Author Homepage Discovery in CiteSeerX

Krutarth Patel,[1] **Cornelia Caragea,** [2] **Doina Caragea,** [1] **C. Lee Giles** [3]
[1] Computer Science, Kansas State University
[2] Computer Science, University of Illinois at Chicago
[3] Information Sciences and Technology, Pennsylvania State University
kipatel@ksu.edu, cornelia@uic.edu, dcaragea@ksu.edu, giles@ist.psu.edu

Abstract

Scholarly digital libraries provide access to scientific publications and comprise useful resources for researchers. CiteSeerX is one such digital library search engine that provides access to more than 10 million academic documents. We propose a novel search-driven approach to build and maintain a large collection of homepages that can be used as seed URLs in any digital library including CiteSeerX to crawl scientific documents. Precisely, we integrate Web search and classification in a unified approach to discover new homepages: first, we use publicly-available author names and research paper titles as queries to a Web search engine to find relevant content, and then we identify the correct homepages from the search results using a powerful deep learning classifier based on Convolutional Neural Networks. Moreover, we use *Self-Training* in order to reduce the labeling effort and to utilize the unlabeled data to train the efficient researcher homepage classifier. Our experiments on a large scale dataset highlight the effectiveness of our approach, and position Web search as an effective method for acquiring authors' homepages. We show the development and deployment of the proposed approach in CiteSeerX and the maintenance requirements.

Introduction

CiteSeerX is a scientific literature digital library and search engine that focuses primarily on the literature in computer and information science. CiteSeerX aims to improve the dissemination of scientific literature and to provide improvements in functionality, usability, availability, cost, comprehensiveness, efficiency, and timeliness in the access of scientific and scholarly knowledge. Besides search capabilities, CiteSeerX provides access to over 10 million academic documents (metadata information along with the full documents), which have been used in many applications such as expert search (Balog and De Rijke 2007), author name disambiguation (Kim et al. 2018; Khabsa, Treeratpituk, and Giles 2014), keyphrase extraction (Patel et al. 2020; Patel and Caragea 2019; Alzaidy, Caragea, and Giles 2019; Gollapalli and Caragea 2014; Caragea et al. 2014; Florescu and Caragea 2017), citation indexing (Giles, Bollacker, and Lawrence 1998), topic classification (Lu and Getoor 2003; Caragea, Bulgarov, and Mihalcea 2015), and collaboration network formation (Chen et al. 2011).

In order to enlarge its document collection, CiteSeerX maintains whitelists / blacklists of URLs and lists of researchers' homepages to direct the crawl for documents. Thus, maintaining comprehensive, up-to-date collections of researchers' homepages is an essential component in CiteSeerX. However, this task is very challenging since not only do new authors emerge, but also existing authors may stop publishing or may change affiliations, resulting in outdated ($4XX$ error) or invalid URLs. An analysis of a subset of $13,239$ homepages available in DBLP revealed that $\approx 42\%$ of them were outdated within a time span of three years. This represents about half of the original homepages in the set. *Given this challenge, how can we automatically augment and maintain accurate lists of researchers' homepages?*

One approach would be to crawl academic websites (e.g., from a university domain) and use a machine learning classifier to predict whether a website accessed during the crawl is an author homepage or not (Gollapalli et al. 2013). However, this approach: (1) is still inefficient (e.g., in terms of bandwidth and storage resources) since only a small fraction of the websites hosted in an academic domain are author homepages (with many websites corresponding to departments, courses, groups, etc), and (2) misses homepages from research industry labs, which do not belong to the academic domain (e.g., $\sim$51% of the homepages in our dataset are not from the .edu domain). An alternative, more efficient approach, is to use a broader Web search for author discovery together with an accurate homepage classifier.

To this end, in this paper, we propose a novel search-then-classify approach to find researchers homepages on the Web and identify them with powerful deep learning models. Our approach is inspired from the way humans search for scholarly information on the Web. Precisely, humans may directly employ an author name as a query, or may issue a "navigational query" (Broder 2002) comprising of representative keywords (or paper titles) of the author, if the author's name (or the correct spelling) is not known. Using hints from the titles, snippets and the URL strings, human searchers are often able to locate the correct homepage from the Web search results, e.g., by navigating from the paper URL to the index of the homepage where the paper is located. To illustrate this process, Figure 1 shows an example of a Web search using the Bing search engine for the title of a paper published in WWW 2008. In the figure, the link shown in hosted

