## 学习排序算法（ZT）

通过上述的文本语义相似度计算之后，对于每一个查询都可以排序生成一个排序列表，即候选列表，设置相应的阈值，即可产生相应的需求跟踪关系。但是，此时仅仅使用了文本语义相似度这一个特征，在数据集中还有一些没有使用的特征。综合使用其他的特征可以提升结果的准确率，因此本文使用学习排序算法来进一步对已有的排序列表进行处理。以下将从两个方面描述学习排序算法：特征选择和排序算法。

### 特征选择

本文选用了五个特征作为学习排序模型的输入，五个特征可以被分为两组：依赖查询的特征和不依赖查询的特征，如表3所示。前三个特征为依赖查询的特征，这几个特征强调查询与候选结果之间的关联关系；后两个为不依赖查询的特征，更强调候选结果文本自身的特点。

（1）语义相似度，通过提出的语义相似度算法WQI计算；

（2）广义Jaccard系数，通过计算文档向量的关系表示两个文档上下文相似度，在本文应用中即为查询语句和一个候选文档之间的上下文相似度，如公式(3.10)所示。

(3.10)

其中，向量和向量均为文档向量，此处文档向量使用词向量的平均值表示。和分别表示向量和向量的数值大小。

（3）IDF之和，在候选结果文本上中计算查询中的词或短语的IDF值，并求和，用来表示当前查询在某一候选结果上的重要程度，如公式(3.11)所示。

(3.11)

其中为查询中的单词或短语，表示在查询结果中的idf值。

（4）关键词数量，表示查询候选结果中关键词的数量，关键词的界定使用TFIDF的值，计算过程与加权策略中介绍的过程相同。

（5）文本长度，表示查询候选结果的长度，在一定程度上体现该结果的有效程度。

表3 学习排序算法选择的特征

|  |  |  |
| --- | --- | --- |
| **特征** | **描述** | **是否依赖查询** |
| 语义相似度 | 使用WQI方法计算的文本语义相似度 | 是 |
| 广义Jaccard系数 | 文本的上下文相似度，文本向量由词向量取平均值表示 | 是 |
| IDF之和 | 候选结果文本上的当前查询中所包含的词或短语的IDF值之和 | 是 |
| 关键字数量 | 候选结果中关键字数量，通过计算TFIDF然后排序和取阈值得到 | 否 |
| 文本长度 | 候选结果文本的长度 | 否 |

### 排序算法

正如第二章中所提到了，单文档方法仅考虑了单个文档与查询的绝对相关度，忽略了文档间的顺序关系；文档对方法考虑了任意两个文档之间的相对前后关系，相比单文档方法的效果更好；文档列表方法需要考虑每次对查询候选结果列表，当文档数量较大时，需要考虑的数量较大，相对而言没有文档对方法的效率高。综合以上原因，Tr-WELR方法模型中使用了文档对方法，并改进了当前成熟的IR SVM算法。

IR SVM的基本思想和文档对方法处理思想一致，都是将排序问题转化为分类问题。IR SVM模型将两个样本和表示成一个训练样本，然后使用SVM模型训练和预测分类。由于IR SVM是对Ranking SVM的改进，下面将首先介绍Ranking SVM，然后再详述IR SVM对Ranking SVM的改进，最后提出在需求跟踪任务中对IR SVM的改进。

#### Ranking SVM

Ranking SVM算法的思想是将排序问题转化为pairwise的分类问题，然后使用SVM分类模型进行学习并求解[43]。在排序应用中，假定有两组查询对应的文档集合，每组查询结果集合中有三个等级，分别是等级1、等级2和等级3。举例说明，在第一组查询结果中有三个对象、和，分别属于等级1、等级2和等级3，如图13所示。图中的向量是第二章公式(2.2)中中的权重向量。但是，此时对于每一个查询得到的候选文档集合，都有一个各自的权重向量，在应用中非常麻烦。因此，对同一组查询结果集合中的不同等级的对象的特征向量进行组合，形成新的特征向量，即把对象映射到另一个向量空间，然后根据另一个向量空间的向量进行排序。比如将上述的三个对象重新组合为：、和，并且给这些对象重新打标签，如将、和标记为负相关，相应的、和标记为正相关，如图14所示，即可将以上的排序问题转化为二值分类问题，在该例子中就可以通过训练线性SVM分类器对上述新的向量空间的向量进行分类，进而可以计算得到同一组查询结果集中的不同等级的向量的前后词序。

