基于页面标题比对和页面特征的

仿冒网页识别微引擎

系统说明

版本号：1.0

**目录**

[1 引言 3](#_Toc27603)

[1.1 产品的目的 3](#_Toc31543)

[1.2 产品的范围 3](#_Toc27892)

[1.3 术语与定义 3](#_Toc13228)

[2 产品的总体描述 4](#_Toc22697)

[2.1 产品的背景 4](#_Toc22390)

[2.2 产品的功能 5](#_Toc17976)

[2.3 产品的特性 6](#_Toc23335)

[2.3.1 仿冒网页检测的常用方法 6](#_Toc20649)

[2.3.2 页面标题比对方法 6](#_Toc2914)

[2.3.3 分词技术概述 8](#_Toc8427)

[2.3.4 基于页面身份特征提取的方法 9](#_Toc2280)

[2.3.5 线性分类器 12](#_Toc10776)

[3 需求分析 14](#_Toc575)

[3.1 用例模型 14](#_Toc12564)

[3.1.1用户用例图 14](#_Toc32620)

[3.2 领域模型 15](#_Toc16848)

[3.3 过程模型 15](#_Toc30864)

[3.3.1 基于标题比对网页检测模型及流程设计 15](#_Toc2040)

[3.3.2 基于页面身份特征的网页检测模型及流程设计 17](#_Toc30322)

[3.3.3 白名单模块设计及流程分析 18](#_Toc28130)

[3.3.4 基于页面身份特征评价模块设计及流程分析 19](#_Toc13358)

[3.3.5参数训练模块设计及流程分析 20](#_Toc15330)

[4 体系结构设计 22](#_Toc13231)

[4.1 系统综述 22](#_Toc895)

[4.2 系统总体架构 22](#_Toc664)

[4.3 功能分解视图 23](#_Toc15648)

[4. 4 构件依赖视图 24](#_Toc7107)

[5 构件设计 25](#_Toc29721)

[5.1白名单构件 25](#_Toc29066)

[5.2页面标题比对模块构件 27](#_Toc31822)

[5.3页面身份特征检测模块构件 29](#_Toc20332)

[5.4 参数训练模块构件 30](#_Toc17861)

[6 部署环境 31](#_Toc30161)

# 引言

## 产品的目的

仿冒网页通常通过模仿知名网站，来进行网络钓鱼。该系统主要使用基于页面标题比对和页面身份特征两种方法对仿冒网页进行识别，并设计和开发识别微引擎。该微引擎就是利用仿冒网页模仿各种知名站点的特点，首先对知名站点网页和疑似网页的标题进行提取，通过相应的标题比对方法，得出基于标题比对的评价结果。同时，又因为白名单中的合法网页数量的局限性，非法网页常常无法找到相对应的合法网页，进而尝试对网页的页面身份特征，进行提取和量化，并使用线性分类器对网页进行分类，得出相应结论。

## 产品的范围

通过在网络流出口处（针对区域网络，如：大型社区）的部署，获取大量网民访问的URL数据,识别出其中的仿冒网页（或倾向于仿冒网页的网页）。

## 术语与定义

|  |  |
| --- | --- |
| **术语** | **说明** |
| 页面标题 | 指HTML中<title>域中的文字 |
| 身份特征 | 指网页的相关特征，包括域名，URL，网页内容等 |
| 仿冒网页 | 通过模仿知名站点，进行相关非法活动，比如网络钓鱼 |

# 产品的总体描述

## 产品的背景

在线互联网服务给广大用户带来便利的同时，以骗取用户各种信息为目的仿冒（钓鱼）网页迅速增多。这些非法网站常常发送给用户大量的垃圾邮件，这些邮件往往可以欺骗那些没有网络安全经验的用户，使得他们相信邮件来自合法可信的组织，并点击其中指向仿冒网页的超链接。而这些仿冒页面常常存在对原始主站页面的高度模仿，使用户降低防范意识，以便进一步进行“网络钓鱼”，获取私密信息，进而损害广大用户的经济利益。根据中国反钓鱼网站联盟2014年12月份的报告，联盟12月份共处理钓鱼网站3940个，而截至 2014 年12月份，联盟累计认定并处理钓鱼网站220033个。12月份联盟接到的钓鱼网站举报中，涉及淘宝网、工商银行、建设银行、中国银联四家单位的钓鱼网站总量占全部举报量的97.24%。其中，仿冒淘宝网的钓鱼网站处于钓鱼网站仿冒对象的第一位。巨大的经济利益驱动使网络钓鱼层出不穷，并已然形成了产业链式的运作模式，网络安全形势日趋严峻。

仿冒网页的生命周期很短，通常其生存时间不超过一个月，有的只存活几天甚至几个小时，在进行网络钓鱼成功后，便立即关闭站点，监管部门很难对其监控和取证，如图1-1是模仿中国工商银行主页的仿冒网页。在这种大背景下，开展针对仿冒网页进行网络钓鱼的研究是极有现实意义的，相应的检测技术的研究成果将应用于净化网络环境，促进互联网相关产业健康可持续地发展。



