行，具体的原始训练模型介绍可以参见 Word2Vec 的源码及其相关论文，此处不再详述。

最后，本实验针对不同的词向量的维度从50～500之间逐渐递增选取，增加纵向对比实验。

5.2.1 时间效率对比分析

如图5所示，通过对比发现，Skip-gram模型的处理时间对于不同大小的词向量维度的敏感度较大，随着词向量维度的增加，NP_Skip以及BSP_Skip模型的时间消耗增长幅度均大于CBOW模型的增长幅度。而NP模型与BSP模型在Skip-gram以及CBOW模型上的时间效率表现存在相互交叉的情况，因此并没有表现出明显的差异。考虑到无论是NP_Skip模型、NP_CBOW模型、BSP_Skip模型还是BSP_CBOW模型的单机训练时间均在[2,5]小时之间，因此，其实际意义上的时间开销（已经是0.7亿条评论大数据）均在可接受的范围内，所以并没有必要在时间效率上对上述模型进行不同的区分和优劣对比。

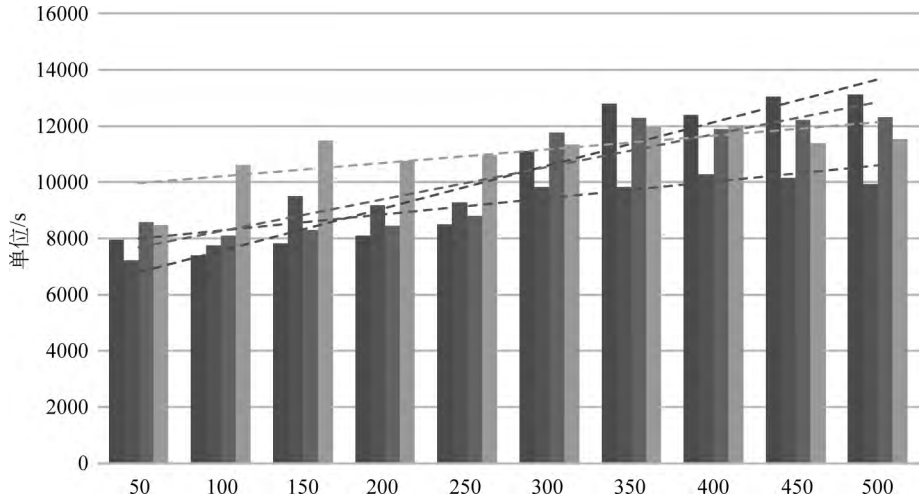

图5 BSP算法与原始训练算法基于不同词向量维度的时间效率对比

5.2.2 评价指标对比分析

• 平均召回率(R)

如表2所示，以直线下划线作为该模型的最好成绩，对比BTP模型与NP模型，在Skip_gram＋Herarchical softmax(SGH)和CBOW＋Herarchical softmax(CBH)实验上的平均召回率分别提升了23%和17%，其中，SGH_NP，CBH_NP最大召回率分别小于SGH_BTP，CBH_BTP的最小召回率，由此可以看出BTP语料预处理策略对于提升Word2vec训练结果的召回率具有显著效果。同时，我们可以发现，由于短评论语料通常字符数较小，并且断句符存在大量的不规范使用情况。因此，从NP模型到BWP模型的提升效果(2.3%，0.3%)远不如BWP模型到BSP模型的提升效果(12.3%，9.9%)以及BSP到BTP的提升效果(8.4%，7.6%)。