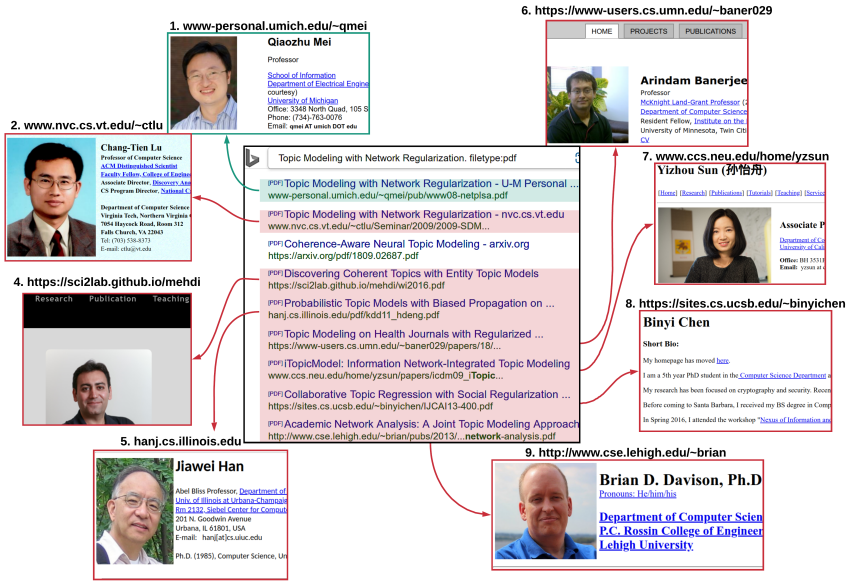

Figure 1: An anecdotal search example using paper title search. Green highlighted response is located on the first author's homepage. Newly discovered authors are highlighted by a red color.

on the homepage of the first author of the searched paper, while the links shown in red point to newly discovered authors. For the paper title search, the homepage of the first author of the searched paper can be accurately retrieved by navigating from the paper's URL to the index of the homepage (see the first result of the figure). Interestingly, notice that homepages belonging to seven other authors (different from the authors of the searched paper) can be discovered through the title search. We posit that this is because scientific paper titles comprise a large fraction of keywords (Litvak and Last 2008), and hence, the words in paper titles serve as excellent keywords to formulate queries that can retrieve topically-related research papers, which are likely to be hosted on researchers' homepages. Indeed, according to previous studies, researchers provide access to their papers (whenever possible) to improve their visibility and citation count, making researcher homepages a likely hub for locating research papers (Lawrence 2001; Gollapalli et al. 2015).

Our search-then-classify approach specifically captures the above aspects by first "issuing" a query to find relevant content from the Web, and subsequently using a *homepage classification* module to identify homepages from the retrieved content (i.e., from the Web search results). Identifying homepages "in the wild" is very challenging since they have different structures and content and the URLs where they are hosted are very diverse and new URLs appear over time (e.g., many researchers use now github for their homepages, which was unlikely 5 or 10 years ago). Using author names and paper titles as queries, together with a homepage classifier, we are able to discover not only homepages of intended authors (e.g., those searched directly by name or those of the searched titles), but also homepages of other authors who work on semantically-related topics. We use deep learning models to learn powerful representations of URLs and page content. Moreover, we explore self training (McClosky, Charniak, and Johnson 2006) in order to reduce the human effort needed to label the data by exploiting unlabeled data for the homepage classification task.

In summary, our contributions are as follows:

- We propose a search-driven homepage finding approach that uses author names and paper titles to find researcher homepages. To our knowledge, we are the first to use *"paper titles"* as queries to discover researcher homepages. Furthermore, we explore Convolutional Neural Networks (CNNs) for author homepage identification,[1] which is a crucial component in our approach.
- We conduct a thorough evaluation of the CNN models trained on both URLs and page content, and show significant improvements in performance over baselines and prior works. Furthermore, we show that self training can improve the performance of the classifier with the small amount of labeled data along with the unlabeled data.
- We perform a large-scale experiment using author names and paper titles from Computer Science as queries, and show the effectiveness of our approach in discovering a large number of homepages. Finally, as part of our contributions, all resulting datasets for author homepage identification and homepage discovery will be made available to further research in this area.[2]We show the development and deployment requirements of our proposed approach in CiteSeerX and the maintenance requirements.

Related Work

Work related to our proposed search-driven homepage classification approach follows in several categories, including

[1]We use author homepage classification or identification interchangeably in this paper.

[2]https://www.cs.uic.edu/∼cornelia/datasets/homepage_discovery

standard supervised learning (based on feature engineering) and graph-based approaches, semi-supervised learning approaches that can leverage unlabeled data and complementary sets of features (or views), deep learning approaches to text classification and focused crawling approaches. We discuss the most relevant works in each of these categories in the remaining of this section.