图1-1 模仿中国工商银行网上银行的仿冒网页

随着互联网的迅速发展，日常生活中的许多事情都在向着网络化的方向发展与迁移，也正因为网络的相对自由与进入时的较低门槛，大量的仿冒网页出现。通常，不法分子通过各种手段模仿电子商务网站或商业银行的页面，使得用户在不知情的情况下泄漏自己的信息。因此，仿冒网页的识别和检测技术的研究，对于防范和降低此类网站对电子商务领域和个人隐私信息的危害有着很大意义。

目前常用的仿冒网页检测方法包括网络异常行为检测、黑名单实时检测、网页间相似性检测、网页异常特征检测、电子邮件检测、个人信息保护策略等。但由于此种问题的复杂性，仅使用单一的检测防范措施，难以达到预期效果。

## 产品的功能

该引擎主要使用两种方法对仿冒网页进行识别——基于页面标题比对和页面特征的仿冒网页识别微引擎。该微引擎就是利用非法网页通常模仿各种知名站点的特点，首先对知名站点网页和疑似网页的标题进行提取，通过相应的标题比对方法，对网页得出基于标题比对的评价结论。同时，又因为白名单中的合法网页数量的局限性，非法网页常常无法找到相对应的合法网页，进而尝试对网页的页面特征，进行提取和量化，以线性分类器的方法对网页进行评价，并给出相应的结论。

## 产品的特性

### 2.3.1 仿冒网页检测的常用方法

仿冒网页进行网络钓鱼的危害，正在逐年增加，这种局面引起了学术界、产业界的关注，同时，信息安全工作者也在积极研究此类问题，并提出了一系列可以采取的防范措施。目前，反网络钓鱼技术基本上可分为3大类：基于浏览器的防范、基于服务器的防范和独立的第三方检测方法。基于浏览器的防范措施，是通过嵌入浏览器开发者提供的检测插件，来检测和提示用户相关网页认证信息。目前应用比较广泛的微软的Internet Explorer(IE)和Google的浏览器都内建了基于黑名单的检测技术，这种基于黑名单的方法，是通过收集已经发现并确认的仿冒网页URL，并构建一个庞大的黑名单库，当用户访问当前URL时，将会与黑名单中的URL进行比较，如果此URL出现在黑名单中，将阻止用户对当前URL的访问，并进行相应的警告和提示。基于服务器的防范，是指服务器通过信息认证来防范网络钓鱼，例如使用动态安全皮肤、使用安全电子证书等。独立的第三方防范措施，其主要目的是发现并共享仿冒网站相关信息，具体方法包括网络异常行为检测、黑名单实时检测、网页间相似性检测、网页异常特征检测、电子邮件检测、个人信息保护策略等。但由于此种问题的复杂性，仅使用单一的检测防范措施，难以达到预期效果。

### 2.3.2 页面标题比对方法

在仿冒网页中，页面的标题首先通常会复制被模仿网页的完整标题，所以基于页面标题比对的方法，首先对黑白名单中网页的标题进行比对。假设某仿冒网页的标题为TITLE1 ，某知名站点的网页标题为TITLE2，如果TITLE1和TITLE2完全相同，且TITLE1和TITLE2所对应的网页域名不同，则此时认为，具有TITLE1的网页为仿冒网页，且仿冒的对象为具有TITLE2的网页。这种基于100%字符串比较的方法，较为直接明了，且比对速度快，效率高。但由于词语及语义的复杂性，在两个网页标题并不100% 相同的情况下，却拥有极为相似的语义，极易使用户发生混淆。下面将介绍三种方法，对此种相近的标题比对方法进一步地介绍和研究。

越来越多的仿冒网页，开始使用标题的近似相近，来躲避常规的字符串检测方法。例如，最近比较热门的综艺节目，《奔跑吧，兄弟》，成为仿冒网页的大热门。而仿冒网页的标题变为 “兄弟，奔跑吧”，从语句语义的角度分析，极易混淆，也很具有迷惑性。面对这种情况，简单的比对方式就无法应对了。

针对近似标题问题，对标题比对方法进行进一步的扩充。首先对需要比对的标题，TITLE1和TITLE2进行分词处理。TITLE1和TITLE2中包含的所有单词记为集合R，同时记q为TITLE1和TITLE2中都包含的单词数目，s是TITLE1中存在，而TITLE2中不存在的单词数目，r为TITLE1中不存在而TITLE2中存在的单词数目。至此，采用非恒定的相似度评价系数(Jaccard系数)，及两个标题的相似度为：q/(q+s+r)

对于刚才举的例子：

标题1：奔跑吧，兄弟

标题2：兄弟，奔跑吧

分词后的结果：

标题1：奔跑，吧，兄弟

标题2：兄弟，奔跑，吧

那么，q = 3，r =0，s =0。则标题1和标题2的相似度为100%。

在这种方法中，因为并不考虑每个单词在每个标题中的位置，故所求相同占比称为非对称相同词语占比。虽然结果可以反映出这两个标题具有很高的相近程度，但100%的近似比仍然不够合理，导致这种现象的发生，是没有考虑两个标题分词后，词语的对称性问题。下面的方法就将分词后词语的位置也加入了比对的策略中。