训练数据表示为，其中为，每一个训练实例由两个特征向量表示，标签由表示，并且的取值为+1和-1，分别表示正相关和负相关。Ranking SVM的最优化问题可以形式化的表示为公式(3.12)。

(3.12)

*s.t.(使得)*

其中为松弛变量，是训练实例的数量，2-范数：https://img-blog.csdn.net/20130624111745156?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvbGVmdF9sYQ==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center，Euclid范数（欧几里得范数，常用计算**向量**长度），即**向量**元素绝对值的平方和再开方，matlab调用函数norm(x, 2)是第二范式，把松弛变量代入公式(3.12)中，可以得到：

(3.13)

其中，加和的第一项是合页损失函数, 第二项为正则项，防止过拟合。

#### IR SVM

IR SVM是Ranking SVM在信息检索领域的一个改进[36]。Ranking SVM在信息检索领域的不足之处包括两个方面：

（1）Ranking SVM是将排序问题转化为分类问题，在学习过程中使用了0-1分类损失函数。在信息检索任务中，最终结果列表中排在前面的对检索效果的影响更大，而Ranking SVM对每两个文档的相对顺序都一视同仁。举例说明，有三个等级，其正确的排序顺序应该为等级1、等级2、等级3，则等级3>等级2和等级3>等级1都是错误的相对顺序，但是二者对Ranking SVM的训练过程造成的影响是相同的，显然，这与实际的排序过程有一定的误差。

（2）另一方面，Ranking SVM同等对待不同查询下的结果对。举例说明：每个文



图13 排序问题示意图



图14 将排序问题转化为分类问题示意图

档使用相应等级来表示，两个查询的结果如表4所示。对于查询一，在转换向量空间时可以有2个“等级3-等级2”的文档对，4个“等级2-等级1”的文档对，2个“等级3-等级1”的文档对，共8个正相关文档对可供训练；对于查询二，则有两个“等级3-等级2”的文档对，八个“等级2等级1”的文档对，四个“等级3-等级1”的文档对，共有14个正相关文档对可供训练。虽然两个查询得到的结果中等级结构相同，但是由于数量不同，查询二对Ranking SVM模型的影响会比查询一大，因此最后的结果会有偏差。同时，这也与信息检索任务中所要求的“每个查询的重要性等同”[23]是不相符的。

表4 排序列表举例

|  |  |
| --- | --- |
| **查询** | **结果** |
| 查询一 | 五个文档：等级3、等级2、等级2、等级1、等级1 |
| 查询二 | 七个文档：等级3、等级2、等级2、等级1、等级1、等级1、等级1 |

由于Ranking SVM以上两个方面的不足之处，IR SVM将二值分类问题改进为代价敏感的分类问题，即对来自不同查询的文档对，或者不同等级的文档对设置不同的损失权重。由于在排序结果列表前面的结果比后面的结果的重要程度高，在计算时会对在前边出错的文档对加大损失函数的权重，即出错的文档在列表中顺序越靠前，所需要付出的代价越高；IR SVM对上述Ranking SVM第二点不足的改进是，对排序出错的文档，如果某条查询的结果列表文档数量较少，则加大损失函数的权重，反之，如果结果列表中文档数量较多，则减轻权重，从而减少因为查询本身对查询效果的影响。IR SVM的最优化问题可以表示成如公式(3.14)所示。

(3.14)

其中表示第*i*个文档对的等级，表示等级下的权重值，表示第*i*个文档对对应的查询文档，表示查询对应的相关文档对的参数。和作为惩罚因子，分别用来弥补上述Ranking SVM两个方面的不足。参数值的确定是使用一个启发式算法：选定一个评价指标，对每个查询，找到能够使得评价指标最优的排序序列，然后随机交换任意两文档的位置，查看评价指标的降低值，重复该过程，最后对降低值取平均值作为分类下对查询文档对所加的权重。参数值是对查询项对应的相关文档的数量取倒数，当相关文档数目较少时，该值相对较大，进而弥补了在文档数较少时Ranking SVM优化函数对其重视程度低的问题。

#### 改进的IR SVM

在IR SVM中考虑到了查询结果列表中排在前面的内容对结果影响较大，因此提出了使用公式(3.14)中的权重参数来解决这一问题，即对不同等级的查询文档设置不同的权重。如上一小节所说，IR SVM使用启发式算法计算值，但是该方法仅仅考虑了由于顺序不同导致的权重值的不同。

