Personal Home Page Classifier

BY

ROBIN JOHN VARGHESE
B.E., Mumbai University INDIA, 2007

PROSPECTUS

Submitted in partial fulfillment of the requirements for
the degree of Master of Science in Computer Science
in the Graduate School of
Binghamton University
State University of New York
2011

Accepted in partial fullfillment of the requirements for
the degree of Master of Science in Computer Science
in the Graduate School of
Binghamton University
State University of New York
2011

Dr. Weiyi Meng ____
Department of Computer Science.

Dr. Lei Yu ____
Department of Computer Science.

ABSTRACT

Search engines are increasingly replacing the role of libraries in facilitating information discovery and access. Search engines can improve, if we expand our understanding of user's behavior and the underlying intent with which users conduct searches. We seek to address one of these search intents. In many scenarios, when a user submits a topic query to a search engine, user does not expect just supporting documents for that search query. Futhermore, user is interested in finding experts from that specific domain. If possible, an ordered list of these experts. Our immediate goal is to develop a strong mechanism to automatcally identify candidate Personal Home-Pages (PHP)s from the web. Additional to a design & implementation of a PHP classifier, we propose a metasearch framework that can assort PHPs. Finally we evaluate our implementation across compiled test data sets and investigate false classifications.

Table of Content

Categories and Subject Descriptors

H.2.8 [**Database Application**]: [database application, data mining, classification]; H.3.3 [**Information Storage and retrieval**]: [information retrieval, metasearch]; I.5 [**Pattern Recognition**]: statistical and structural

Design, Experimentation and Verification

Expert/Faculty/Personal homepage finding, web mining, classification

1. INTRODUCTION

Text REtrieval Conference (TREC) series[1], defines Expert Finding (EF) as a task when given a topic area, returns a ranked-list of people and also returns supporting documents. Since, we approach EF as a web mining activity[6], we coin the term "eXpert Mining"(XM).

The benefits of a system as mentioned in abstract are immense at an Enterprise level. Although, we limit our consideration to a deployment in university and academic circles. Search engines can find many valuable documents, but for some questions it is necessary to find the right person rather than the right document[4]. Faculty members could find collaborators for research activities, departments can locate experts in specific domain, assist special interest groups formulation, etc. Thus, reducing the human search effort, which would have been spent in a similar knowledge discovery process. The techniques that we utilize, could also render results for other routine search activities. (for e.g. E-commerce metasearch, TrendSpotting, Finding a Manufacturer etc.) Automatic expert homepage classification is vital for accurate extraction of expert information from the Web[8].

Primarily, such a classification problem can be generalized as web page categorization[17] and then specialized as a home page finding[2] or a web entity finding[3] task. However, in these tasks the target entity is restricted to three types: person, organization and product. In our work we tighten the scope around person entity type.

As we review(refer section2 for details) major contributions and techniques used in this area, we observe few popular approaches. Firstly, bag-of-words model is widely accepted as web page class representation. Further developments have identified key web page streams like title, headers etc.(additional streams in section4.1) to better delegate classification, instead of weighting all words in a web page equally. Secondly, advances are achieved by harnessing multiple external data sources(e.g. DBLP[4]) maintained by independent groups. Thirdly, recent contributions involve extracting features from neighbourhood pages corresponding to a web page under consideration. These contributions have undeniable benefits, however they rely heavly on text classification techniques. Dependence on data sources limits the ability to scale to other topic domains. Some of these techniques have influenced us to adopt feature based decision tree classification, although their implementation involve shallow and manually constructed decision trees. Evaluation of web graph based dependencies for each web page is computationally expensive. Our approach differs from current body of work on several facets. Firstly, we address data acquition by harnessing Metasearch Engine research[15], thus employing Componenet Search Engines (CSE)s. Secondly, we do-not rely on domain specific data sources, thus permitting us to scale. Finally, we avoid application of manually formulated hueristics rules in PHP classification. Instead, we use a combination of widely used and proven feature along with a collection of new features that we propose in section 4.3. Our contributions have been evaluated to support the goodness of features that we

[1] Overview of TREC 2006 `http://goo.gl/z6o9U`

[2] TREC 2001 Web Track

[3] TREC 2009 Entity Track

[4] The DBLP Computer Science Bibliography

propose. Evaluation involves firstly, harnessing CSEs to extract Search Result Record (SRR)s to create 5 data-sets. One of these data-sets, is choosen for training models(refer section5.2) using 10 fold cross validation. Secondly, we select one of these models as our primary PHP classifier. Next, we report evaluation measures after testing the PHP classifier across other 4 data-sets. Lastly, we also scrutinize the instances that cannot be captured by our PHP classifier.

The remainder of this report is organized as : We describe related work in section 2, we propose our plausible XM architecture, sub-tasks involved and an outline of the entire activity in section 3. Design and development of a PHP Classifier is mentioned in section 4. We evaluate our implementaion across test data-sets in section ??. Finally we conclude in section 6 by setting our sight on future work.

2. RELATED WORK

EF when considered as a Knowlege Management venture, has two streams of research[2]. Foremost *Process-centered*[11] or *Personalization*[10] strategy which focuses on finding individual expertise(e.g. Wiki), thus establishing a *Community model*[22]. Models aligned with EF include professional networking portals that allow expression of personal information and relationships as digital documents, thus quantifying an expert with RDF[5], rules, taxonomies[13] and supplementary human-crafted information[7]. Notable examples such as LinkedIn[6], Epernicus[7], VIVO[8] & INDURE[9] rely on these techniques.

Alternatively *Product-centered*[11] or *Codification*[10] strategy is document driven(e.g. ERP[10]), likewise developing a *Cognitive model*[22]. Approach here is to extract revelant Informtion from the web with minimum human intervention, thus can easily tolerate scale like DBLife[11] & ArnetMiner[12]. We intend to purse this research strategy

The inclusion of expert finding in the TREC Enterprise Track[13] has resulted in a great deal of work in this area. General approaches[5] to solve this problem are divided into query dependent and independent. Since, the differences in both these approaches are isolated to disciplinary stages. We shift our perspective to comprehend how the various approaches perform data acquisition. Most of the systems[14],[15] need data(citation info) to be injected by some supervised technique. We intend to address this concern by utilizing SUNY-MSE[16] and other commercial search engines as CSE for retrieving candidate PHPs.

Many recent research towards EF, revolves around using heuristics, regression, Support Vector Machine (SVM) and other data mining techniques. [1] proposes a novel technique, which involves first identifying seed web sources for a targeted research community and subsequently extracting data pages. [23] has implemented a outstanding architecture and acadamic search services. Relations and depedencies with neighborhood web pages are strongly utilized by [8]. The wealth of research in line with related works is in no respect limited to above inclusions.

