文章编号：

利用词的分布式表示改进作文跑题检测*

陈志鹏[1,2]，陈文亮[1,2]，朱慕华[3]

（1.苏州大学 计算机科学与技术学院，江苏 苏州 215006；2.软件新技术与产业化协同创新中心，江苏 苏州 215006；3.淘宝（中国）软件有限公司，浙江 杭州 311100）

摘要： 作文跑题检测任务的核心问题是文本相似度计算。传统的文本相似度计算方法一般基于向量空间模型，即把文本表示成高维向量，再计算文本之间的相似度。这种方法只考虑文本中出现的词项(词袋模型)，而没有利用词项的语义信息。本文提出一种新的文本相似度计算方法：基于词扩展的文本相似度计算方法，将词袋模型(Bag-of-Words)方法与词的分布式表示相结合，在词的分布式表示向量空间中寻找与文本出现的词项语义上相似的词加入到文本表示中，实现文本中单词的扩展。然后对扩展后的文本计算相似度。本文将这种方法运用到英文作文的跑题检测中，构建一套跑题检测系统，并在一个真实数据中进行测试。实验结果表明本文的跑题检测系统能有效识别跑题作文，性能明显高于基准系统。

关键词： 文本相似度；词分布式表示；跑题检测；文本表示

中图分类号：TP391　　**文献标识码：**A

Exploiting Distributed Representation of Words for Better Off-topic Essays Detection

CHEN Zhipeng[1,2], CHEN Wenliang[1,2],ZHU Muhua[3]

(1.School of Computer Science and Technology, Soochow University ,Suzhou ,Jiangsu 215006, China; 2.Collaborative Innovation Center of Novel Software Technology and Industrialization, Suzhou ,Jiangsu 215006, China;3.Taobao (China) Software Co., Ltd,Hangzhou,Zhejiang 311100, China)

Abstract: Similarity measure is the core component of off-topic essays detection. For computing similarity, the bag-of-words model is widely used. The model represents a text as a vector in which each dimension corresponds to a word, and then computes text similarity. Obviously, such a model leaves out the word semantic information. This paper proposes a new method to compute text similarity: a method exploits word distributed representation. The new method combines the traditional bag-of-words model with the word semantic information. For each word in a text, we search for a set of similar words in a text collection, and then extend the text vector with these words. Finally we compute text similarity with the updated text. Experimental results show that our new method is more effective than baseline systems.

Keywords: text similarity; word distributed representation; digress test; text representation

1 引言

作文跑题指文章偏离了预设的主题。举个例子，例如现在有一个题目“online shopping”,很明显是要求写关于网上购物的文章。如果学生写的文章与此无关，而是写的其他主题的文章，比如写的是关于读书的文章或者是关于大学生活的文章，我们就认为该作文跑题。作文的质量和是否跑题没有必然联系，有的文章虽然写的很短很差，但是并没有跑题。作文跑题的原因很多，可能是作者有意为之，也可能是无意间的提交错误。

* **收稿日期：**　　**定稿日期：**

基金项目：国家自然科学基金资助项目（61203314, 61333018）

作文跑题检测用于判断一篇作文是否跑题，其核心是计算文本之间的相似度，根据相似度和跑题标准来判断文章是否跑题[1]。文本相似度是表示两个文本之间相似程度的一个度量参数。除了用于文章跑题检测，在文本聚类[2]、信息检索[3]、图像检索[4]、文本摘要自动生成[5]、文本复制检测[6]等诸多领域，文本相似度的有效计算都是解决问题的关键所在。

目前最常用的文本表示模型是向量空间模型 VSM (Vector Space Model)。向量空间模型的基本思想是用向量形式来表示文本：$v_d = [w_1, w_2, w_3 ..... w_n]$，其中 w_i 是第 i 个特征项的权重。最典型的向量空间模型是词袋模型（Bag-of-Words）。该方法以文本中的词作为特征项形成向量表示，并且采用词的 TF-IDF 值作为特征权重[1]。词袋模型方法简单而且有一定效果，但是这种方法忽略了文本中词项的语义信息，没有考虑到词与词之间的语义相似度。例如“笔记本”和“手提电脑”这两个词在词袋模型中被认为两个独立的特征而没有考虑这两个词在语义上的相近性。