首先，同样对需要比对的标题，TITLE1和TITLE2进行分词处理。TITLE1和TITLE2中包含的所有单词记为集合R，同时记q为TITLE1和TITLE2中位置和单词均相同的数目，r为TITLE1和TITLE2单词数目的较大值。此时，两个标题的相似度（考虑位置因素）为：q/r

同样，对于刚才举的例子：

标题1：奔跑吧，兄弟

标题2：兄弟，奔跑吧

分词后的结果：

标题1：奔跑(1)，吧(2)，兄弟(3)

标题2：兄弟(1)，奔跑(2)，吧(3)

那么，q =0 ，r =3，则标题1和标题2的相似度为0。

加入分词后的位置因素后，可以从一定程度上说明了两个标题的相异度，但却忽视了相同词语的占比，上述例子中，两个标题的相似度为0，扩大了位置因素的影响效果。综合分析，这两种比对方式的结合可以弥补各自的缺陷，又可以保留整体的比对策略。

上述两种方法都存在相应的局限性，下面将介绍这两种方式的加权求和方法，对于TITLE1和TITLE2，其相似度=k1\*非对称相同词语占比+k2\*对称相同词语占比。

同样，对于刚才举的例子：

标题1：奔跑吧，兄弟

标题2：兄弟，奔跑吧

非对称相同词语占比：100%

对称相同词语占比：0

如果此时k1=k2=0.5，则加权求和的相似度为50%。

经过加权求和的方法求得的标题相似度，则较为合理，不会像单独使用某种方法得出的仅突出某个特性的相似度。

### 2.3.3 分词技术概述

分词，就是将连续的语句序列按照一定的规范重新组合成词序列的过程。英文的分词较为容易，每个单词间有明确的分隔符。而中文分词，需将一个汉字序列切分成一个一个单独的词，但在汉语言中，字、句和段能通过明显的分界符来简单划界，唯独词没有一个形式上的分界符。在针对汉语言文字处理中，中文分词是文本挖掘的基础，中文[分词技术](http://baike.baidu.com/view/2699566.htm)属于[自然语言处理](http://baike.baidu.com/view/18784.htm)技术范畴，其处理过程就是分词算法。现有的分词算法可分为三大类：基于[字符串匹配](http://baike.baidu.com/view/1436430.htm)的分词方法、基于理解的分词方法和基于统计的分词方法。按照是否与词性标注过程相结合，又可以分为单纯分词方法和分词与标注相结合的一体化方法。

### 2.3.4 基于页面身份特征提取的方法

在基于页面标题的比对方法中，需要同时具有黑白名单，又因为白名单中的合法网页数量的局限性，非法网页常常无法找到相对应的合法网页(及被仿冒的网页对象)。而对于既定的网页来说，从其URL地址，页面内容，域名信息等，可以对其身份特征进行提取和量化处理，并以此特征向量对这这个页面进行相应标识。故尝试对疑似名单中网页的身份特征，进行提取和相应标准量化，以线性分类器的方法对疑似名单中的网页进行评价，并给出相应的结论。

微引擎主要从页面URL，网页内容，域名信息，搜索引擎返回信息等方面，选取了页面的8个身份特征。通过提取网页的敏感身份特征，并对其进行量化处理，生成特征向量。

使用特征函数Fi来分别表示其对应的身份特征，定义某一网页的身份特征向量：

F={F1，F2，F3，F4，F5，F6，F7，F8} (2-1)

其中F1为端口是否使用常用端口80，F2为URL是否直接使用IP地址，F3为URL中是否含有过多的“.”，F4为域名注册年龄是否过短，F5为网页中文字域中是否有ICP声明，F6为网页中指向外域和空域的占比是否过多，F7为网页中是否含有过多指向外域的embed标签，F8为网页域名在搜索引擎中是否有返回结果。每个特征函数的输出为实数值，表示网页中对应的身份特征状态，具体定义如下：

F1：端口使用常用端口80。对于仿冒网页，为了使得建站成本降低，常常多个网站共用一个主机，部分仿冒网页常常不使用HTTP协议常用端口80。所以，当网页使用非80端口时，则认为此网页接近仿冒网页。

F1的定义如下：

1 网页使用非80端口

F1=

-1 网页使用80端口

F2：URL直接使用IP地址。对于传统的网页来说，为了避免包含多数字的IP地址记忆，通常会申请一个域名，而仿冒网页，为了节省成本，可能就在某个个人PC机上搭建web服务，在获取利益后，则立即关闭。所以，特征2检测URL中是否包含IP地址。

F2的定义如下：

1 URL直接使用IP地址

F2=

-1 URL使用域名地址

F3：URL中含有“.”的个数。检查URL中是否包含过多的“.”，仿冒网页通常在伪装URL地址时，将URL拼接的很长，来迷惑用户。当URL中“.”的个数较多时（通常 > 5），特征3认为该网页接近仿冒网页。

F3的定义如下：

1 URL中包含的“.”数量大于5个

F3=

-1 URL中包含的“.”数量不大于5个

F4：域名注册年龄。仿冒网页的生存周期通常会很短，往往也就是几天的时间。仿冒网页的域名通常注册时间很短，一方面，这样可以降低费用，另一方面，仿冒网页为避免其网站地址出现在各个著名浏览器的黑名单之中。一般认为，域名注册时间超过12个月的网站更接近于合法网站。