3. ARCHITECTURE

[5] Resource Description Framework

[6] `http://www.linkedin.com/`

[7] `http://www.epernicus.com/`

[8] `http://vivoweb.org/`

[9] `https://www.indure.org/`

[10] Enterprise Resource Planning

[11] `http://dblife.cs.wisc.edu/`

[12] `http://www.arnetminer.org/`

[13] `http://trec.nist.gov/data/t14_enterprise.html`

[14] UIUC's IRIS `http://www.library.illinois.edu/iris/`

[15] Experts@Minnesota `http://experts.umn.edu/search.pl`

[16] `http://www.mysearchview.com/`

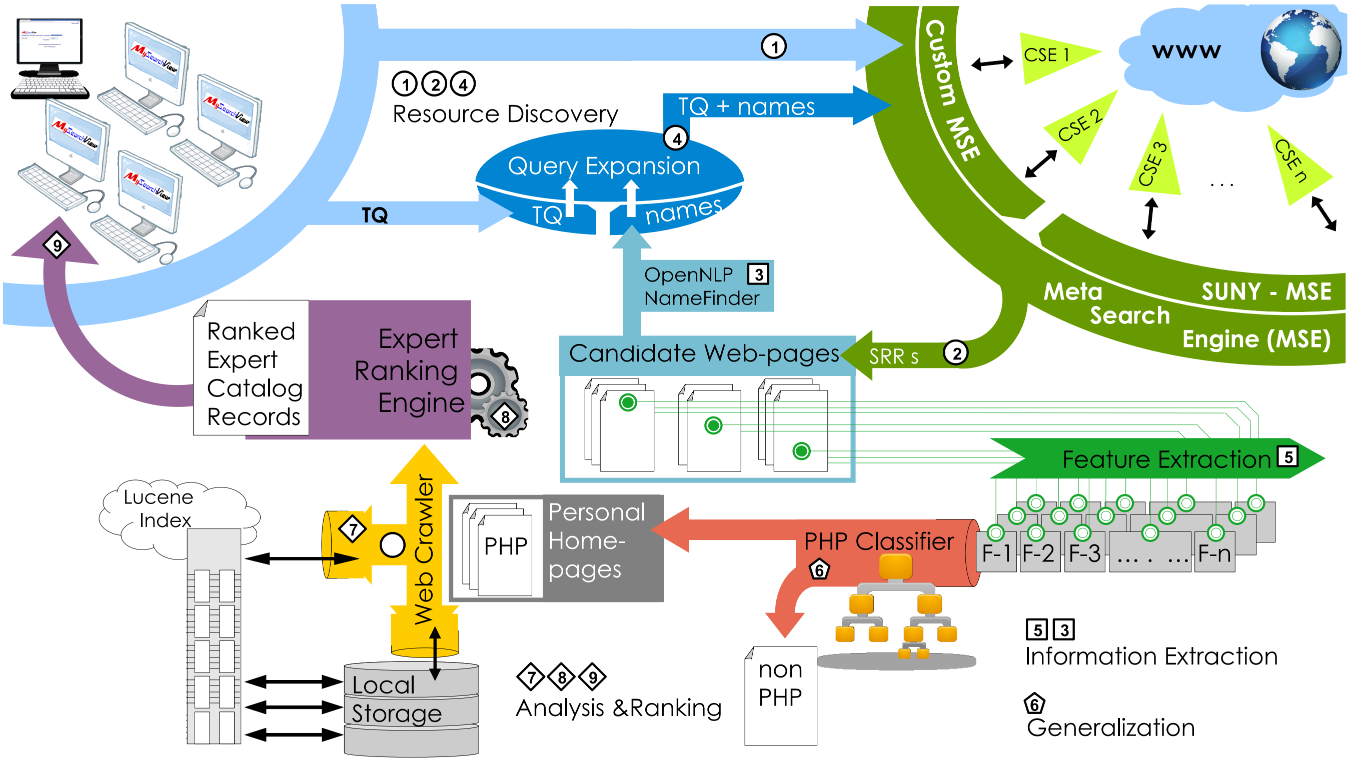


Figure 1: XM Architecture

This is the architecture that we propose for the XM task. XM when considered as a web mining activity[6], can be structured into following sub-tasks.

3.1 Resource discovery

Our primary goal in this sub-task is to find candidate web-pages mentioned in Figure 1. User submitted topic query (TQ)(likewise expanded TQs from next section) indicating an expertise or area of interest is passed on to underlying CSEs. SRRs and corresponding web-pages are collected from CSEs. Thus identifying our candidate web-pages.

3.2 Information extraction

Relevant feature information are scraped from web-pages returned from previous step. Features extracted include hyperlinks associated with each web-page as well as fetures mentioned in section 4. Thus establishing a foundation, before further performing generalization task as advised by[6]. Once names of few experts are recognized[17], we perform TQ exapansion[3] for web pages in concurrence with expert names. TQ are subjected to query expansion and looped back to Resource discovery stage for a modified and refined set of SRRs.

3.3 Generalization

Unlike section 3.1, we identify candidate PHPs in this sub-task. Firstly we identify PHPs (i.e PHPs belonging to experts) from results of SUNY-MSE for the given topic query. PHP Classifier is the integral part of this module. Secondly, we organize and index PHPs and associated documents so as to facilitate score calculation in sub-task 3.4. Thirdly we aggregate and summarize all the information extracted, by performing bibliometric analysis on the documents extracted from PHPs. Thus quantifying different aspects of expertise with scoring mechanisms.

[17] openNLP`http://incubator.apache.org/opennlp/`

Table 1: URL Syntax[R28], [R29], [R30] as referenced in our research

scheme://hostDomain . hostTLD/ pathUser / pathDir / document . extension ? query # fragment			
For e.g. with ref. to http://member.acm.org/˜robin.john/public/index.dtb?v=RvInzznrhYs			
scheme	: http	**hostDomain**	: member.acm
host Top Domain Leve(**hostTLD**)	: org	**pathuser**	: ∼robin.john
pathDir	: public	**document.extension**	: index.dtb

3.4 Analysis & Ranking

Scores calculated in the previous steps will be converged into expert ranks for current TQ. Some of the scoring mechanism we are experimenting with include TQ frequency, prestige associated with a PHP is calculated using inlinks using Yahoo! site explorer API and citation indexes. However, towards our first version of implementation we only use SRR ranks from CSEs to rank the PHPs identified in sub-task 3.3. This sub-task marks end of the entire XM process and returns a ranked list of Expert Catalog Records.

4. PHP CLASSIFIER FEATURE

We account for all features used by PHP classifier in this section. Firstly we organize features into tiers. Secondly, we present brief descriptions for all these features. Final track-able end point for the work presented, would be to create a classification model, referenced as PHP Classifier. Based on the feature vector associated with each web-page, PHP Classifier will be able to classify it into PHP or non-PHP as a data-mining task.

4.1 Feature Tiers

We review features used to achieve a functional classification[17] of web-page. These features are identified to be plausible representatives of a ”personal homepage” for current web-page classification activity.

Primarily, web-page's features are divided into two broad sections on-page features[17], one which is directly located on the page to be classified. Second, features of neighbors[17], which are found on in-link pages referencing or citing the to be classified web-page.

Features from on-page section are focused in our current research. However features from neighborhood pages will be extensively used in our future research. At next tier feature sections are broken into following sets: URL, Title, Email, Hyperlinks, Keywords and visually features. Finally all component features are enlisted in last tier.

We too specify these feature in a convention conformed by[18, 19]. Document, URL, Title, Email, Hyperlink, Images are the streams identified in our work. Features are summarized into tiers and conveyed as mathematical expressions in Tables 3 and 4.

4.2 Feature Extraction Utility Suite

Major components widely reused across our work are defined, using some standard representation in this section. Synsets are composes of semantically equivalent data elements, these data elements are used to create RegEx patterns which are in turn used to assess similarity. A collection of synsets are mentioned in table 2. Below mentioned are few utility functions, which are used for webpage feature vector calculation. All of these functions have a parameter corresponding to the stream(X) they are processing. Thus this function can extract and evaluate any stream identified in section 4.1. The usage and parameters for these functions are specified in Tables 3 and 4.

We use $c(\mathbb{S}, X)$ to determine the number of word in X, where $\mathbb{S}$ $corresponds(\widehat{=})$ sysnsets or regular expression (RegEx) identified in table **??** and $X \triangleq$ to one of the streams pointed out in section 4.1.

$nlp(M_p, X)$ is application of a Natural Language Processing functinality exposed by OpenNLP[18] for name finding, where $M_p \triangleq$ to a person name finder model[19] and again $X \triangleq$ to one of the streams

We rely on a HTMLParser[20] library to extract specific tag sections from a webpage using $hp(< tag >)$. `<title>, <img>, <a>` are some of the HTML tag fragments, that are extracted using HTMLParser in features mentioned ahead.

Levenshtein distance[16] $L(X_1, X_2)$ is calculated for comparing similarity between two strings $X_1 \& X_2$.

We use $wn(t_1t_2...t_n)$ to perform a lookup across WordNet[33] for a string $t_1t_2...t_n$ and it's each 2^n subsequences.

Haar Cascade classifier implemented in openCV[21] was used, as it has a proven ability to perform real-time(fast) face detection. Given an image i_1 from a webpage $hcc(i_1)$ can identify haar like features.

