文章编号：

基于文档发散度的作文跑题检测*

陈志鹏[1,2]，陈文亮[1,2]

（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）

摘要：作文跑题检测是作文自动评分系统的重要模块。传统的作文跑题检测一般计算文章内容相关性作为得分，并将其与某一固定阈值进行对比，从而判断文章是否跑题。但是实际上文章得分高低与题目有直接关系，发散性题目和非发散性题目的文章得分有明显差异，所以很难用一个固定阈值来判断所有文章。本文提出一种作文跑题检测方法，基于文档发散度的作文跑题检测方法。该方法的创新之处在于研究文章集合发散度的概念，建立发散度与跑题阈值的关系模型，对于不同的题目动态选取不同的跑题阈值。本文构建了一套跑题检测系统，并在一个真实的数据集中进行测试。实验结果表明基于文档发散度的作文跑题检测系统能有效识别跑题作文。

关键词：跑题检测；文档发散度；文本相似度

中图分类号：TP391　　**文献标识码：**A

Off-topic Essays Detection Based on Document divergence

CHEN Zhipeng[1,2], CHEN Wenliang[1,2]

(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)

Abstract: Off-topic detection is important in the automated essay scoring systems. Traditional methods compute similarity between essays and then compare the similarity with a fixed threshold to tell whether the essay is off-topic. In fact, there is strong relationship between essay score and the type of topic. It is obviously different between divergent topic and non-divergent topic. It is hard to use a fixed threshold to identify off-topic for all essays. This paper proposes a new method of off-topic detection based on divergence of essays. The innovation of this paper is that we study the divergence of essays, and establish the linear regression model between divergence and threshold. In our method, we set a threshold for each topic dynamically. Finally we establish an off-topic detection system, and test in the real data. Experimental results show that our method is more effective than baseline systems.

Key words: off-topic detection; document divergence; document similarity

1 引言

作文跑题指文章偏离了预先给定的主题。举个例子，例如现在有一个题目“on food safety”,要求写关于食品安全的文章。如果学生写的文章与此主题无关，而是关于其他主题，比如读书或者关于大学生活,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[1]。

* **收稿日期：**　　**定稿日期：**

基金项目：国家自然科学基金资助项目（61572338）

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；陈文亮（1977—），男，博士，主要研究领域为自然语言处理。

作文跑题检测用于判断文章是否跑题，是作文自动评分系统的重要组成模块。传统的作文跑题检测系统一般计算内容相似度，将其与一个固定的阈值进行比较，然后判断文章是否跑题。这种方法简单有效，但是没有考虑文章得分和题目类型之间的关系，而是简单假设所有题目下的跑题阈值都一样。

针对传统方法的不足，本文提出基于文章发散度设立动态阈值的方法。这种方法考虑不同类型的题目，研究题目下文章集合发散度的概念，挖掘跑题阈值与文档发散度之间的关系，根据文档发散度动态选取题目对应的阈值，实验证明此方法可以提高检测系统的性能。

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作文跑题检测方法。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

作文跑题检测的核心是文本相似度计算。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。

传统文本相似度计算一般基于向量空间模型 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，一般用词的 TF-IDF 值[7]作为特征权重[1]。TF-IDF 值表示该词对于文本的重要程度，它由词频和逆文档频率构成：

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），像中文中的“的”、“了”，英文中的“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，可以计算出一个词项的 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 w_i 的词频，$idf(w_i)$ 表示 w_i 的逆文档频率。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如下：

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

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。

对于文本 D，基于向量空间模型，我们可以将 D 表示为向量 $[a_1, a_2, ..., a_k, ..., a_n]$，其中 a_k 为词

[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。

表中第 k 个单词对应的 TF-IDF 值。将文章表示为向量后，便可使用余弦公式计算向量间的相似度，以此来度量文本之间的相似度，公式如下

$$Sim(D_1, D_2) = \cos\theta = \frac{\sum_{k=1}^{n} a_{1k} \times a_{2k}}{\sqrt{\sum_{k=1}^{n} a_{1k}^{2} \sum_{k=1}^{n} a_{2k}^{2}}} \quad (3)$$