针对传统向量空间模型在文本相似度计算中存在的问题，很多研究人员进行了研究，其中词扩展是最常见的一种策略。现有词扩展方法主要采用基于词典的方法，比如使用 WordNet[7]、HowNet 等词典。文献[8]提出了基于 WordNet 词扩展计算英语词汇相似度的方法。文献[9]提出了基于 HowNet 计算词汇语义相似度的方法，并将其用于文本分类。这些方法严重依赖于人工构造的词典资源，在新语言和新领域应用中会遇到很多问题。

针对上述现有方法的不足，本文将词袋模型与词语的语义信息结合起来，提出一种基于词分布式表示[10]的文本相似度计算方法。对文本中单词进行分布式表示，即将它们映射为向量形式，然后在分布式的词向量空间中找出与其语义上相近的词，并将它们加入到文本表示中，最后再计算扩展后的文本相似度。本文将这种方法运用到英文作文的跑题检测中，构建了一套跑题检测系统，并在一个真实数据集上进行了测试。实验结果表明本文的跑题检测系统能有效识别跑题作文，性能明显高于基准系统。

本文的其余部分作如下安排：第 2 节对相关工作进行介绍；第 3 节详细介绍我们提出的计算文本相似度的方法。第 4 节介绍实验和结果分析，第 5 节是结论和下一步工作介绍。

2 相关工作

TF-IDF 方法是一种经典的基于向量空间模型的文本相似度计算方法。它用词的 TF-IDF 值来衡量它对于文本的重要程度，一个词的重要程度与它在文章中出现的次数成正比，但同时也会与它在语料库中出现的频率成反比。这里包含了两个重要的概念：

词频（Term Frequency），即一个词在文档中出现的次数：一个词在文章中出现的次数越多，它对这篇文章就越重要，它与文章的主题相关性也就越高。要注意的是停用词（stop words），像中文的“的”、“了”，英文的“a”、“the”，这些词并不具备这种性质，它们虽然出现的次数比较多，但是它们不能反映文章的主题。应该将它们过滤掉。

逆文档频率（Inverse Document Frequency），如果一个词在文档集合中出现的次数越多，说明这个词的区分能力越低，越不能反映文章的特性；反之，如果一个词在文档集合中出现的次数越少，那么它越能够反映文章的特性。例如，有 100 篇文档，如果一个词 A 只在 1 篇文档中出现，而词 B 在 100 篇文档中都出现，那么，很显然，词 A 比词 B 更能反映文章的特性。

将上面两个概念结合起来，我们可以计算一个词项的 TF-IDF 值，对于一个词项 (w_i)：

$$TFIDF(w_i) = tf(w_i) \times idf(w_i) \quad (1)$$

其中 $TFIDF(w_i)$ 表示当前词项 w_i 的 TF-IDF 值，$tf(w_i)$ 表示词项 w_i 的词频，$idf(w_i)$

[1] TF-IDF 是常用的特征权重计算方法。除此之外，亦可采用二元特征或者以词频作为权重。

表示词项 w_i 的逆文档频率，词项 w_i 的 $TFIDF(w_i)$ 等于 $tf(w_i)$ 乘以 $idf(w_i)$。很显然，词频就等于一篇文档中该词项出现的次数除以文章的总词数，而逆文档频率的计算公式是

$$idf(w_i) = \log \frac{N}{df(w_i)+1} \tag{2}$$

N 表示的是文档集合中文档的总数，$df(w_i)$ 是包含词项 w_i 的文档的总数，加 1 是为了保证分子大于 0。将公式（2）带入到公式（1）中，词项 TF-IDF 值的计算公式为

$$TFIDF(w_i) = tf(w_i) \times \log \frac{N}{df(w_i)+1} \tag{3}$$

根据上述公式计算出文本中每个词项 w_i 的 TD-IDF 值，然后利用这些 TF-IDF 值，将文档转化成一个向量空间模型，再利用余弦公式来计算相似度。余弦公式[11]如下：

$$Sim(D_1, D_2) = \cos\theta = \frac{\sum_{k=1}^{n} a_{1k} \times a_{2k}}{\sqrt{\sum_{k=1}^{n} a_{1k}^{\ 2} \sum_{k=1}^{n} a_{2k}^{\ 2}}} \tag{4}$$