经过3.3节的计算，查询文档已经与一个文档序列建立了关系，并且序列中的文档与查询文档通过文本语义相似度值相关联。和查询项相关的文本语义相似度中不仅包含了文档相对顺序的信息，同时还包含了不同位置的文档与查询项的具体关系和一个可以量化的值，因此可以将上述权重参数使用文本语义相似度来表示，具体表示如公式(3.15)所示。

(3.15)

其中和分别表示第*i*个文档对（pair）中两个文档与查询项的文本语义相似度值，的取值为二者中的较大值与二者差值绝对值的乘积。用来表示文档对中排序更靠前的文档在排序列表中的重要程度；指文档对中两个文档与查询项相关度的差值，用来区分不同等级（不同位置）的文档的重视程度。改进之后的参数省去了启发式取值的繁琐过程，同时包含了查询结果的相对顺序和每个查询结果与查询项的相关程度。此时，学习排序算法的最优化问题可以表示成公式（3.16）所示。

（3.16）

其中为公式（3.15）表示的内容，的意义不变，依然表示查询项对应的查询结果数量的倒数，用来避免某一个查询项有过多的查询结果，从而对训练结果造成偏差。改进的IR SVM算法使用SMO（Sequential Minimal Optimization，序列最小最优化）算法实现，同时由于实例集合非线性可分，因此引入核函数，这里使用高斯核函数，如公式（3.17）所示。

(3.17)

其中，为映射函数，在高斯核函数中，并没有指定的具体函数实现，写出该映射函数旨在方便下面对问题的描述。

易知，公式（3.18）是公式（3.16）的等价形式。

（3.18）

引入拉格朗日乘子并对其求极大极小问题后，可以得到对偶问题，如公式（3.19）所示。

(3.19)

使用SMO对上述对偶问题求解的步骤如算法1所示，计算得到所有的值和*b*值，然后可得到SVM分类器，分类决策函数如公式（3.20）所示。

(3.20)

其中是符号函数，当*z*大于等于0时，函数值为+1，当*z*小于0时，函数值为-1。

|  |
| --- |
| **算法1** 改进的IR SVM中的SMO算法部分 |
| **输入：**训练数据集，其中和均属于，*n*为特征数量，，，核函数中参数，精度值，常数*C*；查询项与查询结果的相似度集合，为查询项和文档的语义相似度值；每个查询项对应的结果数量集合；最大迭代数*maxIter*  **输出：**向量，*b*向量  **初始化工作：**创建*alpha*向量并初始化为0，计算所有的值和值  **while** 当前迭代次数 < maxIter **do**  **for all** **do**  计算当前alpha和b下分类器的值和的误差值  **if** Ei超过的范围 **and** 当前alpha[i] 满足限制条件 **then**  随机选择另一个alpha[j]  同时优化alpha[i]和alpha[j]  **if** 通过当前alpha计算得到的误差值都满足要求精度值 **then**  结束本次循环  **end if**  设置b向量  **end if**  **end for**  **if** alpha在上次循环中没有变动 **then**  当前迭代数加一  **end if**  **end while**  **return** alpha, b |

完成上述步骤后，即可通过分类器预测查询结果文档的两两相对顺序关系，然后对所有的相对顺序关系进行汇总，即可得到整体的文档排序关系。本文在第四章中将改进的方法与未使用学习排序模型的算法、原IR SVM方法分别进行了对比，实验结果表明在本文任务总，改进的IR SVM是有效的，并且比原有的IR SVM方法的精确率更高。

## 本章小结

本章首先给出了基于word embedding和学习排序算法的适航领域需求跟踪算法模型框架，并对框架中包含的几个部分进行了简单介绍。接下来详细介绍了算法模型中改进的部分，其中包括：在计算文本语义相似度时提出的在word embedding的基础上添加加权策略和查询扩展，并以此为基础改进了文本语义相似度的计算算法。由于使用单独的信息检索技术在软件需求跟踪关系恢复任务中对数据的特征使用较少，因此提出了使用学习排序算法对结果的精度做进一步的提升，并改进了IR SVM算法。

## 3.3 学习排序算法(TQS)

在以往的需求可追踪链接研究中，通常只使用文本相似度一个特征作为判断软件文档间追踪关系的依据。而在目前的文本检索，自然语言处理等领域，使用机器学习方法结合文本中的多个特征，如词频、文档逆频率，文本长度等来对文本进行检索、分类已经在得到验证。因此，本文将学习排序应用到需求追踪任务当中。在上一节提到，通过计算每一条查询语句与候选文本间的相似度，可以产生一个排序列表，这个排序列表可以作为学习排序的输入。下面小节中将介绍学习排序算法和选取的特征。