其中 D_1 和 D_2 表示两篇文本，假设词表中一共有 n 个词，a_{1k} 表示第一篇文本 D_1 中单词的 TF-IDF 值，a_{2k} 表示第二篇文本 D_2 中单词的 TF-IDF 值。

基于向量空间模型的文本相似度计算方法简单有效，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”这两个词在向量空间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。针对这一问题，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[8]、HowNet 等词典。Yan[9]提出了基于 WordNet 词扩展计算英语词汇相似度的方法。Zhu[10]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。近年来，随着深度学习的兴起，词向量获得了越来越多重视，许多研究者研究尝试将其融入文本相似度计算，Chen[1]提出利用词向量快速构建词项之间语义关系并进行词扩展，不需要依赖人工构造的字典，面对不同领域的作文检测也有较好的效果。

作文跑题检测源于对作文自动评分系统的研究。传统的作文评分系统，如 PEG[11]、IEA[12]、E-rater[13]等并未直接判断文章是否跑题，而是将内容相关度作为文章特征之一，利用分类或者回归的方法计算新文章的得分。这种方法直接给出文章总体得分，用户无法从中判断出文章是否跑题。针对这种不足，通用的方法是设定一个阈值，将内容相关度与阈值进行对比，以此来判断文章是否跑题。Louis[14]提出了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并与阈值进行对比来判断文章是否跑题。Ge[15]提出一种利用文本聚类来判断文章是否跑题，同样是设定相似度阈值作为聚类终止条件。这些方法相较于传统方法的优点是可以显示判断文章是否跑题，但是传统方法设置的都是固定阈值，即所有题目的阈值都相同，没有考虑不同题目的特点。

与上述方法不同，本文在研究题目发散性的基础上，提出一种设立动态阈值的方法。研究文本集合发散性值的概念和度量方法，分析发散性值和跑题阈值的关系，构建二者的线性关系模型。通过这种方法，我们可以动态计算出每一个题目下的跑题阈值。实验表明，相对于固定设定阈值，基于发散性的阈值设定方法有更好的性能。

3 基于文档发散度的作文跑题检测

本部分前两节详细阐述本文的创新点：文档发散度和基于文档发散度的跑题检测。最后一节介绍基于词扩展的文本相似度计算方法[1]，实验中用此方法计算文章和范文的相似度。

3.1 文档发散度

文档的发散度指的是某一题目下文章集合的发散程度。举个例子，比如有两个题目：“一场足球赛”和“一次难忘的经历”。相对而言，后者的作文集合会更加“多种多样”，不仅仅会有写足球赛的，可能还会有写旅游、料理等等主题的作文。这些文章所叙述的事情没有统一的主题，不同文章的内容之间也没有太多相似性，但是它们却没有跑题。像这样的题目，我们认为其发散度就比较高。这个题目也被称为发散性题目。

由于发散性题目下文章之间的相似性不高，差异较大，本文用文章之间两两相似度均值

来表示文章集合的发散程度。假设某一题目下有 M 篇文章 $\{D_1, D_2..D_m\}$，文章之间两两相似度的均值称为文章发散度值，记为 div，则有公式如下：

$$div = \frac{1}{Num} \sum_{\substack{1 \le i \le m \\ i \le j \le m}} Sim(D_i, D_j) \quad (4)$$

其中，Num 指 1,2,3..m 个数的组合数目，$Sim(D_i, D_j)$ 表示文章 D_1 和 D_2 的相似度，使用 TF-IDF 方法（即公式（3））计算。如果一个题目的发散度越高，则它的发散度值 div 就越低。