其中，D_1，D_2 表示两个文本向量，a_{1k} 表示第一篇文章 D_1 中单词的 TF-IDF 值，a_{2k} 表示第二篇文章 D_2 中单词的 TF-IDF 值。

TF-IDF 方法是一种简单有效的计算文本相似度的方法，但是这种方法并没有考虑词语背后的语义信息，忽视了词与词之间的相似度。人们为了更准确的计算文本相似度，提出了一些基于语义的相似度计算法方法：文献[12]和文献[13]提出了基于本体的文本特征抽取和相似度计算方法。文献[14]提出了基于 HowNet 语义词典的文本相似度计算方法。文献[15]利用 WordNet 语义词典研究局部相关性信息以此来确定文本之间的相似性。这些方法利用了特定领域的知识库来构建词语之间的语义关系，与基于统计学的方法相比准确率有提高，但是知识库的建立是一项复杂而繁琐的工程，需要耗费大量人力。与上述方法不同的是，本文将词进行分布式向量表示，在新的分布式表示空间，自动地找出与某个词项语义上相似的单词，将这些词加入到文本的表示中，然后再用传统的方法对文本进行相似度计算。

作文跑题检测的研究起于国外，目的是为了提高作文自动评分系统的性能。随着研究的深入，许多研究者提出了检测作文跑题的方法。文献[1]提出了一种不需要特定主题训练数据的跑题检测方法。文献[16]利用主题描述来检测作文跑题的方法，通过计算文章与主题描述的相似性来判断文章是否跑题。和这些方法相比，本文的不同之处在于计算文章与范文的相似度来判断是否跑题，计算时采用了基于词分布式表示的词扩展方法，提高了检测系统的性能。

3 作文跑题检测

本文将词的分布式表示和向量空间模型结合，提出一种新的作文跑题检测方法。

3.1 词的分布式表示（Word Distributed Representation）

自然语言处理中，将一个词表示为向量的最简单、最常用方式是 One-hot Representation。这种方法把词表中的每个词表示为一个很长的向量，向量的维度是词表大小，其中绝大多数元素为 0，只有一个维度的值为 1，这个维度就代表了当前的词。比如：“笔记本”和“手提电脑”，“笔记本”的表示为[0,0,0,1,0.....0...]，“手提电脑”的表示为[0,0,0,0,0,0,1,0,0.....0...]。这种表示方法简单有效，不过忽视了词的语义信息，“笔

记本”和“手提电脑”是语义上近似的词，但这种方法表示出的向量却无法反映这点。

词的分布式表示（Word Distributed Representation）是指将词表中的词映射为一个稠密的、低维的实值向量，每一维表示词的一个潜在特征。这种方法基于深度学习，可以表示出词与词之间的联系。例如，“笔记本”表示成$[0.231,0.678,-0.535,0.178.....]$，“手提电脑”表示成$[0.032,0.561,0.233,0.411.....]$，向量的维数可以在训练前通过手工设定，是一个固定的值。虽然我们无法确切解释每一个维度具体表示什么，但是我们可以根据单词的向量形式找出与其语义上相近的词。

3.2 基于词分布式表示的词扩展

基于词的分布式表示，本节先进行词扩展，然后基于词扩展结果计算文档间相似度。基于词扩展的文档相似度计算具体描述如图 3-1 所示：

输入：范文 d_m，待比较文章 d_x

输出：文章的相似度 $Sim(d_m, d_x)$

程序：

1）对 d_m 和 d_x 进行文本预处理，得到处理后的文本单词集合 d'_m 和 d'_x

2)对 d'_m 和 d'_x 中的每一个单词 w 进行分布式表示，计算 w 与语料库中所有单词的相似度，并按照相似度对单词进行排序，选取与 w 相似度最高的 m 个单词加入到 d'_m 和 d'_n 中，得到新的单词集合 d''_m 和 d''_n 。

