

# TREC 自适应信息过滤中的目标优化技术研究<sup>1</sup>

许洪波 王斌 程学旗 白硕

中国科学院计算技术研究所 软件研究室 北京 100080

E-mail: [hbXu@software.ict.ac.cn](mailto:hbXu@software.ict.ac.cn)

**摘要:** TREC 是信息检索领域最重要的国际评测会议, 自适应信息过滤是 TREC 的关键任务之一。自适应阈值调整中评测指标的优化是自适应过滤任务中非常重要的研究方向。我们借鉴优化学习的思想, 以 TREC 的评测指标为目标函数, 提出了针对全程和局部效用指标进行优化的阈值调整方法, 并对两种方法的优缺点进行了比较分析。实验表明, 我们提出的局部效用指标优化方法是非常有效的, 在 TREC-10 和 TREC-11 过滤任务评测中取得了成功。

**关键词:** TREC, 自适应过滤, 阈值调整, 全程效用指标优化, 局部效用指标优化

## Target-oriented Optimization in TREC Adaptive Information Filtering

Xu Hongbo Wang Bin Cheng Xueqi Bai Shuo

Software Division, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080

E-mail: [hbXu@software.ict.ac.cn](mailto:hbXu@software.ict.ac.cn)

**Abstract:** TREC is the most important conference in IR field. Adaptive Filtering track is one of the most important tasks in TREC. Evaluation measure optimization with threshold tuning is one of the most important research areas in Adaptive Filtering. Basing on optimization learning, we take TREC evaluation measure as our target function, and then propose holistic and local target optimization methods in threshold tuning. Experiments show that our local target optimization method is very effective. We've made a great success in TREC-10 and TREC-11 Adaptive Filtering track.

**Keywords:** TREC, Adaptive Filtering, threshold tuning, holistic target optimization, local target optimization

### 1 TREC 信息过滤任务介绍

TREC (Text REtrieval Conference)会议是国际上信息检索领域最重要的会议之一。它由美国 NIST(National Institute of Standard and Technology, 美国国家技术标准局)和 DARPA(Defense Advanced Research Projects Agency, 美国国防高级研究计划局)组织举办, 每年召开一次, 旨在交流该领域内相关的技术并对这些技术在统一的平台上进行评测。自 1992 年以来, TREC 会议已经召开了 11 届, 为文本检索技术的发展起到了十分重要的作用。作为该领域最重要的会议, TREC 吸引了 IBM、Microsoft、AT&T、Sun、Apple、Oracle、MIT、CMU、Cambridge U 等国际重要的研究机构。最近几年, 中国大陆的复旦大学、微软亚洲研究院、中科院计算所、哈工大、清华大学等也开始参加这一重要的国际会议, 并且取得了不俗的成绩。

---

<sup>1</sup>本文研究受国家 973 课题(G1998030413, G1998030510)及中科院计算所领域前沿青年基金(20016280-9 和 20026180-24)资助

信息过滤是 TREC 中最主要的任务之一。它研究的目的是针对用户相对固定的需求，顺序扫描文档流中的每个文档，按照某种准则过滤出满足用户需求的文档推送给用户，并根据用户的反馈逐步调整用户的需求表示、过滤标准和策略，从而使得推送的文档更加符合用户的真实需求。

在 TREC 过滤任务中，用户需求用主题(Topic)来描述。实际系统中，用户需求被量化为用户兴趣(Profile)，后者是实际使用的过滤依据。从本质上来说，用户兴趣是用户需求的形式化表示，二者要尽可能等价才能保证系统推送的文档就是用户真正需要的文档。

TREC 过滤任务的主要子任务是自适应过滤任务(Adaptive Filtering)。它是 TREC 信息过滤研究的中心内容。在自适应过滤任务中，每个主题只给出二到三个正例供训练用，系统要利用这有限的已知信息构造出初始的用户兴趣，然后根据过滤结果调整 Profile，使之尽量趋近真正的用户需求。