F4的定义如下：

1 域名注册年龄不超过12个月

F4=

-1 域名注册年龄大于12个月

F5：网页中文字域的ICP声明。ICP许可证是指由各地通信管理部门核准颁发的“中华人民共和国电信与信息服务业务经营许可”，可用来唯一地标识网站身份。通常，境内的合法网页会在在页面底部显著的位置，声明自己的ICP许可证号。

F5的定义如下：

1 页面文字中不含有ICP声明

F5=

-1 页面文字中含有ICP声明

F6：网页中指向外域和空域的占比。仿冒网页为了达到模仿合法网页的目的，通常将自己的链接指向原合法网页的资源上，或是指向其它大量的合法网页，以增加自己的可信度。定义，Nt为网页中链接对象的总数，Nn为网页中空链接对象个数，No为指向外域的链接对象的个数。如果Nn+No的占比过大，则认为该网站页面的链接异常，定义R=(Nn+No)/Nt ，其值越趋近于1，网页异常程度越大，其值越接近0，反映该网站页面越正常。

F6的定义如下：

1 求得R值大于0.3

F6=

-1 求得R值不大于0.3

F7：网页中含有大量指向外域的<embed>标签。<embed>标签是HTML5新增标签，可以用来插入各种媒体，包括音频和视频。近期发现，仿冒网页也开始使用一些HTML5新增的特性，使得网页在引用原始网页中多媒体资源变得更容易。在特征7中，对于大量引用外域的多媒体资源的网页，认为其接近于仿冒网页。

F7的定义如下：

1 <embed>标签引用外域资源比大于20%

F7=

-1 <embed>标签引用外域资源比不大于20%

F8：网页域名在搜索引擎中的返回结果。合法网页的域名，通常可以在搜索引擎中轻易的获取结果，而仿冒网页因为生存时间短，价值小，搜索引擎很少会收录到。所以，在特征8中，借助搜索引擎，当返回结果中，无相应域名搜索值，则认为此页面接近仿冒网页。

F8的定义如下：

1 在搜索引擎中搜索域名，无相应域名的返回结果

F8=

-1 在搜索引擎中搜索域名，有相应域名的返回结果

### 2.3.5 线性分类器

使用网页爬虫，获取页面的HTML源码，可以对页面的内容特征进行提取，包括链接比，ICP声明等，而对于URL中的特征，采用相应字符串匹配和识别算法来实现提取特征，对于域名和搜索引擎部分，则借助相应的应用接口，完成信息提取，和特征获取。根据上述敏感特征，判断一个网站是否为仿冒网页时，使用线性分类器进行处理，线性分类函数如式(2-2)：

 （2-2）

式(2-2)中，Fi表示网页敏感身份特征的取值。Fi=1时，判断该网站为仿冒（网络钓鱼攻击）网站；Fi=-1时，则表示页面正常，该网站为正常网站。式(2-2)中，wi为8个敏感特征分量的权值，其相应的计算公式如式(2-3)：

 （2-3）

而上式中ei的计算公式为式(2-4)：

 （2-4）

式（2-4）中，TFi和FFi分别为单独使用敏感特征Fi检测页面时的正确率和误判率。

式(2-2)，其实就是一个线性函数，具有8个元，及将有8个影响因素，这8个因素分别对应着8个特征的取值，同时权重系数的引入，是对每个特征影响效用的比重的量化处理，权重系数越大，说明在此线性函数中，此种特征对结果的影响效用越大。