3）使用 $TF-IDF$ 方法计算 d''_m 和 d''_n 的相似度，即 $Sim(d_m, d_x)$ 。

图 3-1 基于词扩展的文本相似度计算

其中，第 1）步中文本预处理主要是去掉文章中的标点和数字，并过滤掉文本中的停用词。第 2）步中词扩展的具体描述如下：假设有文本单词集合 $d:\{w_1, w_2, w_3...w_i...w_n\}$，对于任一单词 w_i，找出与其语义上相近的 m 个单词集合 $\{v_{i1}, v_{i2}, v_{i3}.....v_{im}\}$，将这 m 个单词加入到原文本单词集合中，得到新的文本单词集合 $\{w_1, w_2, w_3.....w_n, v_{i1}, v_{i2}.....v_{im}\}$，对于文本中每一个单词，都进行这样的处理，得到一个扩展后的单词集合 $d':\{w_1, w_2, w_3.....w_n, v_{11}, v_{12}.....v_{1m}.....v_{n1}.....v_{nm}\}$。最后去掉集合中重复的扩展单词，得到最终的单词集合。第 3）步对于扩展后的文本表示使用 TF-IDF 方法计算相似度，得到的结果作为输入文章的相似度。

3.3 跑题检测

在本文跑题检测任务中，对每个作文题目给定 K 篇文章作为范文。利用上节描述的词扩展得到的文本表示，计算学生作文和范文之间的相似度。本文使用余弦相似度（Cosine）来计算相似度。假设给定的 K 篇范文集合记为 D，其中第 m 篇范文记为 d_m（$1 \le m \le K$），学生作文 d_x，则相似度计算过程如下：

首先，使用之前所述的方法计算范文与学生作文的相似度 $Sim(d_m, d_x)$，然后系统取均值作为最终相似度 $Sim(d_x)$，用如下公式计算：

$$Sim(d_x) = \frac{\sum_{m=1}^{K} Sim(d_m, d_x)}{K} \quad (5)$$

我们用最终相似度作为系统对文章的评分，将其与系统的阈值进行对比，以此来判断作文有没有跑题。

4 实验

本节先介绍实验数据，再介绍如何构造标准集，以及实验的评价方法，最后一部分是实验的结果和分析。

4.1 实验数据

本次实验中，我们收集了 10709 篇英文作文，共包括 20 个不同的题目，每个题目下有 500 篇左右的文章。这些文章都有教师对文章的总体评分，评分越高的文章写得越好，为了便于比较，在实验前，我们先对每个题目下的文章评分进行归一化处理，将文章的人工评分映射到 0 到 1 的范围。对于每个作文题目，选择人工评分靠前的 K 篇文章作为我们的范文。

为了学习词语的 idf 值和训练词向量，我们另外收集了 41225 篇不带评分的英文作文。

词向量的训练方法有很多，Bengio 等人提出 FFNNLM 模型[17] (Feed-forward Neural Net Language Model)可以训练出词的向量表示形式，不过 FFNNLM 并非是专门用来训练词向量，词向量只是训练模型过程中产生的副产品。Google 开源了一款专门用来训练词向量的工具 Word2Vec[18]-[20]，它可以根据给定的语料库，通过训练后的模型将词表示成向量形式，并能找出与某个词语义上相近的词。相比较 FFNNLM 模型，Word2Vec 对训练模型做出了优化，运行速度更快。我们的实验使用 Word2Vec 工具[2]来训练词向量。