我们挑选了 10 个真实的题目，每个题目下都有 100 篇文章。计算出每个题目下文章集合的发散度值，按发散性值从低到高排序，如下表：

表 3-1 不同发散性值的题目及其发散性值

发散性	题目	发散性值（div）
高	Free topic	0.020
	Book Report	0.021
	The Most Impressive Campus Activity	0.058
	How to arrange my college life?	0.086
中	Pros and Cons of Mixed Marriages	0.149
	A Good Teacher	0.169
	Food Safety	0.211
	Internet and Privacy	0.273
低	Translation on page 59	0.401
	unit4 翻译	0.461

从表 3-1 中我们可以看到：发散性较高的题目，如 “free topic” 、“Book Report”，对应文章集合的发散性值比较低，而发散性较低的题目，如“Translation on page 59?”和“unit4 翻译”的发散性值相对来说比较高。

3.2 基于文档发散度的跑题检测

在本文跑题检测任务中，对于每一篇学生提交的文章，需要与范文计算相似度，然后与阈值对比。如果相似度小于阈值，则判断为跑题作文；反之，则为不跑题作文。显然，阈值的选取很关键，本文使用基于文档发散度的方法动态选取阈值。

每个题目下的跑题阈值是不同的，所以很难选取一个固定的经验值作为阈值。通过观察可知：发散性题目下，文章与范文相似度较低，阈值较低。而非发散性题目下，文章与范文的相似度较高，阈值相对而言较高。这意味着跑题阈值和发散性值之间是有联系的，我们假设二者之间存在着线性关系。

本文使用线性回归模型来构造文档发散度值与跑题阈值的关系。线性回归模型反应两种或者两种以上变量之间相互依赖的定量关系，应用十分广泛。

根据以上分析，本文假设发散度值和跑题阈值的关系如下：

$$thresholder = a \times div + b \quad (5)$$

其中，$thresholder$ 表示该题目的跑题阈值，div 为该题目下文章的发散度值，a 和 b 是模型的参数。线性回归是一种有监督的学习方法，所以我们需要搜集一定量的样本，对模型进行训练，得到模型参数 a 和 b。构建好线性回归模型后，我们只需要计算出题目下面文章集合的发散性值，就可以根据已经构建好的模型动态地计算出每个文章下面的跑题阈值。

3.3 基于词扩展的文本相似度计算

在计算文章与范文相似度的时候，本文使用基于词扩展的文本相似度计算方法。该方法由 chen[1]等人提出，在计算文本相似度的时候利用单词的语义信息，快速有效。

传统的文本相似度计算方法如之前所述，采用基于向量空间模型的 TF-IDF 方法。基于词扩展的相似度计算方法是对于传统方法的改进：对于某一个文本单词集合，找出其扩展的相似词集合，将其加入到原来的文本集合中，得到新的文本表示集合。在这个新的文本表示集合的基础上，使用 TF-IDF 方法计算相似度。具体来说：

对于文章 D，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 k 个单词集合 $\{v_{i1},v_{i2},v_{i3}.....v_{ik}\}$。对于 d 中所有单词，我们都找出它们的相似词集合，得到一个总的相似词集合 $E=\{v_{11},v_{12}...v_{1k}...v_{i1},v_{i2}.....v_{ik}\}$。去除集合 E 中重复的扩展词，得到最终的扩展集合 E'。最后将 E' 加入到原文本单词集合中，得到最终用于计算的文本表示单词集合。

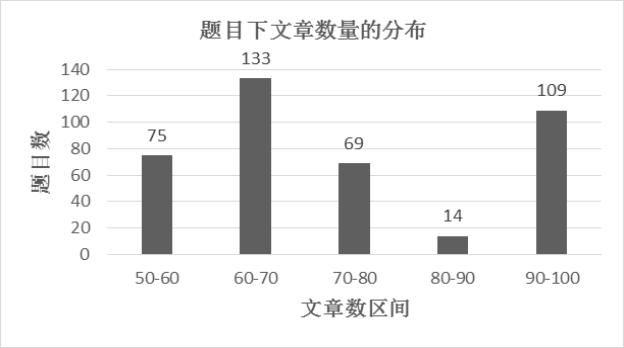
在计算文章和范文之间相似度的时候，对两篇文章都进行词扩展，然后使用 TF-IDF 方法计算相似度。