表2 模型实验结果对比

Dim	Skip_gram＋Herarchical softmax								CBOW＋Herarchical softmax							
	NP		BWP		BSP		BTP		NP		BWP		BSP		BTP	
	R	S	R	S	R	S	R	S	R	S	R	S	R	S	R	S
50	0.422	0.464	0.533	0.487	0.616	0.712	0.725	0.748	0.503	0.452	0.556	0.495	0.615	0.728	0.706	0.756
100	0.445	0.417	0.549	0.446	0.622	0.658	0.736	0.689	0.512	0.421	0.577	0.458	0.663	0.714	0.715	0.731
150	0.484	0.392	0.559	0.418	0.655	0.625	0.744	0.634	0.519	0.404	0.582	0.421	0.684	0.709	0.726	0.704
200	0.501	0.385	0.559	0.407	0.665	0.601	0.756	0.617	0.530	0.395	0.594	0.402	0.683	0.669	0.746	0.687
250	0.511	0.381	0.561	0.380	0.684	0.582	0.761	0.588	0.546	0.392	0.613	0.388	0.701	0.630	0.757	0.674
300	0.563	0.373	0.564	0.379	0.693	0.561	0.772	0.584	0.567	0.387	0.625	0.382	0.716	0.629	0.759	0.641
350	0.565	0.368	0.572	0.378	0.705	0.535	0.781	0.542	0.589	0.375	0.621	0.377	0.712	0.611	0.764	0.622
400	0.564	0.355	0.580	0.370	0.710	0.522	0.785	0.537	0.601	0.366	0.627	0.371	0.719	0.598	0.784	0.614
450	0.566	0.331	0.587	0.367	0.712	0.504	0.794	0.521	0.612	0.350	0.626	0.369	0.720	0.584	0.792	0.605
500	0.565	0.325	0.589	0.350	0.710	0.494	0.796	0.505	0.624	0.324	0.624	0.365	0.726	0.571	0.802	0.585

• 平均相似度(S)

由于不同的向量维度数会导致向量的分散程度不同：一般的，向量维数越大，在总词语数目固定的情况下，同义(属性)词间的分散程度越大，相似度越小(纵向)。因此平均相似度只能作为词向量训练好坏的一个相对参照指标，即：作横向对比。以表 2 中波浪下划线标注的 50 维度上的结果为例，对于召回相同的词语，其相似度越高，表示同义词(属性词)之间的稳定性越高，因此在不同的环境下其应用的可拓展性也就越高。从表 2 中可以看到，无论是对于 Skip_gram 模型还是 CBOW 模型，在不同词向量维度上，BTP 模型的稳定性都是最高的，但相对于 BSP 预处理模型来说，BTP 模型的提升程度却并不十分明显，因此如果在不考虑召回率的情况下，可以任选 BTP 或者 BSP 模型作为评论语料的预处理策略。

5.2.3　查询样例对比分析

为了能够对原始模型(NP)和 BTP 优化后模型产生的词向量的结果产生一个具体的认识和对比，我们选取了两个具有代表性的词汇“EMS”(属性词)以及“差评”(形容词，观点词)，查询了它们在 NP 词向量(200 维)以及 BTP 词向量(200 维)中的前 20 个最相似的结果，如表 3 和表 4 所示。

表 3　查询词“EMS”在 NP 模型和 BTP 模型上的对比结果

Skip_gram＋Herarchical softmax				CBOW＋Herarchical softmax			
NP		BTP		NP		BTP	
1. 圆通	11. 优速	1. ems	11. 速递	1. ems	11. 中通	1. ems	11. 汽运
2. ems	12. 平邮	2. 圆通	12. 中国邮政	2. 邮政	12. 慢递	2. 圆通	12. 快递
3. 申通	13. 汇通	3. 宅急送	13. 优速	3. 顺风	13. 邮局	3. 顺风	13. 中通
4. 宅急送	14. 中铁	4. 申通	14. SF	4. 宅急送	14. 想接	4. 顺丰	14. 速递
5. 中通	15. 顺丰	5. 顺风	15. 国通	5. 顺丰	15. 同城	5. 邮政	15. 物流
6. 韵达	16. 国通	6. 邮政	16. 快运	6. 圆通	16. 韵达	6. EMs	16. 邮局
7. 邮政	17. 慢到	7. 邮局	17. 全峰	7. 申通	17. 快运	7. Ems	17. 送货员
8. 全峰	18. 压货	8. Ems	18. EMs	8. 平邮	18. 速递	8. 宅急送	18. 中国邮政
9. 速递	19. 快运	9. 顺丰	19. MES	9. Ems	19. 顺丰速运	9. ESM	19. 快第
10. 顺风	20. 慢递	10. 韵达	20. ESM	10. 陆运	20. 快弟	10. 韵达	20. 陆运