4.2 构造标准集

标准集里面包含的是人工判断为跑题的文章的集合。由于文章数目较多，不可能人工检查所有文章，因此我们借助教师评分自动构造标准集。构造标准集的步骤如下：

（1）将各个题目下的文章按照人工评分从高到低排序。评分越高说明文章写得越好，这部分文章几乎不会跑题；而分数越低说明文章写得越不好，这里面可能就有跑题的文章出现。

（2）对于每个作文题目的文章，取得分最低的 10 篇文章，人工阅读每一篇文章，判断它有没有跑题，如果跑题则将它加入到标准集中。对于这 10 篇文章，如果它们全是跑题的文章，或者绝大多数是跑题的文章，就接着往上检查 10 篇文章，循环操作直到出现大部分的不跑题文章为止。如果这 10 篇文章只有少部分跑题，或者完全没有跑题的文章，就完成该作文题目的跑题作文人工检查工作。

在构造标准集时，总共两名研究生参与标注，两人标注的一致性[3]为 0.831。对于两人不一样的判断，经讨论后确定正确答案，如果出现无法确定的就当作跑题作文加入到标准集中。

最后得到的标准集共有 54 篇文章。每个题目下的跑题文章数是不一样的，有的题目比较好写，没有文章跑题；而有的题目比较难写，相对而言，跑题文章较多。表 4-1 给出了不同题目下跑题文章的分布。

从表 4-1 中我们可以看出，有 13 个题目下没有跑题文章，占 65%，很大的比例；另外，有 3 个题目下跑题文章数在 1 到 5 篇之间；跑题文章数为在 6 到 10 篇之间和 11 篇以上的

---

[2] https://github.com/NLPchina/Word2VEC_java

[3] 一致性用 Kappa 值衡量，$Kappa=\frac{P_0-P_e}{1-P_e}$，其中 P_0 表示实际一致率，P_e 表示理论一致率

题目数都是 2 个。

表 4-1 跑题文章的分布

跑题文章数	0 篇	1-5 篇	6-10 篇	11 篇及以上
题目数	13	3	2	2

4.3 实验评价方法

我们利用准确率（Precision）、召回率（Recall）和 F1 值来评价系统。首先要构造标准集和预测集两个集合，标准集是正确答案的集合，按上述方法构造。预测集是系统预测答案的集合。我们用 M 来表示标准集合中元素的数目，N 表示预测集中元素的数目，假设预测集中有 K 个元素是标准集中的元素。用 P 来表示准确率，R 表示召回率，F 表示 F1 值，则计算方法如下：

$$P = \frac{K}{N} \tag{6}$$

$$R = \frac{K}{M} \tag{7}$$

$$F = \frac{2}{\frac{1}{P} + \frac{1}{R}} \tag{8}$$

为了更好地分析系统，我们计算召回率取不同值的时候的准确率和 F1 值，具体就是计算出当召回率为 0.1、0.2、0.3.....1.0 的时候的系统的准确率和 F1 值，以此作为我们评价系统的依据。

4.4 实验结果

本次实验，我们共构建了 4 套不同的跑题检测系统。除了上述的 TF-IDF 方法和基于词分布式表示的词扩展方法，还有另外两种方法作为比较：Word2Vec 方法和 Sent2Vec 方法。Word2Vec 方法是进行简单地替代和拼接。用单词训练出的词向量来代替 TF-IDF 方法中的 TF-IDF 值，然后再将所有单词的词向量首尾相连，拼接成一个长的向量，最后使用余项公式来计算相似度。假设之前 TF-IDF 方法中的文章表示为一个 $1\times M$ 的向量，每一维表示一个词的 TF-IDF 值，使用 Word2Vec 训练出的词向量是 N 维，用词向量代替 TF-IDF 值后，文章就表示为一个 $1\times MN$ 的向量。Sent2Vec 方法是使用 Sent2Vec 工具[4]，与 Word2Vec 不同的是它可以对句子进行分布式向量表示，我们将一篇英文文章看作一句话，然后训练出一篇文章的向量表示，直接用余弦公式计算文章之间的相似度。