4 实验

4.1 实验数据

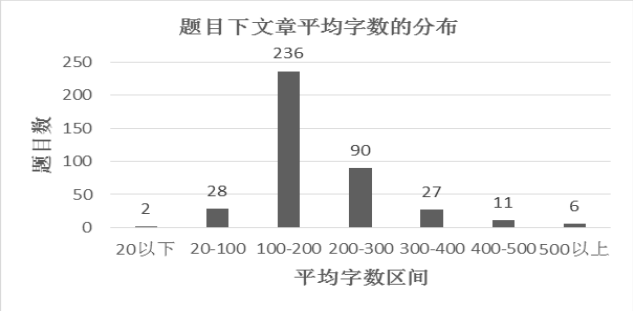
本次实验中，我们向合作机构申请了 30111 篇不同的文章用于实验，一共 400 个不同的题目。平均每个题目下有 75 篇文章。这些题目下文章的平均长度分布如图 4-1：

图 4-1 题目下文章的数量的分布



从图 4-1 中我们可以看出，这些题目下文章数量都大于 50 篇，文章数在 60-70 篇和 90-100 篇这两个区间的题目占了绝大多数。这些题目下文章的平均字数见图 4-2：

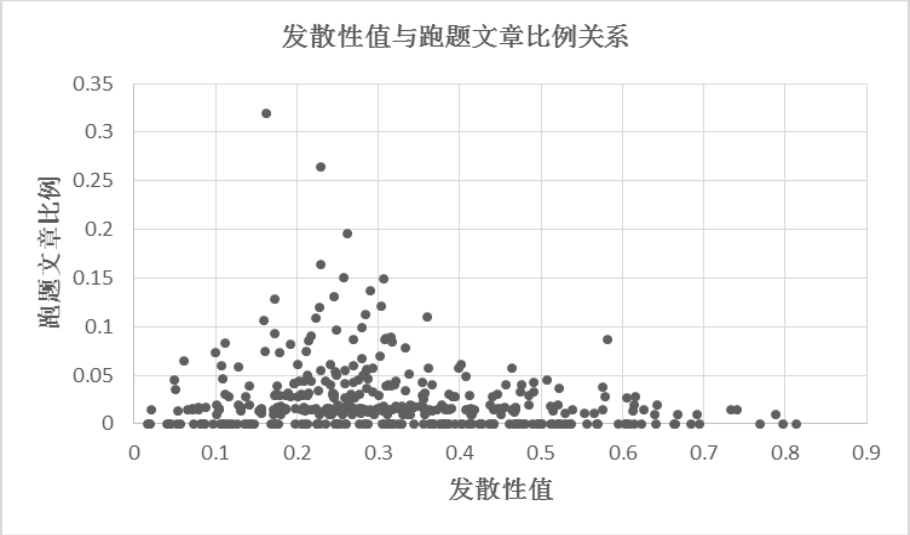
图 4-2 题目下文章平均字数的分布



从图 4-2 中可以看出，这些题目下文章的平均字数集中在 100-200 字的区间。

对于每个题目，我们都进行人工标注，找出其中的跑题文章。为了减少工作量，先用中心向量法找出每个题目下的范文，再计算每篇文章与范文计算相似度。按照评分从低到高进行标注，直到大部分文章都不跑题。不同题目下作文发散性值与跑题文章比例的关系如下图：