### 3.3.1 排序算法选择

在第二章中提到，按照训练样本的不同，学习排序算法可以分为单文档排序、文档对排序和文档列表排序三种方法。其中文档列表排序每次训练对象是整个候选列表，在数据量足够的情况下，拥有最好的效果；而单文档方法由于只考虑了单个文本的特征，效果不如文档对方法。因此，综合效果和实验数据集的规模，本文选择了相对合适的文档对排序算法。 本文根据任务特点选择了针对信息检索领域的排序算法IR SVM，该算法是Ranking SVM在信息检索领域的改进。

#### 3.3.1.1 Ranking SVM

Ranking SVM排序算法主要分为将文档对的排序问题转化为分类问题和使用SVM算法分类模型训练并求解两个步骤。

假设给定查询-候选文档列表，并用一个维的特征向量来表示候选文档，然后输出空间用与查询的相关级别用 来表示，表示文档相关级别的数量，并且相关性级别按照递增排列。本节使用第二章中提到的排序函数将特征计算排序分数， 可以表示为公式(3.6) ，“”表示文档将相对位置关系。

 (3.6)

Herbich等人[51]在2000年首次提出可以将文档对排序问题转化为分类问题。其转化的推导可以假设排序函数为线性函数(可以为任意函数)。表示为公式(3.7)：

 (3.7)

其中是权重向量，该公式表示排序函数为权重向量和特征向量的内积。结合公式(3.8)，可以得到：

 (3.8)

可以由公式(3.8)看到，对于任一文档对，候选文档中 表示文档在文档的靠前位置。可以通过将文本特征向量重新组合为，结合公式(3.7)则可以得到两者相对位置的判别函数。根据线性函数的特性，可以得到每一对文档对相对位置的标签。接下来，可以依次求出候选文档列表中任意一组文档对组合的向量，并为其重新打标签。本文用和分别代表文档对中的第一个和第二个文档特征，两者间先后关系的标签用来表示，形式化的表示如公式(3.9)。

 (3.9)

这样就可以把给定的排序列表作为训练集，构建一个新的带有文档对顺序标签的数据集，见公式(3.10)。

 (3.10)



图15 原始排序问题

这里可以举例说明，假设候选文档与查询文档中共有三个相关等级，现在有两个查询分别对应两个候选文档列表，根据公式(3.7)，每个查询会计算得到一个权重向量，可以用图15形象表示。而本文使用文档对排序方法 ，根据公式(3.10)将文本特征向量两两重新组合为：、和，并给他们打上相应的标签，然后就可以将排序问题转化为了分类问题，如图16所示。



图16 排序问题转化为分类问题

本节使用作为分类算法的训练数据集，使用支持向量机(Support Vector Machine, SVM)作为分类算法，+1表示，为正样本，反之，-1为负样本。这样排序问题转化为0-1分类问题，根据SVM求解过程可以得(3.11)：

(3.11)

在这里是松弛变量，是训练集中实例数量，是第二范式，把松弛变量代入公式(3.12)中，可以得到：

(3.12)

在这里，公式加和的第一项为合页损失函数，第二项是防止过拟合而加入的正则项。

#### 3.3.1.2 IR SVM

Ranking SVM算法作为学习排序模型的排序算法应用于信息检索领域时[43]，有两点不足：

(1) 在上一小节提到，Ranking SVM排序算法的主要思想是将排序问题转化为分类问题。它在训练的时候把不同级别差值文档组，如和，其中 表示与查询相关的等级，同等对待，假如在训练实例中，越多，那么最终得到的结果也会越偏向于。而在信息检索领域的问题中，最终的结果列表中，文档排名越靠前，即r值越高，它对最终检索的结果影响越大，显然Ranking SVM在这里有待改进。

(2) Ranking SVM还有一点没有考虑的因素是，不同查询语句所对应的查询语句对所得的候选文档同等对待，这显示是不合理的。在这里可以举例说明：仍然假设文件的相关等级分为三个等级级别，现有两个查询，每个查询所对应的候选文档如表4所示，从中可以看到，对于结果中可提供正样本数量为14，多于可提供的8个，而由于Ranking SVM对不同查询语句所对应的查询语句对所得的候选文档同等对待，这也会影响最终检索结果。应该调整算法，使文档在相关等级相同的情况下，正样本多的在训练过程中应该对最终结果影响大于。