图 4-1 和图 4-2 是选取 1 篇文章作为范文的实验结果，对于词扩展（WordExtend）方法，每个单词扩展了 50 个词：

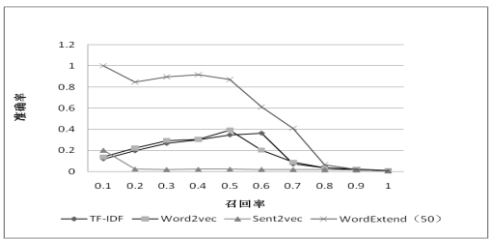

图 4-1 1 篇范文时准确率随召回率变化的曲线

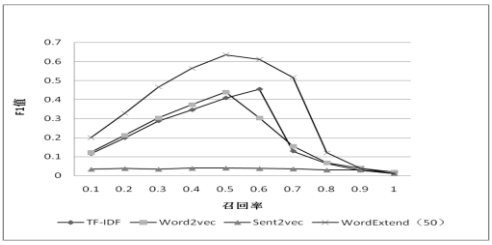

图 4-2 1 篇范文时 F1 值随召回率变化的曲线

[4] http://research.microsoft.com/en-us/downloads/731572aa-98e4-4c50-b99d-ae3f0c9562b9/

图 4-3 和图 4-4 是选取 5 篇文章作为范文的实验结果：

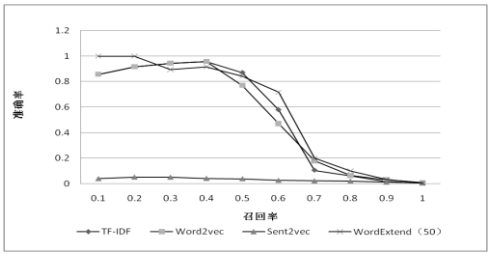

图 4-3 5 篇范文时准确率随召回率变化的曲线

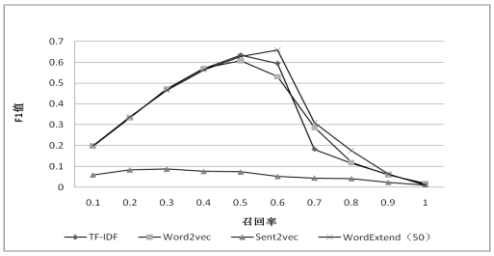

图 4-4 5 篇范文时 F1 值随召回率变化的曲线

从实验结果，我们可以看出：

（1）Word2Vec 方法性能略低于传统的 TF-IDF 方法，Sent2Vec 方法的性能最差，而词扩展方法的性能要明显优于其他 3 种方法。

（2）当范文数为 1 的时候，我们可以看到：R=0.6 的时候，TF-IDF 方法的 F1 到达到峰值，为 0.455，而词扩展方法的 F1 值为 0.611；TF-IDF 方法的准确率只有 0.363；而词扩展方法的准确率为 0.611，相比较之下，使用词扩展方法的系统的整体性能有明显的提升。

（3）当范文数为 5 的时候，TF-IDF 方法的 F1 值最高为 0.635，而词扩展方法的 F1 值的峰值为 0.66，略高于 TF-IDF 方法。

（4）另外，对比范文数为 1 的和范文数为 5 的结果。我们可以发现，范文数少的情况下，词扩展方法的效果比传统的 TF-IDF 方法明显要好很多。这是因为通过词扩展的方式，1 篇范文所包含的语义信息更加丰富，所以系统的判断也会更加准确。在实际使用中这点很有用，因为实际情况下一般不会提供太多范文。

5 结论和下一步工作介绍

本文提出了一种基于词分布式表示的作文跑题检测方法。这种方法将传统的 TF-IDF 方法和单词语义信息相结合，寻找与文本中单词语义上相近的词，并将其加入到文本的表示中，实现了对文本的词扩展。在此基础上，对扩展后的文本用 TF-IDF 方法计算相似度。实验结果表明这种方法要明显优于传统的 TF-IDF 方法。

在接下来的工作中，我们还会进行更深入的研究。比如，文中的词扩展数目是人工选取的 50 个单词，虽然效果提升明显，但还不是最优解，还有待于通过开发集来选取最优值。另外，还可以改进我们词扩展的方式，寻找一种更好的方式来将单词的语义信息融入到文本相似度的计算中。

参考文献

[1] D. Higgins, J. Burstein, Attali. Identifying off-topic student essays without topic-specific training data[J], Natural Language Engineering, 2006, vol. 12(2): 145-159.