$abb(X_1, X_2)$ implements techniques mentioned in [21], where we search the web-page document stream X_2 for a full-form corresponding to acronyms recognized in stream X_1. For e.g. the website for IEEC binghamton, has title as "IEEC - home". Hence $abb(Title, Document)$ will search full form of the acronym IEEC from title stream in Document stream. In this scenario it will find the full form "Integrated Electronics Engineering Center"

Table 2: SynSet & RegEx

Set extensional definition
$\mathbb{T} = \{\sim, \%7e\}$
$\mathbb{A} = \{ \prime s, 's\}$
$\mathbb{H} = \{home, home * page\}$
$\mathbb{D} = \{department, lab, group, facility\}$
$\mathbb{N} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
$\mathbb{P} = \{publication, conference, seminar\}$
$\mathbb{F} = \{faculty, staff, people, member\}$
$\mathbb{L} = \{alumni, prospective * student, future\}$
$\mathbb{D} = \{admission\}$
$\mathbb{B} = \{about, news, events, mission\}$
$\mathbb{C} = \{contact * us\}$
$\mathbb{R} = \{research\}$
$\mathbb{A} = \{paper, article, journal, resume, report\}$
$\mathbb{I} = \{interest, work, current\}$
$\mathbb{O} = \{office * hour\}$
RegEx definition
$\mathbb{E} \triangleq$ RegEx pattern for email

4.3 Feature Definitions

Each feature is framed and expressed using a few characteristics. **Firstly**, a handle name is associated with each feature. **Secondly**, description, rationale supporting feature usage and stastical distribution across class labels[22]. **Thirdly**, the data type used to quantify this feature. **Fourthly**,

[18] `http://incubator.apache.org/opennlp/`

[19] OpenNLP en-ner-person.bin model

[20] `http://htmlparser.sourceforge.net/`

[21] `http://opencv.willowgarage.com/wiki/`

[22] PHP vs *non*PHP

scraper heuristics used to extract this feature into corresponding data type. **Finally**, a clarifying example. `Monospace (fixed-width) fontface` is used for feature name, since they signify names of classes/variables, HTML tag or snippets of code from our implementation. Most of these features are also supported with probability distribution across possible values observed for each feature in our data-sets.

4.3.1 "pathUser Length"

This feature corresponds to the length of `pathUser` sub-string of the URL in number od characters. `pathUser Lengths` are observed to be less than 10 characters on PHPs with probability 0.97 (refer figure:2). Since features values are dynamic values, an integer data type is used to quantify it. RegEx is used to extract `pathUser` sub-string. For instance for `URL`[23]; $|p| = 5$.

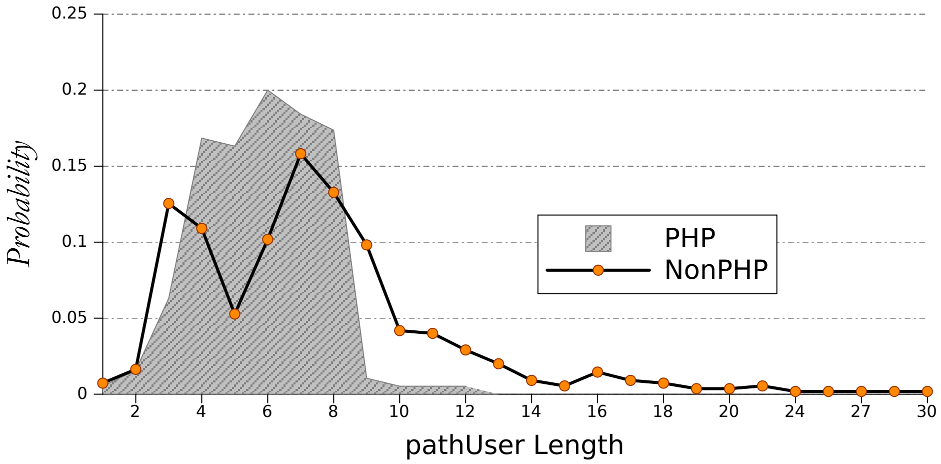


Figure 2: *probability distribution* PHP vs *non*PHP for "pathUser Length"

4.3.2 "Tilde in URL"

Tilde($\mathbb{T}$) often denotes a personal website on a Unix-based server. User-specific directories are accessed using for e.g. using URL syntax[24] `http://example.com/∼user/`. A Boolean data type is used to quantify this feature.

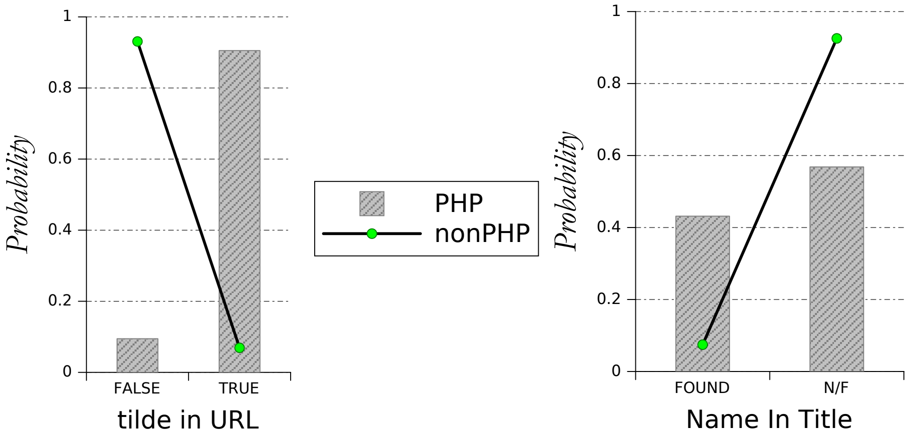


Figure 3: *probability distribution* PHP vs *non*PHP for "Tilde in URL" & "Name in Title"

4.3.3 "Name in TITLE"

[23] `http://www.cs.binghamton.edu/˜meng/`

[24] Apache HTTP Server mod_userdir Documentation

This feature marks presence of a proper name in the title, since experts tend to mention their names in a PHP's title section. This feature too is defined by a boolean data-type. OpenNLP is used to find a proper name within the `<title>` tag stream in webpage's HTML code. For instance a title tag[25] will have `Name in TITLE = "FOUND"`

4.3.4 ‘s in TITLE

feature signifies if title string has ‘s. Apostrophe before s shows singular possession[26]. In this scenario ‘s signifies posession of home page. Again Boolean data type is used. 90% of all web-pages with this feature are identified to be PHP's. For instance w.r.t.`<title>`tag mentioned in feature section 4.3.3. `'s in TITLE" = "FOUND"`. Figure 4

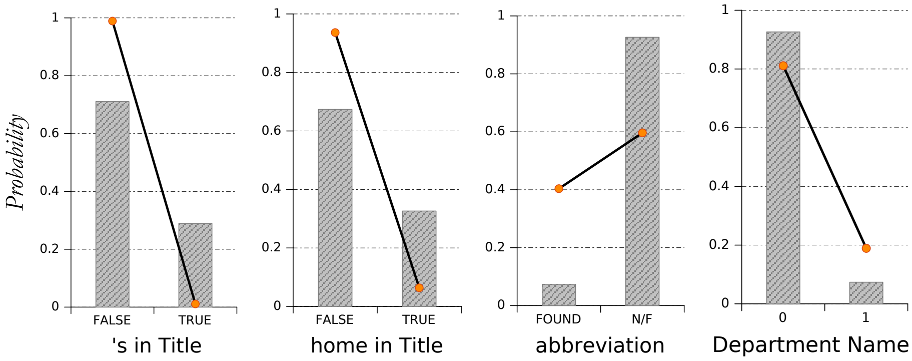


Figure 4: *probability distribution* PHP vs *non*PHP for 's, home & abbreviation in Title

4.3.5 "Home in TITLE"

Keyword "home" or words from synset ($\mathbb{H}$) could be found in title string, since personal home pages explicitly or implicitly reference to home as domicile in describing the home page[14]. Again w.r.t. `<title>`tag mentioned in feature section 4.3.3. `home in TITLE = "FOUND"`. This feature is annexed with synset ($\mathbb{H}$). 65% of web-pages with a positve score are PHPs.