使用此种方式，得出的S值将在区间[-1，1]中，当S接近于-1时，则认为此网页为合法网页，当S值更接近于1时，则认为此网页为仿冒网页。

# 需求分析

## 用例模型

### 3.1.1用户用例图

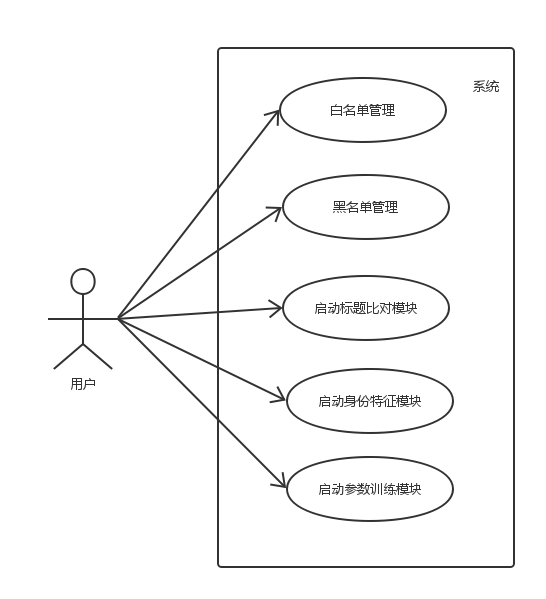


图3-1 用户用例图

## 领域模型

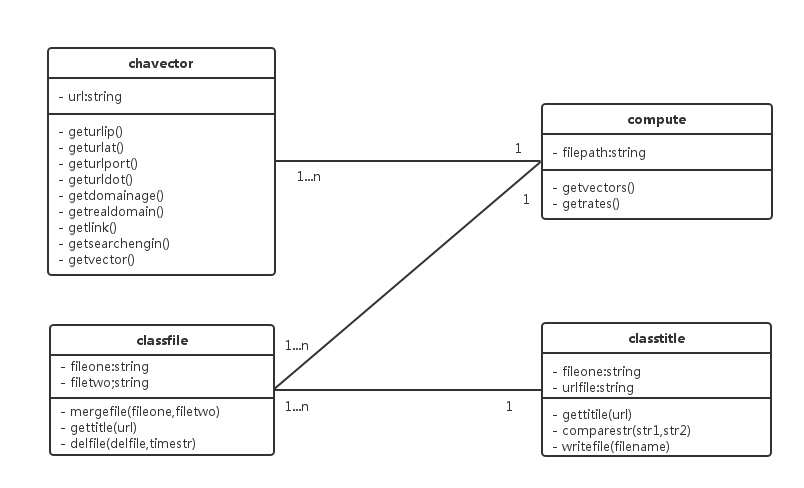


图3-2 类图

## 过程模型

### 3.3.1 基于标题比对网页检测模型及流程设计

如图3-3，是基于标题比对检测模型及流程设计图：

是

否

白名单url网页标题

黑名单url网页标题

标题一样

相似度为100%

对两个标题进行分词处理

分别对两个标题，求得相同词语占比和基于位置的相同词语占比

加权求和，得出相似度

图3-3 基于标题比对检测模型及流程设计

标题比对模型中，首先获取白名单中某个URL的网页标题，黑名单中某个URL的网页标题。第一步，判断两个网页标题是否完全一致，如果标题完全相同，则得出结论，网页标题的相似度为100%，同时标题比对过程结束；如果两个标题并不完全相同，使用第二种比对方法——相似度=k1\*非对称相同词语占比+k2\*对称相同词语占比。当使用这种比对方法时，首先对两个标题进行分词处理，分别计算非对称相同词语占比和对称相同词语占比，最后加权求和，得出标题相似度，同时标题比对过程结束。

### 3.3.2 基于页面身份特征的网页检测模型及流程设计

如图3-4，是基于页面身份特征检测模型及流程设计图：

网页url

对8个页面特征提取

对特征进行量化处理，每个特征返回1或-1

使用加权线性分类器，计算评价值

根据评价值，得出评价结果

图3-4 基于页面身份特征检测模型及流程设计

在页面身份特征作为检测方法的模型中，第一步，使用网页爬虫抓取网页，同时结合url字符串分析，获取8个相应的页面身份特征值。然后，将上一步骤中获得身份特征值进行量化处理，量化为1（表示该网页的此特征异常）或-1（表示该网页的此特征正常），得到网页特征向量F={F1，F2，F3，F4，F5，F6，F7，F8}，其中F1为端口是否使用常用端口80，F2为URL是否直接使用IP地址，F3为URL中是否含有过多的“.”，F4为域名注册年龄是否过短，F5为网页中文字域中是否有ICP声明，F6为网页中指向外域和空域的占比是否过多，F7为网页中是否含有过多指向外域的embed标签，F8为网页域名在搜索引擎中是否有返回结果。

最后，使用加权线性分类器，计算评价值，并得出评价结果，结束检测。

### 3.3.3 白名单模块设计及流程分析

白名单是对已知合法网页的基本信息存储，包括基本的URL和相应的标题内容，当以基于页面标题比对方法检测网页时，可直接获取合法网页的标题信息。该模块具有白名单的增加(如图3-5)，合并（包括去重与排序功能)，删除（如图3-6）等基本功能。

合法的url

记录下抓取时间，抓取url，标题

对数据进行去重和按时间排序

分类存储数据

抓取网页内容，并提取页面标题

图3-5 引擎中白名单模块设计——获取页面标题

已有白名单文件

对已有记录的添加，合并，和按抓取时间删除

对更改后的数据进行去重和排序

图3-6 引擎中白名单模块设计——白名单文件的管理

### 3.3.4 基于页面身份特征评价模块设计及流程分析

在以比对为方式的探测中，白名单的数量总是有限的。而基于页面特征的方式是从网页自身特征出发，不需要与白名单中的网页比对。该模块首先对网页的特征进行提取，然后对各个特征量化处理，最后用线性分类器进行分类，如图3-7所示。

可疑网页url

提取页面的8大特征，

并生成特征向量

使用如下线性分类器：

得到S值

根据S值进行评估

图3-7 引擎中页面特征获取模块设计

### 3.3.5参数训练模块设计及流程分析

参数训练模块是较为特殊的模块，该模块与基于页面特征评价模块紧密结合，为线性分类器提供特征占比参数。因为仿冒网页的生存周期一般较短，使用线上的学习方法，会有大量的因时效性问题带来的无效的网页，使得线上学习出现困难。所以本模块是基于线下的本地训练模块，如图3-8。