[2] A.Huang. Similarity measures for text document clustering[C]//in Proceedings of the New Zealand Computer Science Research Student Conference, 2008, 44-56.

[3] KUMAR N. Approximate string matching algorithm [J].International Journal on Computer Science and Engineering, 2010, 2(3): 641-644.

[4] COELHO T A S,CALADO P P,SOUZA L V, 等. Image retrieval using multiple evidence ranking[J]. IEEE Trans on Knowledge and Data Engineering, 2004, 16(4): 408-417.

[5] KOY, PARK J, SEO J. Improving text categorization using the importance of sentences[J]. Information Processing and Management,2004, 40(1): 65-79.

［6］ THEOBALD M,SIDDHARTH J, SpotSigs: robust and efficient near duplicate detection in large web collection[C]. //Proc of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM Press, 2008: 563-570.

［7］ Miller G．Wordnet：An On-line Lexical Database[J]. International Journal of Lexicography, 1990, 3(4): 235-244.

［8］ 颜 伟, 荀恩东. 基于 WordNet 的英语词语相似度计算[C]//计算机语言学研讨会论文集. 2004.

［9］ 朱嫣岚, 闵锦, 周雅倩, 等. 基于 HowNet 的词汇语义倾向计算[J]. 中文信息学报, 2006, 20(1):14-20.

［10］Lee, Daniel D, H. Sebastian Seung. Algorithms for non-negative matrix factorization[C]//Advance in Neural Information Processing System.MIT Press,2001:556-562.

［11］张霞, 王建东, 顾海花. 一种改进的页面相似性度量方法[J]. 计算机工程与应用, 2010, 46(19): 141-144.

［12］Sánchez J A, Medina M A, Starostenko O, 等.Organizing Open Archives via Lightweight Ontolog to Facilitate the Use of Heterogeneous Collection[J]. Aslib Proceedings, 2012, 64(1): 46-66.

［13］Vicient C, Sánchez D, Moreno A. An Automatic Approach for Ontology-Based Feature Extraction from Heterogeneous Documental Resource[J]. Engineering Application of Artificial Intelligence, 2013, 26: 1092-1106.

［14］Liu Q, Li S J. Semantic Similarity Calculation Based on HowNet [C]// Proc of the 3rd Chinese Lexical Semantics Workshop. Taipei, China, 2002: 59-76.

［15］Ramage D, Rafferty A N, Manning C D. Random walks for text semantic similarity[C]//Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing. Suntec, Singapore, 2009: 23-31.

［16］A. Louis, D. Higgins. Off-topic essay detection using short prompt texts[C]//In Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications, Los Angeles, California, 2010:92-95.

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

［18］Tomas Mikolov, Kai Chen, Greg Corrado, 等. Efficient Estimation of Word Representations in Vector Space[C]//In Proceedings of Workshop at ICLR, 2013.

［19］Tomas Mikolov, Ilya Sutskever, Kai Chen, 等. Distributed Representations of Words and Phrases and their Compositionality[C]//In Proceedings of NIPS, 2013.

［20］Tomas Mikolov, Wen-tau Yih, Geoffrey Zweig. Linguistic Regularities in Continuous Space Word Representations[C]//In Proceedings of NAACL HLT, 2013:746-751.

作者简介：

陈志鹏（1991—），男，硕士研究生，主要研究领域为自然语言处理。Email: chenzhipeng341@163.com。

陈文亮（1977—），男，博士，主要研究领域为自然语言处理。通讯作者。 Email:wlchen@suda.edu.cn。

朱慕华（1981—），男，博士，主要研究领域为自然语言处理。Email:zhumuhua@gmail.com。