4.3.6 "Abbreviation in TITLE"

This feature denotes weather a valid abbreviation could be found in `<title>` tag. This will assist in filtering out departmental or organizational web-pages, which could have acronyms and PHPs very unlikely will have abbreviations. 95% of the times when an abbreviation is found in web-page it is identified to be nonPHP. Techniques mentioned in[21] were implemented, where we search the web-page for a full-form corresponding to an acronym found in `<title>`tag[27]. `abbreviation in TITLE ="FOUND"`

4.3.7 "Department name in TITLE"

This feature counts occurrences of any of the signifying word phrases associated with synset ($\mathbb{D}$) in `<title>`tag. This feature too acts as good nonPHP indicator. Web-pages with $\mathbb{D}$ synset keywords in title, could be of departmental, organizational, conference type with a probability of 0.72. An integer data type is used. Illustrated by `<title>`tag[28] `Department Name in TITLE = 1`

4.3.8 "email"

[25] `<title>Weiyi Meng's Home page</title>`

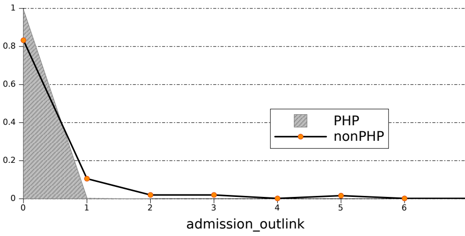
[26] GrammarBook

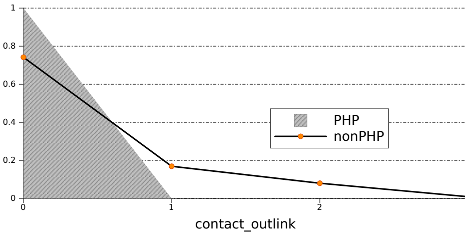
[27] <title>IEEC - Home</title>

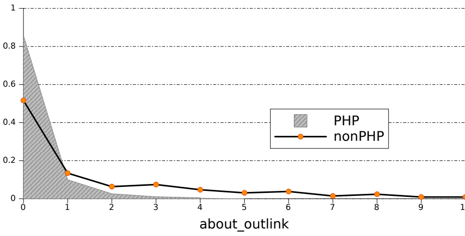
[28] <title>Chemistry Department at Stony Brook</title>

Table 3: Feature Tiers URL & Title

Feature	Expression	Tier2	Teir1
4.3.1 pathUser Length	$\lvert\text{pathUser}\rvert$	**URL**	**On-Page Feature**
4.3.13 pathUser-wn	$wn(pathUser)$		
4.3.14 pathUser numeric characters	$c(\mathbb{N}, pathUser)$		
4.3.2Tilde in URL	$= \begin{cases} 1 & \text{if } c(\mathbb{T}, \text{URL}) > 0 \\ 0 & \text{else} \end{cases}$		
4.3.3Name in TITLE	$= \begin{cases} 1 & \text{if } nlp(M_p, \text{Title}) > 0 \\ 0 & \text{else} \end{cases}$	**Title**	
4.3.4 's in TITLE	$= \begin{cases} 1 & \text{if } c(\mathbb{T}, \text{Title}) > 0 \\ 0 & \text{else} \end{cases}$		
4.3.5home in TITLE	$= \begin{cases} 1 & \text{if } c(\mathbb{H}, \text{Title}) > 0 \\ 0 & \text{else} \end{cases}$		
4.3.6abbreviation in TITLE	$abb(Title, Document)$		
4.3.7 Department Name in TITLE	$= \begin{cases} 1 & \text{if } c(\mathbb{D}, \text{Title}) > 0 \\ 0 & \text{else} \end{cases}$		







This feature marks presence of an email address in the web-page. RegEx are used to find such a pattern, which is recorded as a boolean data type. email syntax as referenced `email_User_ID@email_domai` Email munging discussed in section 5.3.3 will hamper this feature's extraction.

4.3.9 "email_User_ID"

This feature is associated with email feature, it denotes if a user-id could be found in the email. w.r.t. email[29] found in webpage[30] `email_User_ID = abwagner`

4.3.10 "email_domain"

This feature too is associated with email feature, it denotes if an email_domain was found in the email. user_id and email_domain are introduced purely to validate the feature selection algorithms. As they have no more information gain than "email" feature. Similarly w.r.t email[29] `email_domain = buffalo.edu`. However features 4.3.9 and 4.3.10 have the same information gain as 4.3.8. These feature are introduced just as a experiment basis for ReliffF feature ranking in section 5.2. We ingore these two feature during our final PHP Classifier generation.

4.3.11 "pathUser vs email_User_ID"

A prospective pattern can be observed in PHPs from academic backgrounds. Web-pages with similar$(L(pathUser, email_User_ID) < 3)$ pathUser and email_User_ID have a 0.92 probability to be PHPs. For e.g. w.r.t URL30 and email29 where `email_User_ID = abwagner; pathUser = abwagner;` hence `pathUser vs email_User_ID value = 1`.

[29] abwagner@buffalo.edu

[30] `www.acsu.buffalo.edu/~abwagner/`

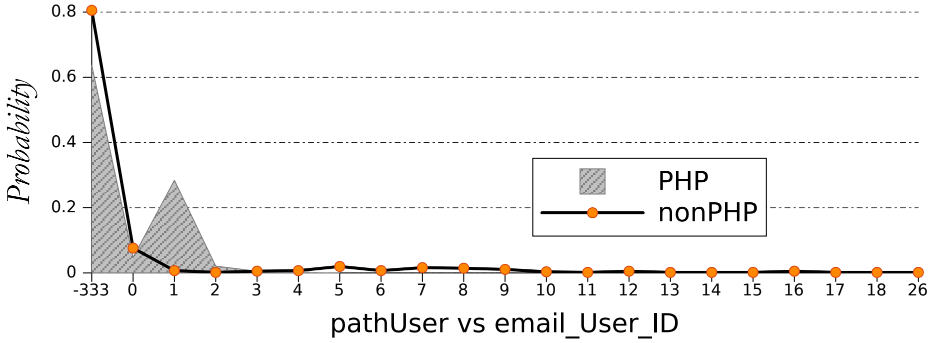


Figure 5: *probability distribution* PHP vs *non*PHP for "pathUser vs email_User_ID"

4.3.12 "host_domain vs email_domain"

Similarily we include host_domains and email_domains have a fair likelihood(75%) of being similar($L(host\_domain, email\_domain)$ <5), thus having less or close to zero distance $L(X_1, X_2)$. For instance w.r.t. URL[30] and email[29] where `host_domain = acsu. buffalo.edu; email_domain = buffalo.edu;` hence `host_domain vs email_domain = 5`.

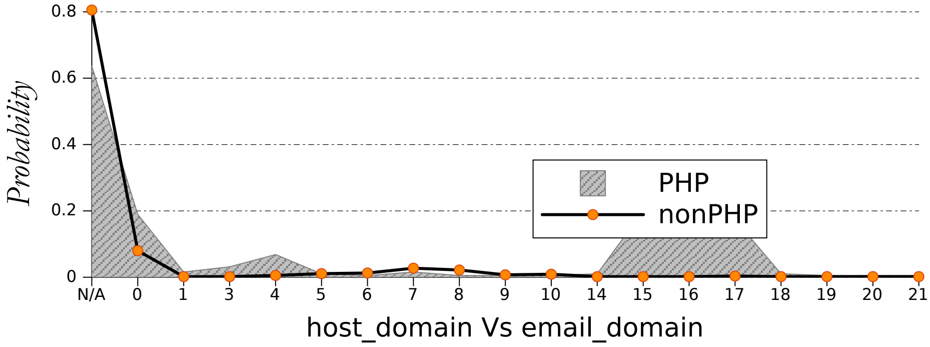


Figure 6: *probability distribution* PHP vs *non*PHP for "host_domain vs email_domain"

4.3.13 "pathUser_wn"

PHPs are observed to have `pathUser` section of the URL composed of user-name for ex. `abwagner, meng, rvarghe1` i.e non dictionary terms. Since, presence of dictionary terms usually signifies sub-domain, directory description which in-turn indicated a nonPHPs. This feature is ranked second in our feature ranking mentioned in table 5.1. A boolean marks if `pathUser` encomposes a dictionary term. All sub-strings of `pathUser` are checked, if present in WordNet[31]. For e.g. w.r.t URL[32] `pathUser = programsearch` constitutes of two sub-strings that can be found in WordNet.