表4 排序列表实例

|  |  |
| --- | --- |
| **查询** | **结果** |
|  | 文档相关等级分别为：r3, r2, r2, r1, r1 |
|  | 文档相关等级分别为：r3, r2, r2, r1, r1, r1, r1 |

由于Ranking SVM算法的以上两个不足点，本文使用了它在信息检索领域的改进算法IR SVM，通过对它的优化Ranking SVM的损失函数，将0-1分类问题转化为代价敏感的二分类问题。具体来说就是，对于同一查询下的不同文档对，相关级别越高在最终排序结果中的重要性越大，这时候，对该文档对赋予较大的损失权重；对不同查询语句下的候选文档列表，为列表条目越少的文档对数赋予较大的损失权重。基于以上IR SVM改进损失函数可以表示为公式(3.13)。

(3.13)

在这里表示第个文档对的相关等级，表示等级下的权重值，为第个文档对所对应的查询，表示查询对应的相关文档对的参数。和作为惩罚因子，分别用来弥补上述Ranking SVM两个方面的不足。

对于参数值的确定，使用的是一种启发式算法：首先选定一种评价指标，如MAP, NDCG等。对于每条查询语句或者文档，首先找到它的最佳排序列表，然后随机挑选其中一对文档，交换两者的位置，记录下指标的变化值，重复该过程，直到所有文档对交换完成，最后使用评价指标变化的平均值作为参数的值。对于参数，它的主要作用是平衡候选文档列表条目数目所带来的影响，可以用最大候选文档条目和当前查询候选文档条目表示，可由公式(3.14)表示。

(3.14)

其中，代每条查询所对应的候选文档数量。于是可以简单的计算出来。

#### 3.3.1.3 求解IR SVM

令时，公式(3.13)可以等价变化为二次规划问题，如公式(3.15)。

(3.15)

|  |
| --- |
| **算法 1**: 坐标下降法 |
| 初始化为0,其中，令  迭代直至算法收敛： |

本文使用坐标下降法(Dual coordinate descent method)求解对偶问题来间接求解公式(3.15)的对偶问题从而得到它的最优解，首先引入拉格朗日引入拉格朗日乘子，然后对其求极大极小问题后，可以得到它的对偶形式，如公式(3.16)所示。

(3.16)

坐标下降法每次选择一个进行优化，记为迭代指定次数或目标函数值变化量达到指定精度时停止，其过程如算法1所示。

### 3.3.2 特征选择

在学习排序中，影响最后排序结果的另一个因素是文本特征的选择，在以往的研究工作中已经证明一些特征在信息检索领域的有效性。本文从常用的特征中选取五个特征作为排序算法输入。这五个特征中的前四个是依赖查询特征，这些特征强调了查询和候选文档间的关系，最后一个是不依赖查询特征，体现了候选文档本身的特点，主要作用是可以用来加快查询速度。

(1)文本相似度：表示查询查询文档与候选文档间语义的相似程度，可以由本文提出的相似度算法CWI计算得到。

(2)IDF之和：查询中词语在候选文档列表中的IDF值之和，它体现了查询在其对应的结果集的重要程度，可用公式(3.17)表示。

(3.17)

在这里代表查询文档中的单词，代表查询所对应的候选文档列表。

(3)广义Jaccard系数：通过文档向量表示查询文档和任一候选文档的上下文相似度，其计算方法由公式(3.18)所示。

(3.18)

这里代表查询文档的向量表示，是候选文档的向量表示，它们可以由公式(3.3)计算得出。

(4)缺失词数量：即在使用词向量表示文本时，缺失词的数量，体现缺失词语在查询中的重要程度。

(5)文本长度：表示查询候选文档列表的长度，学习排序中常用的不依赖查询特征，可以一定程度体现结果的有效程度。

## 3.4 本章小结

本章首先介绍了面向发动机适航的软件需求追踪算法的模型框架，然后介绍了基于词向量的文本相似度算法，算法包括了语义相似度的计算和缺失词语的动态处理，最后，介绍了机器学习排序将排序问题转化为0-1分类问题的原理，并给出了损失函数的优化和推导过程，结合当下的流行的机器学习技术，通过组合文本中的多个特征从而提升结果精度。