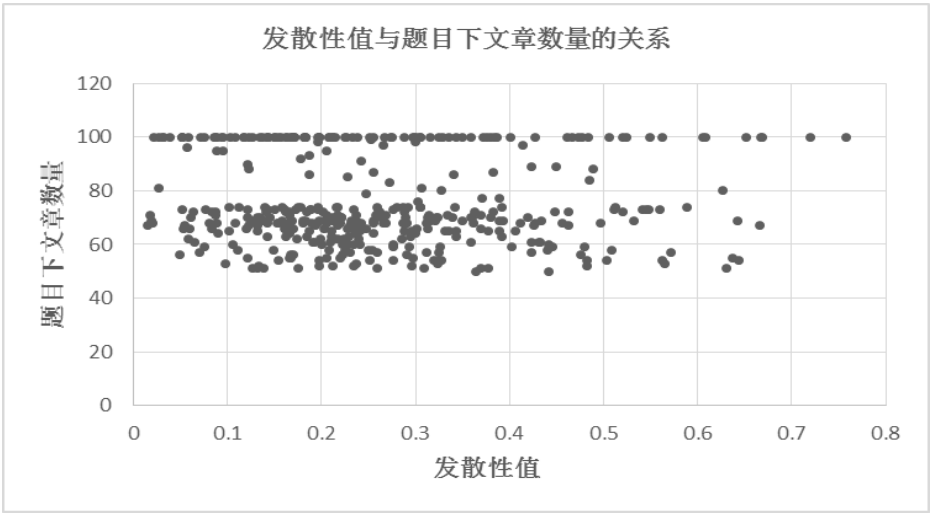
图 4-3 发散性值与跑题文章比例关系



从图 4-3 中可以看出：如果题目下文章的发散性较强或者较弱，即处于上图横轴的两端，这些文章中跑题作文的比例都不高；而发散性中等的（0.2-0.3 左右）题目下，跑题文章占的比例相对较高。这符合我们标注时发现的规律：发散性题目下跑题文章的较少。同一性较高的题目下，比如文章翻译，跑题作文的比例也不高。

另外，我们还统计出了发散性值与题目下文章数量的关系，见下图：

图 4-4 发散性值与题目下文章数量的关系



实验中使用了 Google 开源的 word2vec[16,17,18]工具包[2]。这个工具包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。为此，我们又申请了 3209128 篇学生作文作为 word2vec 的训练语料。同时，这 3209128 篇文章还用来生成词表，以及训练单词的 idf 值。在生成词表的时候，我们过滤掉了出现次数低于 5 次的单词，主要是为了过滤掉拼写错误的单词。

为了学习每个题目下跑题阈值和文章发散度的关系，需要一个训练集。每个训练实例为一个题目下的文章发散度和跑题阈值，发散度用之前所述的方法计算，跑题阈值根据人工标

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

注的结果来选取：对题目下所有文章按照系统得分从低到高排序，选取跑题文章中得分最高的文章和它下一篇不跑题文章的得分的均值作为阈值。例如一个题目下，跑题文章中得分最高的文章分数为 0.1，它下一篇文章为不跑题文章，得分 0.2，那么阈值就等于（0.1 + 0.2）÷ 2 = 0.15。如果一个题目下没有跑题作文，那么阈值就选取最低得分的一半。

4.2 选取范文

由于实验所使用的题目数量较多，很难人工选取每个题目下的范文范文。因为这会耗费大量的时间和人力。为此我们采用了中心向量法自动选取范文。

首先，基于向量空间模型，将所有文章表示成向量。同样，使用 TF-IDF 值作为权重。假设有 M 篇文章，词表中有 n 个词，每篇文章表示成如下向量形式：

$$V(D_1)=[a_{11},a_{12},a_{13}...a_{1n}]$$
$$V(D_2)=[a_{21},a_{22},a_{23}...a_{2n}]$$
$$\cdots\cdots$$
$$V(D_m)=[a_{m1},a_{m2},a_{m3}...a_{mn}]$$