[31] http://wordnet.princeton.edu/

[32] http://www.suny.edu/programsearch/

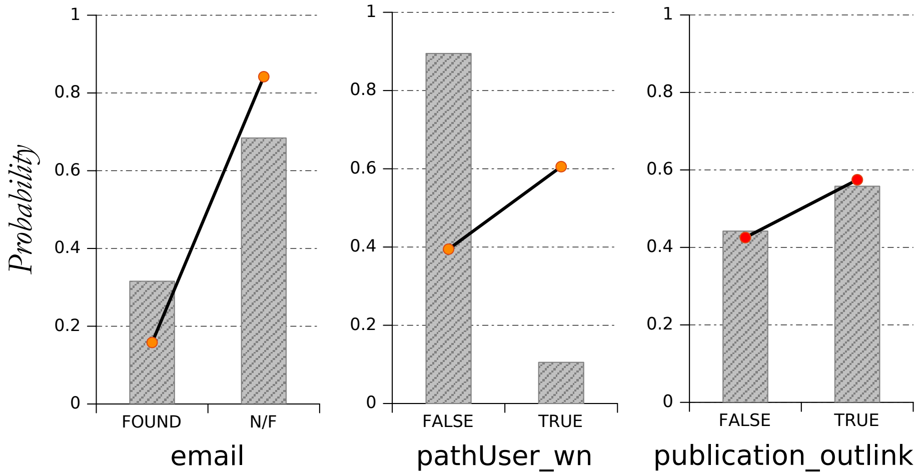


Figure 7: ***probability distribution* PHP vs *non*PHP for email, pathUser_wn & publication_outlink**

4.3.14 "pathUser numeric character"

This feature signifies numbers in `pathUser`. This feature is utilized to filter out departmental course websites. RegEx is used to count such an occurrence with an integer data-type. For ex. URL: [33] `pathUser = cs432` hence feature value = 3

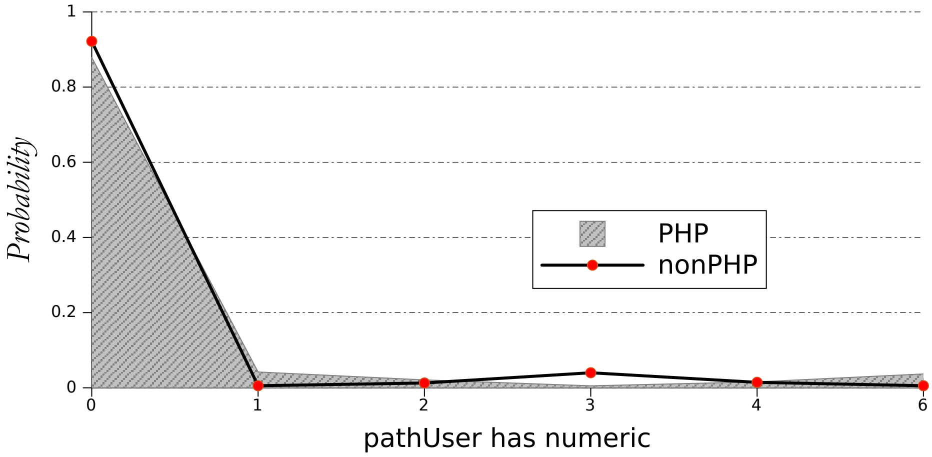


Figure 8: ***probability distribution* PHP vs *non*PHP for ”pathUser has numeric”**

4.3.15 "publication_outlink"

Experts usually provide a hyperlink with anchor phrase publication(publication, biography, etc.) with an intention, that visitors can easily find their list of publications. This feature helps us to narrow down to web-pages with research orientation. A boolean marks if a hyperlink with publication as a keyword from synsets exists on web-page. Similar to `publication_outlink`, below mentioned features too have author's intent and outlinks associated with each of them.

97% of all the pages with `faculty_outlink` are Departmental pages, pointing to their faculty list.

`alumni_outlink` from Web - pages providing resource for their alumni are nonPHP(University pages) in 98% instances.

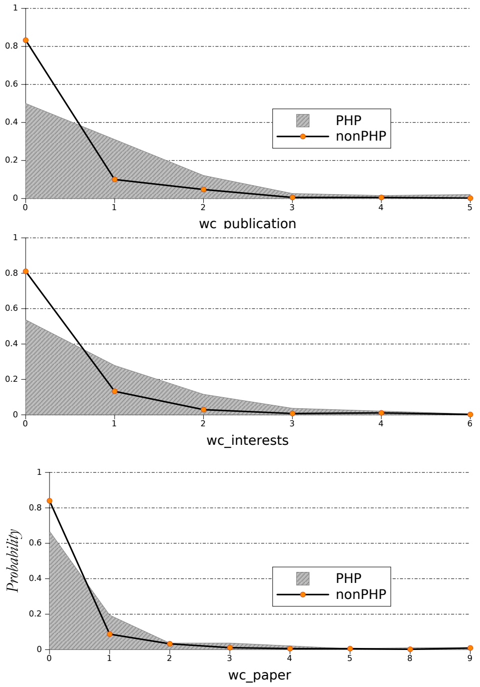
University pages provide `admission_outlink` as resource for their prospective students.

`about_outlink` from any organization pointing to it's ”About Us” web-page

[33] `http://www.cs.virginia.edu/cs432/`

Table 4: Feature Tiers

Feature	Expression	Tier2	Teir1
4.3.8email	$c(\mathbb{E}, Document)$	**Email**	**On-Page Feature**
4.3.9email-User-ID			
4.3.10email-domain			
host-domain 4.3.12Vs email-domain	$L(host\_domain, email\_domain)$	**Email URL**	
pathUser 4.3.11Vs email-User-ID	$L(pathUser, email - User - ID)$		
4.3.15publication-outlink	$c(\mathbb{P}, hc("a"))$	**Hyperlink**	
4.3.15 faculty-outlink	$c(\mathbb{F}, hc("a"))$		
4.3.15 alumni-outlink	$c(\mathbb{L}, hc("a"))$		
4.3.15 admission-outlink	$c(\mathbb{D}, hc("a"))$		
4.3.15 about-outlink	$c(\mathbb{B}, hc("a"))$		
4.3.15 contact-outlink	$c(\mathbb{C}, hc("a"))$		
4.3.16wc-publication	$c(\mathbb{P}, Document)$	**Keyword**	
4.3.16 wc-research	$c(\mathbb{R}, Document)$		
4.3.16 wc-paper	$c(\mathbb{A}, Document)$		
4.3.16wc-interests	$c(\mathbb{I}, Document)$		
4.3.16wc-office-hour	$c(\mathbb{O}, Document)$		
Number 4.3.17of Images	$\lvert hp("img") :$ image size > 200 pixels$\rvert$	**Graphical**	
Number 4.3.18of Faces	$\lvert hcc(hp("img"))\rvert$		



All the pages with `contact_outlink` are organization pointing to it's contact information.

Above mentioned features are represented by integers, expressing number of times any of the associated synset keywords appear as anchor-text(outlinks) in the web-page. Web-pages with anchor-text keywords such as faculty, alumni, admission, about and contact tend not to be PHPs.

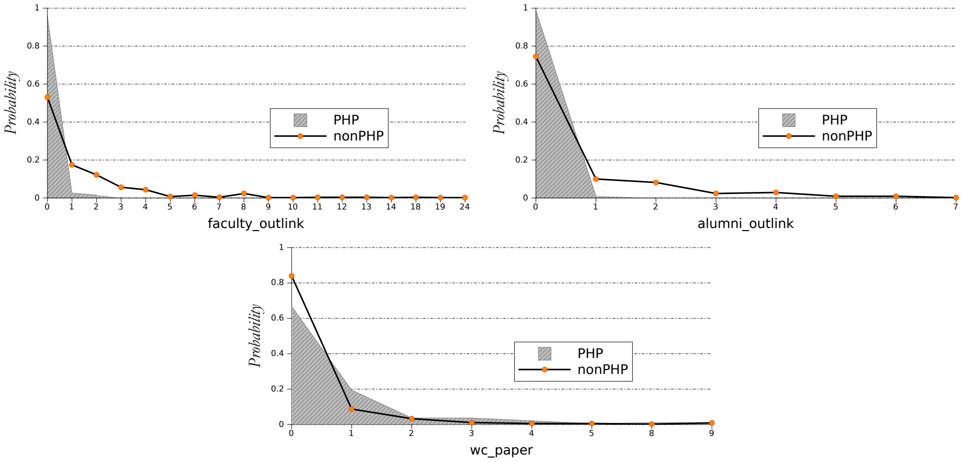


Figure 9: ***probability distribution*** **PHP vs** ***non*****PHP for faculty & alumni_outlink**

4.3.16 Keyword / Bag of words

PHPs in academic circles have a high term frequency of a few keywords like publication, paper, interests, phrase "office hour", etc. Each feature has a synset mentioned in section**??** associated with them. Term frequency of corresponding keywords in the web-page are accumulated into an integer data type. Similarly following features too are computed as per their keyword pool coupled with `wc_research`, `wc_paper`, `wc_interests`, `wc_publication` and `wc_office_hour`. Finally we support these words with term frequency-inverse document frequency(tf-idf) weight for PHP vs nonPHP 1-grams and 2-grams.(still working on this .....)