在 TREC 自适应过滤任务中，如果一篇被扫描的文档能够满足用户的要求，则把它检出并推送给用户；否则该文档就被抛弃。用户对检出的文档进行相关性确认，从而指导 Profile 的调整。TREC 的自适应任务很逼真地模拟了实际的过滤系统的运行过程。实现时，我们可以首先设定一个初始阈值  $T$ ，即过滤的检出门槛。每扫描到一篇文档，则计算向量化的文档跟系统构造的用户兴趣 Profile 之间的相似度  $\text{sim}(\text{doc}, P)$ ，若  $\text{sim} > T$ ，则该文档被检出并推送给用户；否则舍弃之。显然，阈值  $T$  的作用很重要，过低则会检出大量不相关文档，过高则可能漏掉许多相关文档，两种极端都会损害过滤系统的性能。被检出的文档只是过滤系统认为的“相关”文档，其真实相关性（即是否真正满足用户需求）要在用户浏览或者确认后得知。系统根据用户给出的对于文档相关性的肯定或者否定的反馈意见，调整用户兴趣使之更加接近用户的真实需求，同时可以根据检出文档中相关和不相关文档的比例等因素来决定是否应升高或者降低检出阈值以提高系统的过滤性能。

自适应过滤任务实现的关键技术主要包括：（1）用户兴趣的初始化和初始检出阈值的确定；（2）用户兴趣的反馈学习；（3）阈值调整和评测指标优化。对于用户兴趣的初始化和初始阈值的确定以及用户兴趣的反馈学习方面，我们在论文<sup>[1]</sup>中已有介绍，本文不予详细讨论。自适应阈值调整是优化评测指标、提高过滤系统性能的重要手段。TREC 鼓励参加者提交针对评测指标进行优化的结果并指明优化的指标和所用方法等。由于指标函数的优化是一个包括优化时机、优化策略、优化幅度等因素的复杂的非线性系统，获得良好的优化效果比较困难，因而成为过滤任务研究的重点和难点之一。本文主要介绍我们在自适应阈值调整中优化评测指标的方法和技术<sup>[1][2]</sup>。

## 2 TREC 自适应过滤任务的评测指标

TREC 过滤任务使用四种评测指标：效用指标(Utility)，F-beta 指标，准确率和召回率。其中以效用指标和 F-beta 指标为主，而效用指标更重要些，是 TREC 评测排名的主要依据。

### 2.1 效用指标 Utility

线性效用指标通用的计算公式为： $\text{Linear utility} = A \cdot R^+ + B \cdot N^+ + C \cdot R^- + D \cdot N^-$ 。其中变量  $R^+$  表示过滤时检出的相关文档数， $R^-$  表示未检出的相关文档数， $N^+$  表示检出的不相关文档数， $N^-$  表示未检出的不相关文档数，如表 1 所示。

	相关的文档数	不相关的文档数
检出的文档数	$R^+ (A)$	$N^+ (B)$
未检出的文档数	$R^- (C)$	$N^- (D)$

表 1 过滤系统各种文档含义

效用指标值越大，表示过滤系统在相应主题上表现的性能越好。具体到每一年，TREC

实际使用的效用指标计算公式会略有不同，一般是针对以往结果评测中出现的问题进行适当调整，其目的就是为了使得评测指标更合理，更有说服力，能够更真实的反映过滤系统对于用户需求的满足程度。在 TREC-10 和 TREC-11 中，效用指标公式为： $T10U(T11U)=2*R^+-N^+$ 。 $T10SU$ 、 $T11SU$  分别是对  $T10U$  和  $T11U$  进行归一化的结果，是 TREC 最终评测系统性能时使用的效用指标。TREC-10 和 TREC-11 效用指标的不同之处是其归一化公式有所不同。

## 2.2 F-beta 指标

F-beta 指标最初是 van Rijsbergen 提出来的，是召回率和准确率的函数，通过一个自由参数 beta 来决定召回率和准确率在评测中所占的权重。在 TREC-10 和 TREC-11 中设定  $\beta=0.5$ ，表示在 F-beta 指标的计算中更强调准确率。 $\beta=0.5$  时 F-beta 指标的计算公式如下：

$$T10F(T11F) = \begin{cases} 0, & \text{if } R^+ = N^+ = 0; \\ \frac{1.25 * R^+}{0.25 * R^+ + N^+ + 1.25 * R^+}, & \text{otherwise.} \end{cases} \quad (1)$$

$$\text{MeanT10SF} = \text{Mean } \Sigma (T10F) \quad (2)$$

即整个过滤系统的 F-beta 值是所有主题的 F-beta 值的平均值。

## 3 自适应过滤中的全程效用指标优化方法

由于 TREC 过滤任务的评测指标实际上是模拟用户对于过滤结果的主观评判，评测指标值越大表明用户对于系统的过滤结果越满意。因而针对评测指标进行的优化将使得系统性能更优，过滤结果更能满足用户的需求，反过来又可以印证 TREC 过滤任务的评测指标是否合理，即是否能够比较真实的反映用户要求的满意程度，从而使得整个评测变得更加有意义。