其中，等号左侧 $V(D_m)$ 表示第 m 篇文章的向量形式，右侧是其向量的具体表示，共 n 维，每一维都是相应单词的 TF-IDF 值。我们定义中间向量为所有向量相加后和的均值。使用如下公式计算：

$$V_{\text{中心向量}}=\frac{1}{m}[V(D_1)+V(D_2)+V(D_3)...+V(D_m)] \tag{6}$$

如果把一个文章向量看成向量空间中的一个点，那么中心向量就是这些点的中心。离中心向量的距离最近的文章就可以作为范文。即：

$$D_{\text{范文}}=\arg\min_{D_k}(Sim(D_k,\text{中心向量})) \tag{7}$$

4.3 实验评价

我们利用准确率（Precision）、召回率（Recall）和 F1 值来评价系统。将 400 个题目按照题目分为 10 份，做 10 倍交叉验证。每次取其中的 1 份，共 40 个题目，作为测试集，其余 9 份作为训练集。通过训练集训练出阈值和发散度的回归关系模型。测试时，首先计算出每个题目下的文章发散度，然后根据学习好的回归模型求出阈值，找出系统评分小于阈值的文章，假设有 N 篇，其中 K 个是正确的判断（即和人工判断一致），设这个题目下所有跑题文章数为 M，则：

$$P=\frac{K}{N} \tag{8}$$

$$R=\frac{K}{M} \tag{9}$$

$$F=\frac{2}{\frac{1}{P}+\frac{1}{R}} \tag{10}$$

如果 $M=0,K=0$，说明题目下没有跑题文章，而且预测出结果也是没有跑题文章，那么 $R=1$。每一次测试都计算出测试集的准确率，召回率和 F1 值。最后求 10 次实验结果的平均。

4.4 系统实现

在本文提出的方法中，我们使用 weka 开源工具包[3]学习跑题阈值和文章发散度的线性回归模型参数。

除了本文提出的方法，本次试验还实现了其他两种阈值选取方法用于对比：

- **固定阈值法**，该方法来自于陈志鹏等[1]。我们使用训练集选取固定阈值。和其它方法一样，首先用中心向量法找出每个题目下的范文。再使用词扩展方法计算出每篇文章与范文的相似度，作为系统评分。接着选取固定阈值，我们构造一个预测集用于选取阈值。首先按照系统评分对所有文章排序。我们按照得分从低到高选取文章作为预测集。一开始选取得分最低的文章加入到预测集中，然后选取得分第二低的文章加入……以此类推，得到一个个预测集。我们计算出预测集召回率为 0.1,0.2,0.3...1.0 时的 F1 值，F1 值最大时说明这时候预测集判断的效果最好。取此时预测集中跑题文章得分的最大值作为固定阈值。找到固定阈值后，对测试集中所有文章均使用此阈值进行判断。
- **估计阈值法**。这个方法和本文提出的动态选取阈值的方法大体一致。唯一的不同点是训练时没有通过人工标注来获得每个题目的阈值，而是采用了一种估计的方法判断文章是否跑题。首先在训练集中随机选取 20 题目进行人工标注，得到里面跑题文章的集合。计算出跑题文章所占的百分比，比如 0.01。假设所有题目下跑题文章都占该比例，计算出题目下跑题文章的数量，以此估计出跑题的文章。例如题目下有 100 篇文章，那么估计有 100 × 0.01 = 1 篇文章跑题，即认为系统得分最低的 1 篇文章是跑题作文。用这个方法估计出训练集中每个题目下文章的阈值。然后和动态选取阈值的方法一样，训练出阈值与发散度的关系曲线，使用测试集进行测试。这个方法的优点是省时省力，不需要标注太多题目。