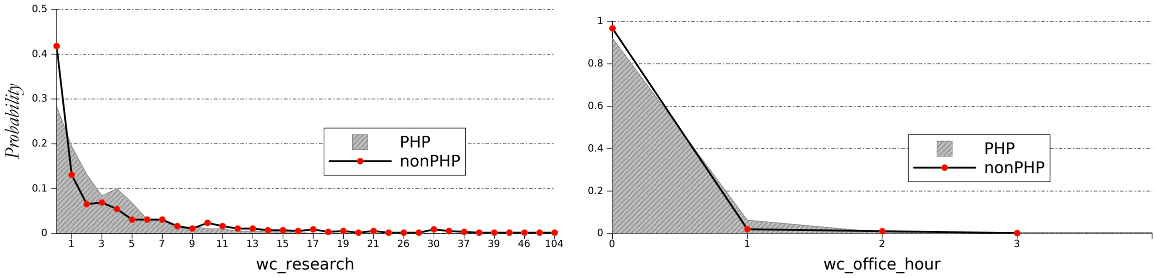


Figure 10: ***probability distribution*** **PHP vs** ***non*****PHP word count for research & office_hour**

4.3.17 "Number of Images"

Feature denotes number of `<IMG>` tags in web-page's HTML code. `<IMG>` tags are extracted from a given web-pages. The images are downloaded to local memory to be further processed by the next feature. This also helps us to identify the ratio of total images to images with facial features. We ignore images that are less than 200 pixels.

4.3.18 "Number of Faces"

Experts usually add their pictures to their PHP. Feature denotes the number of human faces found on the web-page. Each of the images accounted in previous feature is processed, to identify facial parameters. This feature will serve us to identify PHP characteristic in 45.7% of PHPs which have facial images.

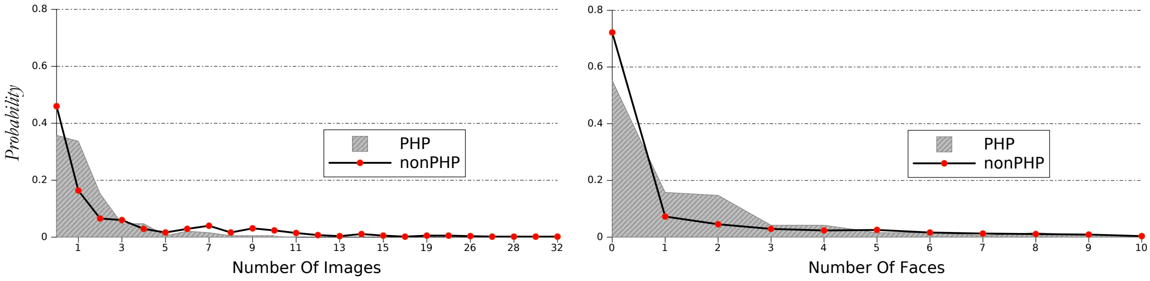


Figure 11: ***probability distribution*** **PHP vs** ***non*****PHP word count for research & office_hour**

5. TESTING

Implementation includes two modules, one to fetch candidate webpages and other to classify web-pages to PHP. For the purpose of evaluation we perform focused search on four SUNY University centers (Binghamton, Albany, Stony Brook & Buffalo). We assemble data-sets, a portion(27%) of which is used to train the PHP classifier. Finally we evaluate the models with remaining portion of the data-set.

Table 6: Model selection & evaluation measures

Measures	Naïve Bayes	SMO	RBF Network	J48/C4.5	ADTree	ADTree URL	ADTree Title	ADTree Keyword	PHP Classifier
Correctly Classified Instances	88.57%	96.07%	88.21%	97.14%	96.43%	95.36%	90.71%	89.64%	93.92%
Incorrectly Classified Instances	11.43%	3.93%	11.79%	2.86%	3.57%	4.64%	9.29%	10.36%	6.08%
Kappa statistic	0.5812	0.8182	0.3765	0.8626	0.8326	0.7851	0.3509	0.4006	0.8452
Mean absolute error	0.1412	0.0393	0.1598	0.0549	0.0716	0.1324	0.2653	0.2243	0.0996
Root mean squared error	0.3199	0.1982	0.2895	0.1682	0.173	0.2295	0.3166	0.3118	0.2096
Total Number of Instances	280	280	280	280	280	280	280	280	740
TP Rate	0.886	0.961	0.882	0.971	0.964	0.954	0.907	0.896	0.939
FP Rate	0.143	0.132	0.549	0.131	0.132	0.158	0.672	0.572	0.069
Precision	0.922	0.961	0.869	0.971	0.964	0.954	0.916	0.881	0.942
Recall	0.886	0.961	0.882	0.971	0.964	0.954	0.907	0.896	0.939
F-Measure	0.897	0.961	0.874	0.971	0.964	0.954	0.881	0.883	0.94
ROC Area	0.928	0.914	0.834	0.87	0.973	0.9	0.695	0.749	0.983

Table 5: Sample Topic Queries

xml parser, virtualization, data mining, metasearch engine
database systems, oncology, fundamentalism, foreign policy,
neurons, Severe Acute Respiratory Syndrome,
childhood depression, micro electro mechanical systems,
Polymer, Face Expression Analysis, Game Theory, terrorism,
solar energy, Psychological, Ecological Society, geophysics,
health literacy, music Composition, transmission,
electron microscopy, environmental issue

5.1 Test Bench Setup

We use Google, Bing and Yahoo's advanced search options to build three CSEs to perform focused searches for resource discovery as mentioned in section 3.1.

We compiled a collection of topic queries which are research topics, academic disciplines or interdisciplinary scientific fields. We extract three pages(i.e 20-30 uniques SRRs) from each of CSEs for each topic query. These topics queries and associated SRRs are again grouped together into 5 pools. Where each pool acts as a Data set $T_1 - T_5$. A Data-set constitutes of instances, where each instance is a web-page(i.e URL). A bag of "concepts" is associated with each data-set corresponding to topic queries, which were used to retrieve the web-pages instances composing the data-set. URLs in the Data-sets are manually labelled into PHP or nonPHP. At this step we also filter out any web-pages which result in a `Page not Found` error.

All features defined in section 4.3 are extracted from each webpage, thus achieving information extraction as mentioned in section 3.2.

5.2 Model Selection

The task at hand in this section, firstly is to generate different models, based on various data-mining algorithms (classifiers predictions are not calibrated - they are the raw model predictions) Secondly, perform empirical tests to compare and evaluate goodness of these classifier, using measures mentioned in table 6, Receiver Operating Characteristic (ROC) & Precision Recall plot.

Table 7: Ranked Features

Rank	Feature	Section
0.73143	Tilde in URL	4.3.2
0.31321	pathUser-wn	4.3.13
0.23071	Name in Title	4.3.3
0.22464	publication-outlink	4.3.15
0.16571	Home in TITLE	4.3.5
0.11036	‘s in TITLE	4.3.4
0.09995	host-domain vs email-domain	4.3.12
0.09931	pathUser vs email-User-ID	4.3.11
0.09143	Department name in TITLE	4.3.7
0.06875	wc_publication	4.3.16
0.05631	wc_paper	4.3.16
0.05321	user-id	4.3.9
0.05321	email	4.3.8
0.05321	email-domain	4.3.10
0.05071	wc_interests	4.3.16
0.04138	pathUser Length	4.3.1
0.03107	contact-outlink	4.3.15
0.02917	pathUser numeric character	4.3.14
0.02335	Number of Images	4.3.17
0.01872	wc_research	4.3.16
0.00866	alumni_outlink	4.3.15
0.00734	faculty_outlink	4.3.15
0.00714	Abbreviation in TITLE	4.3.6
0.00397	about_outlink	4.3.15
0.00393	wc_office_hour	4.3.16
0.00241	admission_outlink	4.3.15
0.00235	Number of Faces	4.3.18

Thirdly, evaluate goodness of our features. Finally, determine a suitable classifier and its corresponding optimal tuning parameters.

We have chosen 5 different classification model generation algorithms for evaluation. Alternating Decision tree (ADTree) , Sequential Minimal Optimization (SMO) , C4.5(J48), RBF network(RBF) and Naïve Bayes(NB). For each classifier mentioned before we use data-set T1 for training and use the rest of the data-sets as a large final test set in evaluation section. After a 10-fold cross validation on T1 data-set we obtain 5 classifier models. Weka[9] is utilized for all our data-mining needs(model generation, evaluation etc.).