在 TREC-10 的过滤任务中，我们采用的是比较简单直观的全程效用指标优化方法<sup>[2]</sup>。虽然在最终的评测中取得了不错的结果<sup>[4]</sup>，但是也暴露出很明显的问题，比如出现了很多主题的一篇文档都没有检出的情况，对这些主题的过滤失败极大地损害了整个过滤系统的性能。后来的分析表明，这主要是由于过滤系统对于检出阈值调整不力导致的结果。

全程效用指标优化策略把全程的效用指标作为目标函数，根据其计算公式  $T10U=2*R^+-N^+$  采取优化措施。因为  $N^+=N-R^+$ ，所以  $T10U=2*R^+-N^+=3*R^+-N$ 。因此全程效用指标优化的目标函数设为： $f_h=3*R^+-N$ ，其中， $R^+$  是迄今为止检出的相关文档总数， $N^+$  是迄今为止检出的不相关文档总数， $N$  是迄今为止检出的文档总数。

全程效用指标优化的方法是：每完成一个反馈区间就计算一次当前的目标函数  $f_h$  值，根据  $f_h$  取值的不同情况决定升高还是降低或者保持文档检出阈值  $T$ ，从而实现对于过滤检出文档的控制。全程效用指标优化的阈值调整公式如下：

$$T = \begin{cases} T + \Delta t, & \text{if } f_h < 0; \\ T - \Delta t, & \text{if } f_h \geq 0 \text{ and } \Delta N < \Delta n; \\ \text{保持不变,} & \text{otherwise.} \end{cases} \quad (3)$$

当  $f_h \geq 0$  而  $\Delta N < \Delta n$  时，虽然当前系统对于效用指标的贡献为正（增益），但检出的文档数过少（小于某个事先定义的下限  $\Delta n$ ），这说明检出阈值已经处在过高的水平上，因此要适当降低之。若  $f_h \geq 0$  而  $\Delta N \geq \Delta n$ ，说明系统对于效用指标的贡献为正并且检出的文档数适量，当前的检出阈值比较合理，因而不必调整。当  $f_h < 0$  时，当前系统对于效用指标的贡献为负（损失），这说明系统在迄今为止的时间内已经检出了太多的不相关文档，而导致这种情况的原因必然是检出阈值降到了过低的水平上，因此我们要适当提高阈值，减少检出的不相关文档数。这样做是基于一个合理的假设：根据给定的用户需求和有限正例构造出来的用户兴趣（即过滤中心），与用户的真正需求是一致的，二者不存在较大的偏差，更不会出现完全对立的情况。

基于这个前提，系统在过滤时总体上会做出正确的判断，每一次过滤的结果都应该是多数相关文档的相似度大于不相关文档的相似度，如果按照相似度排序的话，则应该是多数相关文档排在不相关文档之前。因此，提高检出文档的阈值时，首先过滤掉的应该是不相关文档。这样，相关文档在检出文档中的比例就会提高，相应的，不相关文档的比例自然会下降，从而使得效用指标向正增益的方向发展，这就达到了优化全程效用指标的目的。

全程效用指标优化策略的缺点是多次反馈后由于检出文档的累积效应会导致过滤系统逐渐丧失对阈值进行有效调整的能力。在过滤的初始阶段， $R^+$ 和 $N^+$ 都很小，目标函数 $f_0$ 对于 $R^+$ 和 $N^+$ 的增加比较敏感，因而过滤效果相对理想。但是，随着检出文档数的增多， $R^+$ 和 $N^+$ 由于累积效应越来越大，新反馈区间内增加的检出文档数对于 $f_0$ 的影响就会越来越小。当 $R^+$ 和 $N^+$ 增大到远远超过一个反馈区间的检出文档增量的时候，新反馈区间内增加的检出文档数对于 $f_0$ 的影响力将变得微乎其微，这使得T10U对于新的反馈越来越不敏感，即使是来自一个出现较大偏差的区间的反馈，从而失去了优化效用指标的意义。这时，检出阈值往往会徘徊在某个固定位置附近，似乎过滤系统已经进入成熟阶段，但这往往是一个假象，会误导过滤系统在低水平徘徊。当某个新区间出现较大过滤偏差时本应继续大力调整阈值，但累积效应使得 $f_0$ 很难捕捉到这种反馈信息，一般要经过若干个连续区间的偏差之后才能发现并进行调整，这种滞后效应将会损害系统的最终性能。