4.5 实验结果

我们首先用测试集中所有文章来进行测试，10 倍交叉验证，取平均值作为最后结果。下表 4-1 是实验结果，所有实验中词扩展的数目为 3，即每个词扩展 3 个词。

表 4-1 实验结果（测试集中所有题目）

	P(准确率)	R(召回率)	F1 值
基于发散度的动态阈值法	0.856	0.867	0.862
固定阈值法	0.863	0.846	0.854
估计阈值法	0.826	0.860	0.843

从结果中我们看到基于发散度的动态阈值法效果最好；固定阈值法效果次之，10 次实验中固定阈值平均在 0.1 附近；效果最不好的是估计阈值法。估计阈值法是动态阈值法的简化版本，比较简单，效果和我们的方法比有明显差距。

测试集有一些题目没有跑题作文,这部分题目占题目总数的 31%。我们针对这个情况做了具体分析。如果考虑测试集中有跑题文章的题目，而不考虑没有跑题文章的题目，实验结果如表 4-2 所示。

表 4-2 实验结果（只考虑有跑题文章的题目）

	P(准确率)	R(召回率)	F1 值
基于发散度的动态阈值法	0.919	0.776	0.842
固定阈值法	0.924	0.726	0.814
估计阈值法	0.895	0.764	0.824

从表 4-2 中可以看出，只考虑有跑题文章的题目时，选取动态阈值的方法效果要比选取固定阈值的方法好。基于发散度的动态阈值法比固定阈值法高出 3 个百分点，效果最好。固

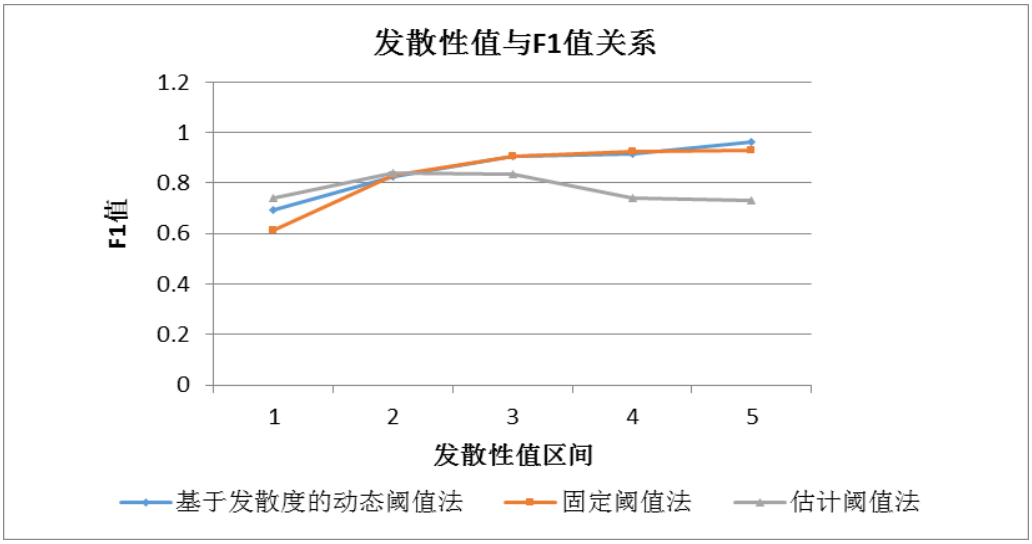
[3] http://www.cs.waikato.ac.nz/ml/weka/

定阈值法和估计阈值法效果差不多。固定阈值的方法准确率较高，估计阈值法召回率较高。

结合表 4-1 和表 4-2 还可以看出，固定阈值方法的变化幅度较大，F1 值降低了 4 个百分点；而选取动态阈值的方法变化却不是很大，这说明动态选取阈值的方法有着较好的稳定性。在判断有跑题文章的题目时，动态选取阈值的方法性能要明显优于固定选取阈值的方法。