ADtree classifier leads all the other alogorithms in ROC(figure 13) as well as in recall & precision space(figure 14).The nature of this case is such that feature values are highly skewed and kurtosis, along with presence of binary variables with high correlation between the attributes. Such properties are hypothesized[12] to favor tree-based classification models over Statistical Regression.

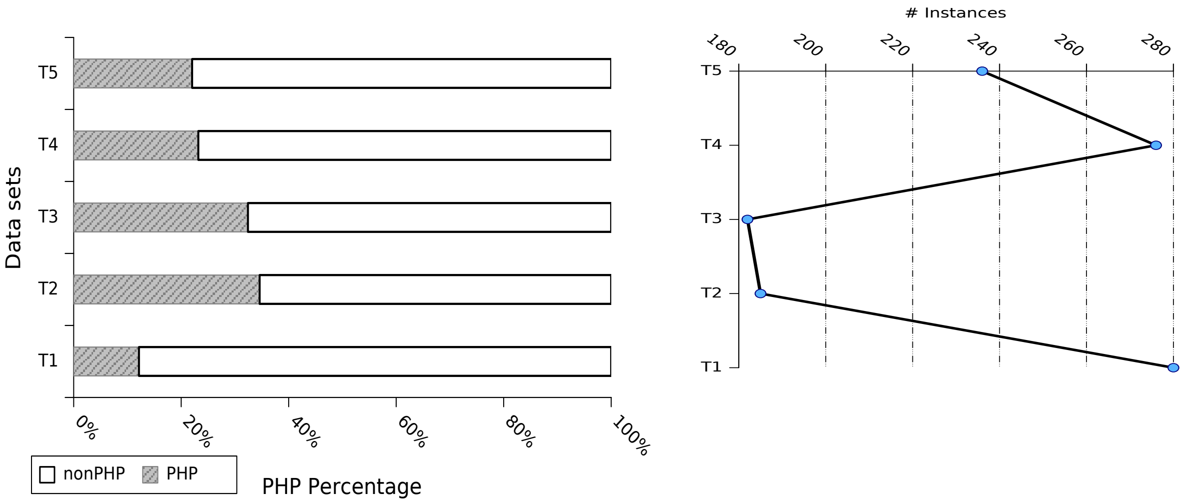


Figure 12: Data-set PHP:*non*PHP mix, number of instances

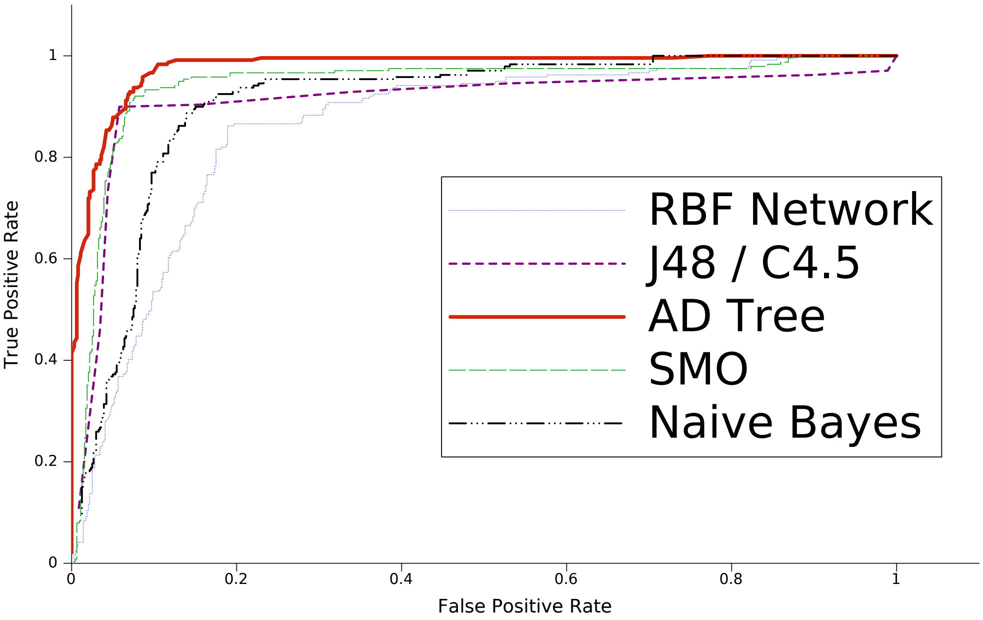


Figure 13: ROC curve

We employ ReliefF[20] feature weighting alogithm to evaluate relevance of our feature in table 5.1. Next we select three feature subsets URL, Title and synset centric. URL and Title subsets correspond to tier 2 features mentioned in table 3. Synset centric feature set includes keyword & hyperlink features from table 4, alongwith `name & 's in title` features. Lastly we generate 3 ADTree established models for each feature set. Evaluation metrics for these 3 models are listed in table 6.

Inline with our feature ranking, URL based features perform best. Finally, we generate our PHP classifier model comprising of all the features with the exception of `user-id` and `email-domain`. Since, they share the same rank and information (gain) as `email`.

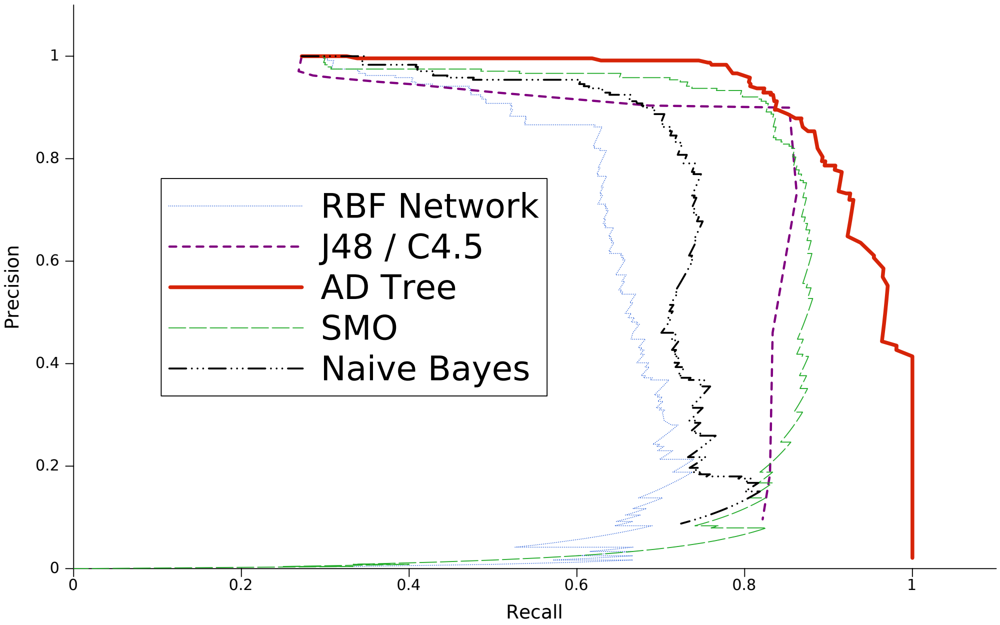


Figure 14: ROC curve

5.3 Evaluation

As per our deduction from 5.2, we select ADTree algorithm for model generation. We now evaluate PHP classifier model selected in section 5.2, with test-set comprising of data-set T2 through T5. Evaluation measures are again mentioned in table 6. Futhermore we scrutinize 49 false classification (FC)s made by PHP classifier. Towards this firstly, we identify all reasons for FC. Secondly, since multiple reasons play a role in FC, we visually represent all erroneously classified instances using a venn diagram.

5.3.1 NLP based

Incorrectly tagging title section of the web-page as person names, is observed to be the primary reasons for FCs. NLP techniques have limitaions in extacting person names in 4 cases. On the contrary in 22 instances have false positive recognitions. 5 of these instances are observed to have a blank title.

5.3.2 Near Default Feature Vector

8 web pages have very few traceable features. Hence the feature extraction module returns a near default feature vector for these pages.

5.3.3 Email munging

4 false negatives are influenced by email munging, hence all email features fall back to deafault values.

5.3.4 Graphic hyperlinks

Few web pages have hyperlinks embedded around `<IMG>` tags. However this can be resolved to some extent if we include in our scope to parse the `alt` attribute. Thus they do not have any extractable hypertext associated with them.

5.3.5 Unique email id

Also few non PHPs mention a single contact email id, which is extracted by our email features. These are scenarios where a faculty member hosts Departmental Pages(DP) in personal web spaces. Since these pages don't hold any information about a person, they are labelled as nonPHPs during our supervised labelling process. Co-occurance of unique email id and DP on faculty website result in 7 FCs.