#### 4 基于局部效用指标优化策略的自适应过滤技术

鉴于全程效用指标优化时由于累积效应导致目标函数对新增检出文档数不敏感、优化滞后的问题，我们提出了针对局部指标函数的优化策略。

##### 4.1 针对局部指标函数的优化方法

局部效用指标优化的基本思想仍然是根据效用指标 $T10U=2^*R^+-N^+$ 的计算结果来调整检出阈值 $T$ ，但这里的 $R^+$ 和 $N^+$ 不再是迄今为止检出的相关和不相关文档总数，而仅指一个过滤区间内新检出的相关和不相关文档数。局部效用指标优化的目标函数选择为： $f_1=2^*R^+-N^+$

局部效用指标优化方法在调整检出阈值时充分考虑了准确率和召回率对目标函数的影响，优化过程中尽量使二者保持平衡，具体的阈值调整公式是：

$$T = \begin{cases} T - \Delta t, & \text{if } R^+ + N^+ = 0; \\ T + 2\Delta t, & \text{if } f_1 \leq -R^+; \\ T + \Delta t, & \text{if } f_1 > -R^+ \text{ and } f_1 \leq \delta; \\ T - \Delta t, & \text{if } f_1 > \delta \text{ and } \Delta N < \Delta n; \\ \text{保持不变, } & \text{otherwise.} \end{cases} \quad (4)$$

其中 $R^+$ 即目标函数中的 $R^+$ ， $\delta$ 为某个很小的正值。 $\Delta t$ 为阈值调整幅度，它表示优化的力度， $\Delta t$ 大则优化力度大反之则力度小。局部效用指标优化过程为：

(1) 当 $R^++N^+=0$ ，即该过滤区间检出的总文档数 $N$ 为0时，检出的相关和不相关文档数也都是0，因而目标函数 $f_1=0$ ，这说明检出阈值水平过高，虽然效用指标没有损失，但召回率可能受损，因此我们适当降低阈值使得有文档可被检出，从而增加相关文档检出的可能性；

(2) 当 $f_1$ 比0增益小很多时，说明系统在当前区间内检出了过多的不相关文档，这时应大幅提高检出阈值，以期迅速纠正错误，避免后续区间继续造成严重损失；

(3) 当 $-R^+ < f_1 \leq -\delta$ 时，当前区间对于系统总效用指标的贡献为负，但造成的损失不大，这时可以适当提高检出阈值以使后续区间能够在一定程度上弥补上一区间的损失。而当 $-\delta < f_1 \leq \delta$ 时，虽然当前区间基本未给总效用指标带来增益或者造成损失，但我们本着“宁缺毋滥”的原则，仍然适当提升检出阈值，这跟TREC评测时更强调准确率的做法是吻合的；

(4) 当 $f_1 > \delta$ 而 $\Delta N < \Delta n$ 时，虽然当前区间对于效用指标的贡献为正，但检出的文档数过

少（小于某个事先定义的下限  $\Delta n$ ），这表明检出阈值有过高的危险，因此我们适当降低之以利于提高系统的召回率。

(5) 若  $f_i > \delta$  而且  $\Delta N \geq \Delta n$ ，表示当前区间对于效用指标的贡献为正并且检出的文档数适量，这说明当前的检出阈值比较合理，因而不必进行调整。

必须注意的是，阈值升高或者降低都不能无限制的进行下去，阈值过高或过低都会由于失真反而损害系统的过滤性能。在实际过滤系统中我们设定检出阈值的上下限。

可以看出，局部效用指标优化策略是一种贪心策略，它针对各个区间的用户反馈实时调整检出阈值，通过追求各个反馈区间的局部最优来获得全局的最优，体现了一定的功利性，但能够避免过滤区间之间的连续损失，增强区间之间得失的互补性，因而有利于达到全局的最优性。局部效用指标优化的另一个优点是消除了累积效应，根据每个反馈区间的结果独立调整检出阈值，这使得检出阈值可以处于比较剧烈的震荡状态；又由于这种震荡只是在局部进行的，即使震荡时存在偏差，波及的范围也仅限于少数后续区间，副作用很小，而检出阈值在剧烈震荡的状态下可以快速找到准确位置。