最后，我们对实验结果做进一步分析，研究题目发散性和 F1 值之间的关系。我们将所有题目按照文章的发散性值由低到高排序，分为 5 份，每份 80 个题目，第 1 份到第 5 份的平均发散性值依次增高。在发散性最强的 1 区间中，有 31 个题目没有跑题文章，占区间总体的 38%。计算每份的平均 F1 值。结果如下图：

图 4-5 发散性值与 F1 值关系



从图中可以看出，在面对发散性较强的题目时选取动态阈值的方法比固定阈值法的性能好。随着题目发散性逐渐变弱，估计阈值法的 F1 值明显下降，其他两种方法的 F1 值都不断上升。总体来看，对于发散性较强和较弱的两种题目，基于发散度动态选取阈值的方法要好于固定阈值的方法，而对于发散性一般的题目，两种方法差距并不明显。

综上所述，基于发散度选取动态阈值的方法性能最好。处理有跑题作文的题目时，该方法明显好于固定阈值的方法。面对发散性较强的题目时，该方法性能也优于固定阈值的方法。

5 总结和展望

本文构造了一个跑题检测系统，相对于传统选取固定阈值的方法，该方法的创新之处是基于文档发散度动态地选取阈值，从而判断文章是否跑题。经过实验比较，该方法在面对有跑题文章的题目时，尤其是发散性较强的题目时，性能明显优于固定选取阈值的方法。作文跑题检测还有许多研究空间，比如如何更加准确地对发散度较高的题目进行检测等，还有许多方向可以进一步研究。

参考文献

[1] 陈志鹏，陈文亮，朱慕华. 利用词的分布式表示改进作文跑题检测[J]. 中文信息学报，2015, 29(5):178-184.

[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] Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, Introduction to Information Retrieval[M]. Cambridge University Press,2008:83-84.
[8] Miller G. Wordnet：An On-line Lexical Database[J]. International Journal of Lexicography, 1990, 3(4): 235-244.
[9] 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. 2004:89-97.
[10] 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.
[11] Page,E.B.Project Essay Grade:PEG[A].In Shermis,M.D.&Burstein,J.C.(eds.). Automated Essay Score : A Cross-Disciplinary Perspective[C].NJ:Lawrence Erlbaum Associates,2003:43-54.
[12] Landauer,T.K., Laham,D.& Foltz, P.W. Automated essay scoring and annotation of essays with the Intelligent Essay Assessor[A]. In Shermis, M.D.&Burstein,J.C.(eds.). Automated Essay Scoring：A Cross-Disciplinary Perspective[C].NJ：Lawrence Erlbaum Associates,2003:87-112.
[13] Burstein, J. The E-rater Scoring Engine: Automated essay scoring with natural language processing[A]. In Shermis, M.D.&Burstein, J.C.(eds.). Automated Essay Scoring : A Cross-Disciplinary Perspective[C]. NJ : Lawrence Erlbaum Associates. 2003 : 113-121.
[14] 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.
[15] 葛诗利,陈潇潇.文本聚类在大学英语作文自动评分中应用[J].计算机工程与应用,2009,45(6):114-121.
[16] Tomas Mikolov, Kai Chen, Greg Corrado, 等.Efficient Estimation of Word Representations in Vector Space[C]//In Proceedings of Workshop at ICLR, 2013.
[17] Tomas Mikolov, Ilya Sutskever, Kai Chen, 等. Distributed Representations of Words and Phrases and their Compositionality[C]//In Proceedings of NIPS, 2013.
[18] 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。地址：江苏省苏州市 十梓街 1 号 苏州大学 计算机科学与技术学院 理工楼 406 室 邮编：215006，电话：15995763057。Email: chenzhipeng341@163.com。

陈文亮（1977—），男，博士，通讯作者，主要研究领域为自然语言处理。地址：江苏省苏州市 十梓街 1 号 苏州大学 计算机科学与技术学院 理工楼 311 室 邮编：215006，电话：15618055040。Email:wlchen@suda.edu.cn。