表 4　查询词“差评”在 NP 模型和 BTP 模型上的对比结果

Skip_gram＋Herarchical softmax				CBOW＋Herarchical softmax			
NP		BTP		NP		BTP	
1. 真想	11. 添堵	1. 差平	11. 心甘情愿	1. 中评	11. 你们	1. 中评	11. 全 0 分
2. 中评	12. 故意	2. 低分	12. 坏评	2. 老子	12. 这种	2. 坏评	12. 说法
3. 忍无可忍	13. 别找我	3. 高分	13. 差品	3. 极度	13. 换算	3. 好评	13. 退货
4. 恶意	14. 无心	4. 全一星	14. 不错	4. 找麻烦	14. 因此	4. 差平	14. 退钱
5. 愤怒	15. 要挟	5. 认栽	15. 恶评	5. 你	15. 添堵	5. 二分	15. 认栽
6. 坚决	16. 认栽	6. 好评	16. 0 星	6. 差平	16. 上图	6. 一星	16. 零颗星
7. 宝贝	17. 你	7. 中评	17. 天理难容	7. 退换货	17. 哎	7. 最高分	17. 麻烦
8. 过不去	18. 垃圾	8. 一星	18. 千万不要	8. 换认	18. 退	8. 传说	18. 跪
9. 没商量	19. 抗议	9. 两星	19. 半星	9. 退货	19. 差到	9. 0 星	19. 0 分
10. 师	20. 好评	10. 查评	20. 零星	10. 换	20. 吃太甜	10. 一颗星	20. 零分

通过表3可以发现：BTP模型的预处理策略能够有效的发现属性词的相似词及其变异，甚至是错误的拼写词。例如，SGH_BTP模型中的“ESM、MES”(误输入)、“ems、EMs”(大小写变形)等。同时可以发现，BTP模型的属性词召回率明显高于NP模型。通过表4可以发现：BTP模型对于同义词的召回率同样较好，而NP模型中甚至出现了较多将查询词的被修饰词判定为相似词的情况，例如，真想(差评)，坚决(差评)等。但同时也需要看到，对于NP模型和BTP模型都出现了查询词的反义词被判定为相似词的情况，这种误判需要在后续的研究中进一步优化。

6 结论

Word2vec词向量训练的优化问题不仅仅需要考虑训练算法的内部结构，对于不同类型的训练语料的预处理同样值得思考。本文针对评论短文本在Word2vec词向量训练中存在的问题，结合评论短文本的自身特征提出了基于属性主题分割的语料预处理算法BTP。基于0.7亿条大规模评论短文本的实验表明，BTP算法的预处理策略对于提升词向量模型的训练结果具有显著意义。本文针对评论短文本的大规模词向量训练结果对于其他关于包括评论短文本情感分析、评论短文本属性特征提取(聚类)等的应用都具有较大的参考意义。

参考文献

[1] Yuan Y, He L, Peng L, et al. A New Study Based on Word2vec and Cluster for Document Categorization [J]. Journal of Computational Information Systems, 2014, 10: 9301-9308.

[2] 张剑峰，夏云庆，姚建民. 微博文本处理研究综述[J]. 中文信息学报，2012，26(4)：21-27.

[3] 杨铭，祁巍，闫相斌，等. 在线商品评论的效用分析研究[J]. 管理科学学报，2012，15(5)：65-75.

[4] 陈燕方，李志宇. 基于评论产品属性情感倾向评估的虚假评论识别研究[J]. 现代图书情报技术，2014，9：81-90.

[5] 任亚峰，尹兰，姬东鸿. 基于语言结构和情感极性的虚假评论识别[J]. 计算机科学与探索，2014，8(3)：313-320.

[6] Pang B, Lee L. Opinion mining and sentiment analysis [J]. Foundations and trends in information retrieval, 2008, 2: 1-135.

[7] Mikolov T. Word2vec project[CP]. 2013, https://code.google.com/p/word2vec/.