在过滤后期，系统的反馈学习能力接近饱和，当前的用户兴趣基本达到或者接近了准确的用户需求，过滤进入成熟阶段。但这是相对于初始阶段来说的，并没有一个确定的界限，此时的过滤结果仍然有可能出现较大波动，检出阈值也要进行相应调整。全程效用指标优化方法对这种情况较难适应。因为文档的累积效应使它不容易检测到出现在过滤后期的较大的结果波动，即使检测到，也要花费多个后续区间的代价才能完全纠正偏差，而系统在这些区间上的过滤效果就会被牺牲掉。局部效用指标优化策略则可以很好的解决这个问题。无论在过滤初期还是后期，由于不累加检出文档，一旦过滤结果出现较大偏差，会立刻在当前区间的目标函数  $f_i$  上表现出来，从而使系统能够及时捕捉到过滤结果的异常，调整检出阈值。

#### 4.2 全程和局部效用指标优化结果的比较

图1是我们的过滤系统在 TREC-10 测试集上分别采用全程和局部目标函数优化制导的阈值调整方法时效用指标  $T10SU$  随统计区间变化的对比曲线。黑色折线对应的过滤结果为 ICTAdaFT10Fa( $T10SU=0.206$ ,  $T10F=0.387$ )，采用全程目标函数优化方法；红色折线对应的过滤结果为 LocalOptimizF( $T10SU=0.265$ ,  $T10F=0.397$ )，采用局部目标函数优化方法。详见第5节结果分析。在图1的左部，横轴一个刻度代表每6个反馈区间(每扫描10000篇文档为一个反馈期间)组成的一个统计区间，这样总共78万多篇文档划分为13个统计区间。纵轴表示过滤系统的评测指标分值。

从图中我们可以看出，采用局部目标函数优化方法时效用指标的波动变化比全程目标函数优化更剧烈，并且震荡幅度也更大。例如，局部目标函数优化的效用指标最大增幅为0.06（区间1-2），而全程目标函数优化的效用指标最大增幅仅为0.03（同样为区间1-2）。从第6区间到第7区间，局部目标函数优化的效用指标下降了0.005，随后则不断回升，总共上升了0.029(0.010+0.015+0.004，区间8-10)。而与此同时，全程目标函数优化的效用指标在下降了0.009后却只回升了0.022(0.012+0.010，区间8-9)。特别的，局部目标函数优化的效用指标在区间10-11出现了0.027的最大下降后又上升了0.023(0.012+0.011，区间12-13)，几乎完全恢复到了先前的水平。与之形成鲜明对比的是，同样在这期间，全程目标函数优化的效用指标剧降了0.022却并未恢复，反而又继续下降了0.002。很明显，全程目标函数优化未能检测到过滤后期效用指标的大幅下降以至于它没有采取相应的步骤来弥补系统性能的损失。如果我们绘制出全部78个反馈区间的曲线图，局部目标函数优化方法的这种优势就会更加明显。

使用全程目标函数优化方法，效用指标值从初始的0.290增加到最终的0.354，增幅仅为22%(0.064)。而使用局部目标函数优化方法时，效用指标值从0.317最终增加到了0.429，增幅高达35%(0.112)，是全程目标函数优化增幅的1.5倍多。同时，局部目标函数优化的效用

指标最终达到了比全程目标函数优化高得多的水平。两种方法的巨大差距说明局部目标函数优化方法比全程目标函数优化方法具有更好的学习能力，其主要原因是前者有效克服了文档累积效应的影响。图 1 的右部是两种优化方法的 F-beta 指标 T10F 随统计区间变化的对比曲线。从中同样可以得出局部目标函数优化优于全程目标函数优化的结论。