本地黑白网页数据文件

提取页面的8大特征，并分别按以下步骤处理

分别计算ＴＦｉ和ＦＦｉ ，ＴＦｉ和ＦＦｉ分别为单独使用敏感特征Ｆｉ检测页面时的正确率和误判率

计算 

计算权重 

得到各个特征的参数比

图3-8 引擎中本地参数训练模块设计

在本地训练时，需要首先将已经存在本地的网页进行分类处理，并统一为莫一种既定格式，以便下面的学习模块学习使用。基本的过程与提取页面身份特征类似，只是在计算权重值时，使用式(2-3)，式(2-4)来计算相应权值。这种计算权重占比的方法，其实是基于概率模型的，即在已知的网页中，某个特征在网页中出现的次数越多，且分类效果越好，其相应的权重就会越大。

# 体系结构设计

## 系统综述

该引擎主要使用两种方法对仿冒网页进行识别——基于页面标题比对和页面特征的仿冒网页识别微引擎。该微引擎就是利用非法网页通常模仿各种知名站点的特点，首先对知名站点网页和疑似网页的标题进行提取，通过相应的标题比对方法，对网页得出基于标题比对的评价结论。同时，又因为白名单中的合法网页数量的局限性，非法网页常常无法找到相对应的合法网页，进而尝试对网页的页面特征，进行提取和量化，以线性分类器的方法对网页进行评价，并给出相应的结论。

## 系统总体架构

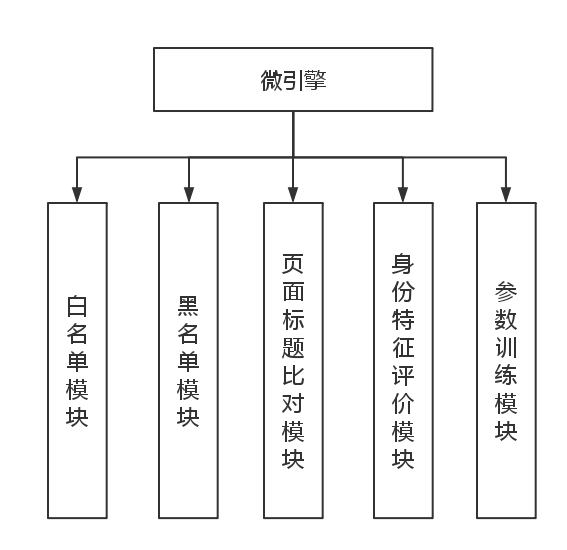


图4-1 系统模块图

## 功能分解视图

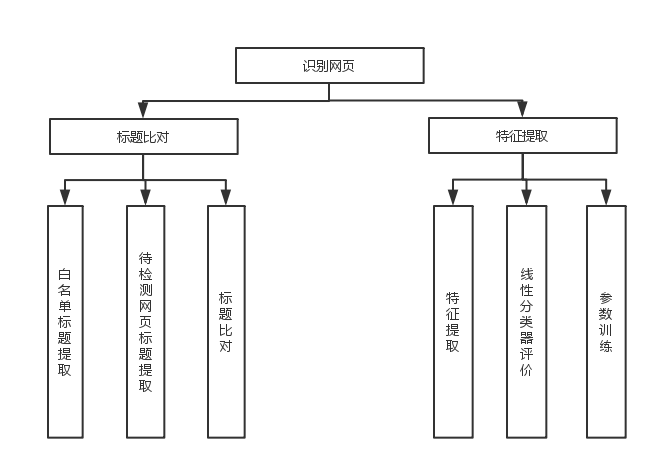


图4-2 功能分解图

## 4.4 构件依赖视图



图4-3 构建依赖图

# 构件设计

## 5.1白名单构件

对于在Linux下运行的命令行程序，关于命令的帮助说明，可以对整个程序的命令进行全面的解释和说明。在字符界面中运行以下命令：

python filemain.py -h

在字符界面中会提示以下信息：

|  |
| --- |
| usage: filemain.py [-h] [-a] [-m] [-d] argumentone argumenttwo  positional arguments:  argumentone the first argument is filepath  argumenttwo the second argument is filepath or time string  optional arguments:  -h, --help show this help message and exit  -a, --add add the whiteurl,the argumentone is the typewhite,the  argumenttwo is url file  -m, --merge merge the whiteurl,  -d, --delete delete the whiteurl by time string the format is like  "[2016-03-21 19:33:34]" |

Linux中shell字符界面中对于中文的支持不够灵活，故系统中帮助信息统一使用英文。在上述帮助信息中，具体介绍了两个参数的含义，和各个操作选项。在实现的其它模块中，都有对相应命令的帮助信息选项，在下面的介绍中将不再重复。

**白名单添加功能**

白名单添加模块，就是将用户提交的URL，获取相应的标题，按照统一格式存储。例如，testurl.txt包含如下URL，如表5-1：

表5-1 testurl.txt文件格式及内容表

|  |
| --- |
| URL条目 |
| http://www.cac.gov.cn  http://www.huanqiu.com  http://www.china.com.cn  http://www.xinhuanet.com  http://sh.qihoo.com 360 |