[8] Xue B, Fu C, Shaobin Z. A Study on Sentiment Computing and Classification of Sina Weibo with Word2vec [C]//Proceedings of the 2014 IEEE International Congress on, 2014: 358-363.

[9] Tang D, Wei F, Yang N, et al. Learning sentiment-specific word embedding for twitter sentiment classification[C]//Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 2014: 1555-1565.

[10] Godin F, Vandersmissen B, Jalalvand A, et al. Alleviating Manual Feature Engineering for Part-of-Speech Tagging of Twitter Microposts using Distributed Word Representations[C]//Proceedings of NIPS 2014Workshop on Modern Machine Learning and Natural Language Processing (NIPS 2014), 2014: 1-5.

[11] Ghiyasian B, Guo Y F. Sentiment Analysis Using SemiSupervised Recursive Autoencoders and Support Vector Machines[EB/OL], Stanford. edu, 2014: 1-5.

[12] 张林，钱冠群，樊卫国，等. 轻型评论的情感分析研究[J]. 软件学报，2014，12：2790-2807.

[13] 周泓，刘金岭，王新功. 基于短文本信息流的回顾式话题识别模型[J]. 中文信息学报，2015，291：015.

[14] 郑小平. 在线评论对网络消费者购买决策影响的实证研究[D]. 中国人民大学硕士学位论文，2008.

[15] 张紫琼，叶强，李一军. 互联网商品评论情感分析研究综述[J]. 管理科学学报，2010，13(6)：84-96.

[16] 邢永康，马少平. 统计语言模型综述[J]. 计算机科学，2003，30(9)：22-26.

[17] Bengio Y, Ducharme R, Vincent P, et al. A neural probabilistic language model[J]. The Journal of Machine Learning Research, 2003, 3: 1137-1155.

[18] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality[C]//Proceedings of the Advances in Neural Information Processing Systems, 2013: 3111-3119.

[19] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space. arXiv preprint arXiv: 1301.3781[DB\OL], 2013: 1-16.

[20] Zhang W, Xu W, Chen G, et al. A Feature Extraction Method Based on Word Embedding for Word Similarity Computing[C]//Proceedings of the Natural Language Processing and Chinese Computing, 2014: 160-167.

[21] Iyyer M, Enns P, Boyd-Graber J, et al. Political ideology detection using recursive neural networks[C]//Proceedings of the Association for Computational Linguistics, 2014: 1-11.

(下转第120页)

guistics (ACL), 2010: 384-394.
[19] Y Hong, X P Zhou, T T Che, et al. Cross-argument inference for implicit discourse relation recognition [C]//Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM), 2012: 295-304.
[20] C C Chang, C J Lin. LIBSVM: a library for support vector machines[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2001, 2(3): 389-396.
[21] 徐凡, 朱巧明, 周国栋. 基于树核的隐式篇章关系识别[J]. 软件学报, 2013, 24(5): 1022-1035.

朱珊珊(1992—),硕士研究生,主要研究领域为篇章分析。
E-mail: zhushanshan063@gmail.com

丁思远(1992—),硕士研究生,主要研究领域为事件关系检测。
E-mail: dsy.ever@gmail.com

洪宇(1978—),通信作者,副教授,主要研究领域为信息抽取,信息检索,事件关系检测等。
E-mail: tianxianer@gmail.com

(上接第110页)

[22] 黄建传. 汉语标点句统计分析[D]. 北京语言大学硕士学位论文, 2008.
[23] 何玉. 基于核心词扩展的文本分类[D]. 华中科技大学硕士学位论文, 2006.
[24] Banker K. MongoDB in action[M]. Manning Publications, 2011.

李志宇(1991—),博士研究生,主要研究领域为自然语言处理,网络结构嵌入,社会网络分析。
E-mail: zhiyulee@ruc.edu.cn

周小平(1985—),博士研究生,主要研究领域为社会网络分析,网络隐私保护。
E-mail: zhouxiaoping@bucea.edu.cn

梁循(1965—),通信作者,博士生导师,教授,主要研究领域为社会计算,机器学习。
E-mail: xliang@ruc.eud.cn