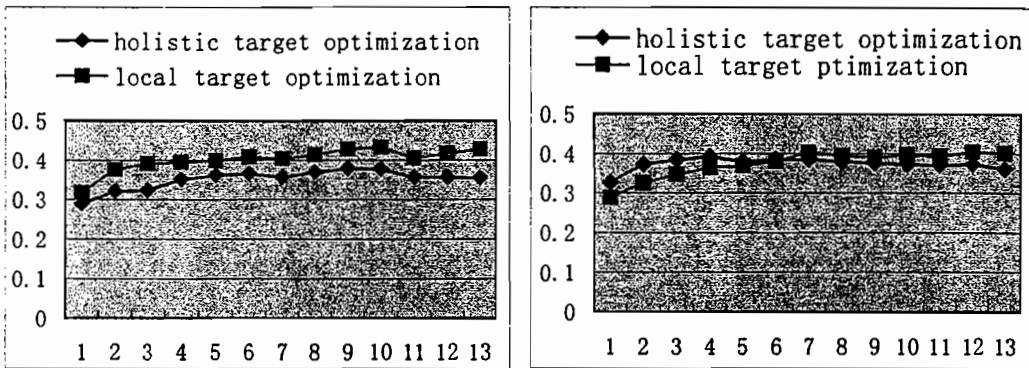


图 1 两种方法的过滤系统在各个统计区间的对比(左 T10U 右 T10F)

## 5 实验结果比较和分析

我们在 TREC-10 自适应过滤任务中采用的是针对全程效用指标函数的优化策略，取得了总体评测第四名的较好成绩，但跟前两名的差距很大。表 2 是我们提交的过滤结果<sup>[2]</sup>和 TREC-10 最终评测结果的前两名<sup>[4]</sup>。

Run ID	MeanT10SU	MeanT10F	Precision	Recall	Retrieved	Relevant
ICTAdaFT10Ua	0.204	0.368	0.421	0.325	573495	338958
ICTAdaFT10Ub	0.205	0.366	0.398	0.356	527761	326201
ICTAdaFT10Uc	<b>0.207</b>	<b>0.354</b>	<b>0.390</b>	<b>0.330</b>	<b>605659</b>	<b>356909</b>
ICTAdaFT10Fa	<b>0.206</b>	<b>0.387</b>	<b>0.427</b>	<b>0.364</b>	<b>595586</b>	<b>360399</b>
第一名	<b>0.291</b>	<b>0.518</b>	<b>0.538</b>	<b>0.496</b>	<b>811181</b>	<b>460769</b>
第二名	<b>0.263</b>	<b>0.499</b>	<b>0.657</b>	<b>0.345</b>	<b>263374</b>	<b>214248</b>

表 2 TREC-10 自适应过滤任务的评测结果

在 TREC-10 过滤任务的最终评测结果发布以后，我们总结了结果中存在的问题，通过分析原因，我们提出了以局部效用指标配合 F-beta 指标为目标函数的优化策略，并进行了进一步试验。实验结果跟我们预期的效果完全吻合，主要评测指标 T10SU 的得分由 0.207 上升到 0.270，提升幅度高达 30%多，超过了 TREC-10 第二名的评测结果，相当令人满意。这充分证明局部效用指标优化策略的引入是成功的。表 3 是对 TREC-10 任务采用局部效用指标优化策略的过滤结果，其中 LocalOptimizU 着重考虑对效用指标 T10U 进行优化（加大阈值提升幅度  $\Delta t$ ），LocalOptimizF 强调对 F-beta 指标 T10F 的优化（多种策略降低阈值），而 LocalOptimizR 则着重对召回率进行优化（加大阈值下降幅度  $\Delta t$ ）。

Run ID	MeanT10SU	MeanT10F	Precision	Recall	Retrieved	Relevant
LocalOptimizU	<b>0.270</b>	<b>0.371</b>	<b>0.438</b>	<b>0.332</b>	<b>544727</b>	<b>377209</b>
LocalOptimizF	<b>0.265</b>	<b>0.397</b>	<b>0.482</b>	<b>0.336</b>	<b>532813</b>	<b>368252</b>
LocalOptimizR	0.263	0.396	0.477	0.338	536132	369523