在命令行敲击如下：

python filemain.py -a whiteurltest testurl.txt

将结果存储在whiteurltest文件中，运行结果如表5-2：

表5-2 whiteurltest文件格式及内容表

|  |
| --- |
| zhang@ubuntu:~/web$ cat whiteurltest |
| [2016-04-08 17:39:20] 1 http://www.cac.gov.cn 中共中央网络安全和信息化领导小组办公室  [2016-04-08 17:39:22] 2 http://www.huanqiu.com 环球网\_全球生活新门户\_环球时报旗下网站  [2016-04-08 17:39:23] 3 http://www.china.com.cn 中国网--网上中国  [2016-04-08 17:39:26] 4 http://www.xinhuanet.com 新华网\_传播中国报道世界  [2016-04-08 17:39:31] 5 http://sh.qihoo.com 360新闻搜索-搜一下，知天下 |

在白名单文件中，每条记录都包含4个字段，第一个字段是网页标题抓取的时间，第二个字段是在本文件中的记录序列，第三个字段是网页的URL，第四个字段是相应的网页标题。

**白名单删除功能**

白名单删除模块是对白名单的简易管理，由用户指定白名单文件路径，和一个时间字符串，该模块会将白名单中时间值小于用户输入时间值的记录删除，在删除模块的中间过程，会调用白名单合并模块的功能，故下面将不会单独介绍合并功能。例如，使用如下命令：

python filemain.py -d whiteurltest "[2015-04-18 17:39:26]"

对上一模块中的白名单进行管理，whiteurltest文件中的记录将变为如表5-3：

表5-3 whiteurltest文件格式及内容表

|  |
| --- |
| zhang@ubuntu:~/web$ cat whiteurltest |
| [2016-04-08 17:39:26] 1 http://www.xinhuanet.com 新华网\_传播中国报道世界  [2016-04-08 17:39:31] 2 http://sh.qihoo.com 360新闻搜索-搜一下，知天下 |

**白名单模块中的异常检测功能**

一个健壮性强的程序，需要对各种异常进行合理的检测和友好提示信息。当网络环境出现问题时，本程序将提示类似如下的信息：

network meets some bad things:

Reason: [Errno -3] Temporary failure in name resolution

当用户敲击的命令存在问题时，包括命令选项不全，字符串格式有误，文件路径存在错误等等，系统都会有相应的信息提示，下面是两个常见错误：

用户输入命令的选项不全：

zhang@ubuntu:~/web$ python filemain.py

此时，提示信息：

usage: filemain.py [-h] [-a] [-m] [-d] argumentone argumenttwo

filemain.py: error: too few arguments

在白名单删除模块，时间字符串格式存在问题：

python filemain.py -d whiteurltest "2015-04-18 17:39:20"

此时，提示信息：

timestr is not right

同样，在其他实现的模块中，都有对各种异常的检测，当介绍其它模块中，将不会重复描述。

## 5.2页面标题比对模块构件

**对于网页标题抓取的关键代码实现**

整个引擎中，各种网页信息的获取都离不开网页爬虫的使用。在白名单模块和页面标题比对模块中，都有对网页标题抓取的过程，表5-4是使用python语言实现gettitle()函数的具体代码：

表5-4 gettitle()函数的具体实现代码表

|  |
| --- |
| python实现代码 |
| def gettitle(self,url):  result=[]  req = Request(url)  try:  response = urlopen(req) #解析url，获取网页源码  except URLError , e : #异常处理  if isinstance(e.reason, socket.timeout):  return None  print "network meets some bad things: "  print "Reason: ",e.reason  return None  else:  if response!=None: #解析网页html源码  the\_page = response.read()  char=chardet.detect(the\_page) the\_page.decode(char['encoding'],'ignore').encode('utf-8','ignore')  realurl= ''  realurl = response.geturl().decode('utf-8','ignore')  soup = BeautifulSoup(the\_page)  title=''  if soup.title!=None: #获取标题  title=soup.title.string  result.append(realurl.encode('utf-8','ignore'))  result.append(title.encode('utf-8','ignore'))  finally:  print 'gettitle success ----',url  if len(result)==2:  return result  else:  return None |

**标题比对方法实现**

这个部分是对上面介绍的标题比对模型的实现，通过用户提供一个白名单，和一个需要进行比对的url文件，程序通过双重循环来实现两两比对，并将比对结果写入到用户指定的文件中。例如，如下命令，实现将testurl.txt中网页标题与whiteurltest文件中的标题内容进行比对，并将比对结果存入testtitle中：

python titlemain.py whiteurltest testurl.txt testtitle

程序执行完后，查看testtitle文件内容，表5-6中显示了testtitle文件的部分内容。每条记录的第一个字段表示比对完成时间，下面分别是白名单中的url及相应标题，和比对文件中的url及相应标题，最后是两个网页标题的相似程度值，单位是百分比。