5.3.6 *Default sitemap*

5 faculty websites are observed to have the default sitemap page, however these sitemap pages have outlinks to publication information, `Tilde in URL & pathUser-wn` features. Since these pages are not specifically intended as PHPs, we classify them as non PHPs in supervised labelling.

5.3.7 *Default Vector*

3 pages were found to be either redirecting or resulting in a `Page Not Found` error during our test runs. However, these pages we identifed to be performing normally during our labelling stage.

5.3.8 *False Classification Visualization*

It is clear from figure 15 that majority of FCs are false positves. Issues like unique email id and DP on faculty website are noted to co-occur. NLP based limitations are observed to be the sole issue in 10 instances. We were unable to discover the reasons for 8 false positives and 4 false negatives.

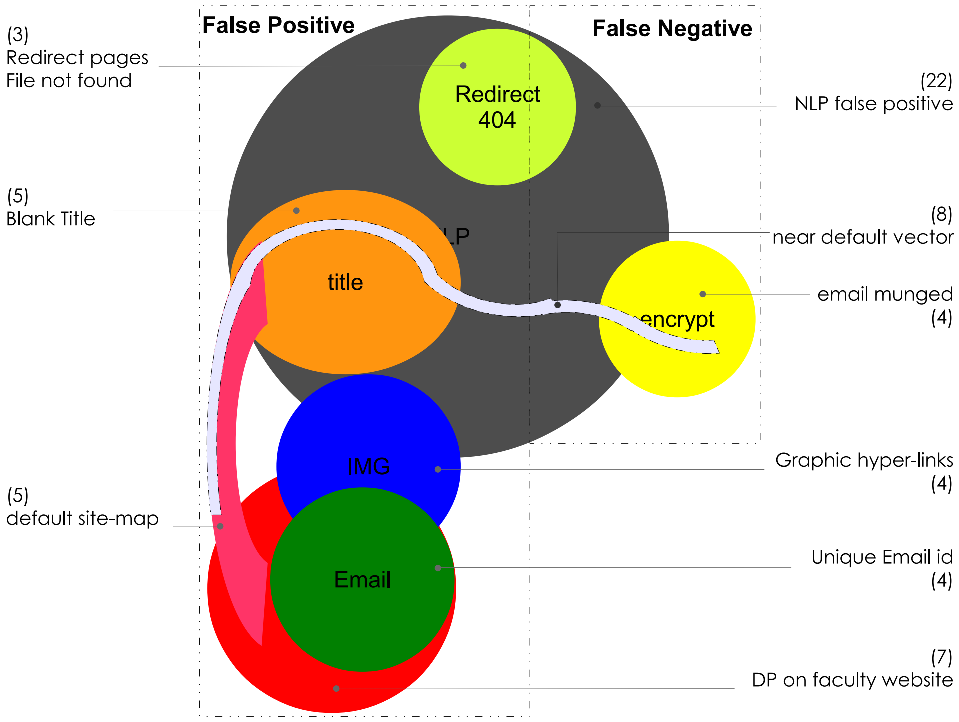


Figure 15: False Classification Visualization

6. CONCLUSIONS & FUTURE DIRECTIONS

PHP classifier model developed does achieve promising results, although there appears to be room for improvements.

Two sections need to be addressed, firstly improvements to current PHP Classifier. Secondly, implementation to the entire XM process. We have intentions to substitute low information valued features in PHP classifier with new features, for e.g.$c(\mathbb{I}, X)$ where $\mathbb{I}$ is synset with first and second person pronouns. A critical limitation of our solution is, we are unable to find experts whose PHPs have not been crawled and indexed by any of our underlying CSEs. A plausible solution would be to incorporate a few more features in our existing PHP classifier. This in-turn will equip our classifier to make multi-class classification into PHP, Departmental Home-Page (DHP) and Ineligible web-page. DHPs may have out-links to faculty pages which have not been crawled before by our CSEs. As from a perceptive of proof of concept, we did implement the entire XM process. Thus, identifying integral modules such are expert ranking logic, biblometric analysis and result caching.

7. ACKNOWLEDGMENTS

I am deeply indebted to Dr. Weiyi Meng for their invaluable guidance, insight, and support in the preparation of this study. I would also like to thank all my professors who inspired me during my master's program, my family for providing me a chance to do Masters of Science and all my friends, who supported and helped me during my Master's degree.

I express our thanks to the open source community for their valuable contributions as programming APIs.

8. REFERENCES

[1] F. C. A. Doan, R. Ramakrishnan. Community information management. *IEEE Data Engineering Bulletin, Special Issue on Probabilistic Databases, 29(1)*, December 2006.

[2] A. Abecker and L. Elst. Ontologies for knowledge management. In P. Bernus, J. Blazewics, G. Schmidt, M. Shaw, S. Staab, and R. Studer, editors, *Handbook on Ontologies*, International Handbooks on Information Systems, pages 713–734. Springer Berlin Heidelberg, 2009.

[3] K. Balog, E. Meij, and M. de Rijke. Language models for enterprise search: Query expansion and combination of evidence. In *IN THE FIFTEENTH TEXT RETRIEVAL CONFERENCE (TREC 2006). NIST*, 2007.

[4] N. Craswell, D. Hawking, A.-M. Vercoustre, and P. Wilkins. P@noptic expert: Searching for experts not just for documents. In *In Ausweb*, pages 21–25, 2001.

[5] H. Deng, I. King, and M. R. Lyu. Formal Models for Expert Finding on DBLP Bibliography Data. *Data Mining, IEEE International Conference on*, 0:163–172, 2008.

[6] O. Etzioni. The world wide web: quagmire or gold mine? *Communications of the ACM*, 39:65–68, 1996.

[7] Y. Fang, L. Si, and A. Mathur. Facfinder: Search for expertise in academic institutions. 2008.

[8] Y. Fang, L. Si, and A. P. Mathur. Discriminative graphical models for faculty homepage discovery. *Inf. Retr.*, 13:618–635, December 2010.

[9] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11:10–18, November 2009.

[10] M. T. Hansen, N. Nohria, and T. Tierney. What's your strategy for managing knowledge? *Harvard Business Review*, 77(2), 1999.

[11] O. Khn and A. Abecker. Corporate memories for knowledge management in industrial practice: Prospects and challenges. *Journal of Universal Computer Science*, 3(8):929–954, Aug. 1997.

[12] R. D. King, C. Feng, and A. Sutherland. Statlog: Comparison of classification algorithms on large real-world problems. *Applied Artificial Intelligence*, 9(3):289–333, 1995.

[13] J. Li, H. Boley, V. C. Bhavsar, and J. Mei. Expert finding for ecollaboration using foaf with ruleml rules. In *Proc. of the 2006 Montreal conference on eTechnologies*, pages 53–65, 2006.

[14] S. L. Marcia J. Bates. An exploratory profile of personal home pages: Content, design, metaphors. *Online Information Review*, 21:331–340, February 1997.

[15] W. Meng, C. T. Yu, and K. L. Liu. Building efficient and effective metasearch engines. *ACM Computing Surveys*, 34(1):48–89, 2002.

[16] G. Navarro. A Guided Tour to Approximate String Matching. *ACM Computing Surveys*, 33, 1999.

[17] X. Qi and B. D. Davison. Web page classification: Features and algorithms. *ACM Comput. Surv.*, 41:12:1–12:31, February 2009.

[18] T. Qin, T. yan Liu, J. Xu, and H. Li. Letor: A benchmark collection for research on learning to rank for information retrieval.

[19] S. Robertson, H. Zaragoza, and M. Taylor. Simple bm25 extension to multiple weighted fields. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, CIKM '04, pages 42–49, New York, NY, USA, 2004. ACM.

[20] M. Robnik-Sikonja and I. Kononenko. An adaptation of Relief for attribute estimation in regression, 1997.

[21] A. S. Schwartz and M. A. Hearst. A simple algorithm for identifying abbreviation definitions in biomedical text. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pages 451–462, 2003.

[22] J. Swan, S. Newell, H. Scarbrough, and D. Hislop. Knowledge management and innovation: networks and networking. *Journal of Knowledge Management*, 3(4):262–275, 1999.

[23] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. Arnetminer: extraction and mining of academic social networks. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 990–998, New York, NY, USA, 2008. ACM.