表 3 基于局部效用指标优化策略的自适应过滤结果

2002年,我们继续参加了TREC-11的过滤任务评测。相对于TREC-10的过滤任务来说,TREC-11过滤任务难度增大,主要是训练集和测试集中只有少量文档的相关性是已知的,绝大多数文档都是未判定文档,因而大大增加了反馈学习的难度。在这种标准答案不确定的情况下,要保证反馈学习的有效性和可靠性是一个相当困难的问题。我们在TREC-10优化实验的基础上进一步完善了针对局部效用指标的优化策略,同时针对今年的过滤任务采取了相应对策,包括反馈区间的调整、用户兴趣的学习等等。最终我们提交的结果在TREC-11自适应过滤任务评测中取得了两项指标(T11SU和T11F)均排名第一的优异成绩<sup>[3]</sup>。这证明局部指标函数优化策略在不同学习条件下具有良好的适应能力。表4是我们在TREC-11自适应过滤任务中的评测结果<sup>[1]</sup>以及跟第二、三名最好结果的比较<sup>[3]</sup>。

Run ID	MeanT11SU	MeanT11F	Precision	Recall	Retrieved	Relevant
ICTAdaFT11Ua	<b>0.405</b>	<b>0.244</b>	<b>0.310</b>	<b>0.197</b>	<b>4166</b>	<b>2532</b>
ICTAdaFT11Ub	<b>0.403</b>	<b>0.245</b>	<b>0.311</b>	<b>0.199</b>	<b>4261</b>	<b>2543</b>
ICTAdaFT11Uc	<b>0.403</b>	<b>0.242</b>	<b>0.307</b>	<b>0.200</b>	<b>4214</b>	<b>2546</b>
第二名	0.390	0.214	0.334	0.140	2876	2010
第三名	0.369	0.222	0.279	0.157	3077	1817

表4 TREC-11自适应过滤任务的评测结果

## 6 结论

TREC自适应过滤任务有别于一般的信息过滤任务,它更强调过滤系统的自适应学习能力。在TREC-10过滤任务中,我们在调整阈值时针对全程效用指标函数进行优化<sup>[2]</sup>。在TREC-11过滤任务中,我们分析了全程效用指标优化方法的优缺点,借鉴优化学习的思想,提出了以局部效用指标为目标函数的优化算法制导阈值调整,通过追求各个反馈区间的局部最优来达到全局的最优,同时尽量避免出现检出文档数为0的情况。在TREC-10和TREC-11测试集上的实验表明,我们提出的局部效用指标优化方法是非常有效的。尤为可喜的是,我们在TREC-11自适应过滤任务的最终评测中取得了两项指标均排名第一的好成绩。

## 参 考 文 献

- [1] Hongbo Xu, Zhifeng Yang, Bin Wang, Bin Liu, Jun Cheng, Yue Liu, Zhe Yang, Xueqi Cheng, Shuo Bai. TREC-11 Experiments at CAS-ICT: Filtering and Web, In *The Tenth Text REtrieval Conference (TREC 11)*, 2002.
- [2] Bin Wang, Hongbo Xu, Z. Yang, Y. Liu, X. Cheng, D. Bu, S. Bai, TREC-10 Experiments at CAS-ICT: Filtering, Web and QA, In *The Tenth Text REtrieval Conference (TREC 10)*, page 109, 2001.
- [3] S. Robertson, I. Soboroff, The TREC 2002 Filtering Track Report, In *The Eleventh Text REtrieval Conference (TREC 11)*, page 208, 2002.
- [4] S. Robertson, I. Soboroff, The TREC 2001 Filtering Track Report, In *The Tenth Text REtrieval Conference (TREC 10)*, page 26, 2001.
- [5] Yi Zhang, James P. Callan, Maximum Likelihood Estimation for Filtering Thresholds, SIGIR 2001, page 294-302, 2001.
- [6] Y. Zhang, J. Callan, The Bias Problem and Language Models in Adaptive Filtering, In *The Tenth Text REtrieval Conference (TREC 10)*, page 78, 2001.
- [7] T. Ault, Y. Yang, kNN, Rocchio and Metrics for Information Filtering at TREC-10, In *The Tenth Text REtrieval Conference (TREC 10)*, page 84, 2001.
- [8] S. Alpha, P. Dixon, C. Liao, C. Yang, Oracle at TREC 10: Filtering and Question-Answering, In *The Tenth Text REtrieval Conference (TREC 10)*, page 423, 2001.
- [9] Arampatzis, Unbiased S-D Threshold Optimization, Initial Query Degradation, Decay, and Incrementality, for Adaptive Document Filtering, page 596, 2001