表5-5 testtitle文件格式及内容表

|  |
| --- |
| zhang@ubuntu:~/web$ cat testtitle |
| [2016-04-09 01:14:34] http://news.ifeng.com 凤凰资讯-凤凰网 http://www.china.com.cn 中国网--网上中国 15.2380952381  [2016-04-09 01:14:34] http://www.huanqiu.com 环球网\_全球生活新门户\_环球时报旗下网站 http://www.xinhuanet.com 新华网\_传播中国报道世界 8.28571428571  [2016-04-09 01:14:34] http://www.china.com.cn 中国网--网上中国 http://news.ifeng.com 凤凰资讯-凤凰网 15.2380952381  [2016-04-09 01:14:34] http://www.china.com.cn 中国网--网上中国 http://www.xinhuanet.com 新华网\_传播中国报道世界 6.66666666667  [2016-04-09 01:19:01] http://news.ifeng.com 凤凰资讯-凤凰网 http://news.ifeng.com 凤凰资讯-凤凰网 100 |

## 5.3页面身份特征检测模块构件

**页面身份特征检测模块实现**

页面特征模块主要是对用提供的url，获取身份特征后，便会使用线性分类函数计算相应的评价值。例如，如下命令，将会提取上文中的testurl.txt中的网页特征，计算完成后，并将结果存入testvector中：

python vectormain.py testurl.txt testvector

程序执行完后，查看testvector文件内容，表5-6中显示了testvector文件的内容。每条记录的第一个字段表示网页的url，第二个字段是此种方法给出的评价值。在testurl.txt中提供了5个合法的网页url，从结果中看到评价值均接近于-1，表示testurl.txt中的网页更接近于合法网页。

表5-6 testvector文件格式及内容表

|  |
| --- |
| zhang@ubuntu:~/web$ cat testvector |
| http://www.xinhuanet.com -0.789473684208  http://news.ifeng.com -0.789473684208  http://news.163.com -0.789473684208  http://sh.qihoo.com -0.789473684208  http://www.huanqiu.com -0.999999999998 |

## 5.4 参数训练模块构件

**参数训练模块输入**

参数训练是一个较为特殊的模块，本模块主要是为特征值计算提供相应的特征占比参数。因为仿冒网页的生存周期较短，线上训练会存在大量url不可抓取的情况。本地参数训练，是利用以往保存的网页源码，在本地搭建一个web服务，程序通过学习本地的网页特征，来得出相应的参数值。

下面是一个该模块的输入样例，每条记录包含3个字段，第一个字段是本地web服务网页url（测试时，已经在本地172.31.59.57的主机上搭建了web服务，另外因为知名站点的生存稳定性，白名单中的网页此字段就为其url，在获取时，直接进行网页爬取），第二个字段是网页本来的url，最后一个是标识字段，1代表已知非法网页，-1代表已知合法网页。

表5-7 testvector文件格式及内容表

|  |
| --- |
| zhang@ubuntu:~/web$ cat testvector |
| http://172.31.159.57/钓鱼分类/银行/中国建设银行/hqxsvu/main.html http://admin.hqxsvu.com/app/ccbMain/ 1  http://172.31.159.57/钓鱼分类/银行/工商银行/dddsg/main.html http://dddsg.pw/icbc/ 1  http://172.31.159.57/钓鱼分类/门户/qq邮箱/qq.y.compuinter/main.html http://qq.y.compuinter.com/588/log.htm 1  http://172.31.159.57/钓鱼分类/门户/中国移动/hjf-10086/main.html http://hjf-10086.com/ 1  http://news.sohu.com/ http://news.sohu.com/ -1  http://news.163.com/ http://news.163.com/ -1  http://news.ifeng.com/ http://news.ifeng.com/ -1  http://www.cac.gov.cn/ http://www.cac.gov.cn/ -1 |

**参数训练模块输出**

在特征选取上，一共选取了8个特征，所以参数训练的结果是8个特征的占比参数。在测试样例中，分别选取了100个本地黑名单网页，100个白名单网页，得出的参数训练结果如表5-8。表中的值越大，代表着在这批训练样本中，该值对应的特征，在分类网页时，分类影响因子越大。

表5-8 参数训练结果

|  |
| --- |
| 8个特征的训练权重值 |
| 0.01219512195 0.0365853658537 0.0284552845528 0.345528455285 0.126016260163 0.10162601626 0.0447154471545 0.30487804878 |

# 部署环境

该节介绍了系统所需要的硬件环境和软件环境，以及在网络配置的环境。

* 硬件环境

系统运行开发设备列表如下表6-1所示：

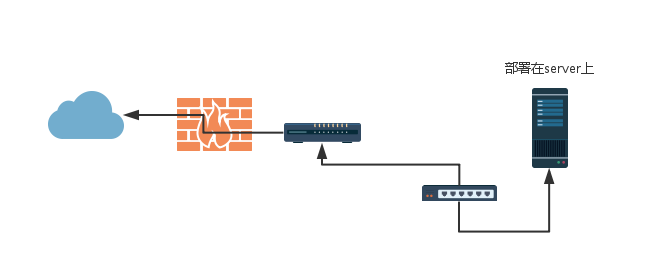
表6-1 硬件配置表

|  |  |
| --- | --- |
| **硬 件** | **详**细要 求要**求** |
| 处理器 | CPU主频在2.5GHz以上 |
| 内存 | 至少512M |
| 外存 | 至少10G |
| 联机 | 连接互联网 |

* 软件环境

1. centos 7 64位系统
2. Python2.7.9

* 系统布置示意图，如图6-1所示：

图6-1 微引擎部署图