Supervised and Graph-based Approaches. Homepage finding is well-studied in information retrieval. The homepage finding track in TREC 2001 resulted in various machine learning systems for finding homepages (Xi et al. 2002; Upstill, Craswell, and Hawking 2003; Wang and Oyama 2006). Author/researcher homepage identification is a type of homepage finding task, which has been studied extensively in the context of digital libraries such as CiteSeer (Li et al. 2006) and ArnetMiner (Tang et al. 2008). Among works focusing specifically on researcher homepages, both Tang et al. (2007) and Gollapalli et al. (2013) treated homepage finding as a binary classification task and used various URL and webpage content features for classification. Ranking methods were also explored for homepage finding using the top terms obtained from topic models (Gollapalli, Mitra, and Giles 2011). Qi and Davison (2009) used HTML structure-based features and content-based features for classifying webpages. Wang and Oyama (2006) studied the problem of collecting researcher homepages for Japanese websites using on-page and anchor text features. Ye et al. (2012) also focused on finding high quality researcher homepages. Kang et al. (2011) used the last name of a given author followed by a title of one of his/her publication as a query to a search engine to locate a publication list of the author.

Semi-supervised Learning Approaches. For a given webpage, both URL and the HTML content can be used for classifying the webpage. Multi-view learning is usually considered as maximizing the unanimity between different views (Long, Yu, and Zhang 2008; Christoudias, Urtasun, and Darrell 2012). Co-training (usually with two views) is a type of multi-view learning. Gollapalli et al. (2013) proposed an algorithm for "learning a conforming pair of classifiers" that imitates co-training by using the URL and the HTML content as two different views. Jing et al. (2017) addressed the webpage classification problem using a discriminant common space by learning a multi-view shared transformation in a semi-supervised way. Self training (McClosky, Charniak, and Johnson 2006) uses labeled and unlabeled data usually with a single view to improve the classifier performance.

Deep Learning Approaches. Despite the effectiveness of feature engineering used in traditional machine learning, this is a labor intensive task and sometimes fails to extract all the discriminative information from the data (Goodfellow, Bengio, and Courville 2016). Existing models for homepage classification/identification generally use hand-engineered features extracted from URLs and page content (Gollapalli et al. 2015; Tang, Zhang, and Yao 2007). However, improved semantic representations can be obtained directly from the data using deep learning, and can help avoid problems related to feature engineering. For example, Kim (2014) used Convolutional Neural Networks for representation learning for sentence classification and achieved remarkable results. The author used one convolutional/pooling layer (consisting of filters of three different sizes), together with a word embedding layer that encodes tokens in the input sequence and experimented with several variants of word embeddings, including fixed pre-trained vectors, or randomly initialized word vectors later tuned for a specific task. Zhao et al. (2019) used neural network based homepage identifier to find a homepage of a given researcher within a set of HTML pages which are retrieved for a given researcher name as a query. In contrast, inspired from Kim (2014), we use CNNs for representation learning for homepage classification and aim to identify homepages of semantically-related authors (not just the targeted author) for a given query.

Path 1: Author Name Query
Eric T. Baumgartner filetype:html
Path 2: Paper Title Query
Solving Time-Dependent Planning Problems. filetype:pdf

Table 1: Example of author name and paper title queries.

Focused Crawling. Focused crawling was introduced by Chakrabarti et al. (1999) to deal with the information overload on the Web in order to build specialized collections focused on specific topics. Zhuang et al. (2005) demonstrated the feasibility of using author homepages as alternative resources to collect research papers that are missing from an academic digital library. Garcia et al. (2017) proposed a framework to gather publication list of different researchers by using author names as queries to a web search engine.

As opposed to previous approaches that used researcher names and their affiliations to locate a given researcher's homepage, we focus on researcher homepage discovery and propose an approach that mimics how people skim through the search results to discover homepages on the Web.

Task and Application Description

Task Description Our task is to automatically discover new researchers' homepages on the Web, and augment and maintain up-to-date lists of homepages in open-source digital libraries to enable effective and efficient crawls for collecting documents. To address this task, we propose a search-then-classify approach for discovering researchers' homepages from the Web that mimics the search process adopted by humans. Author names and paper titles, freely available on the Web for specific subject disciplines are used to form suitable queries in our approach. Specifically, Path 1 starts with queries for authors names, while Path 2 starts with queries for paper titles. To identify researchers' homepages, pages retrieved on either path are classified with a CNN model. Table 1 shows examples of queries issued in Path 1 (author names) and Path 2 (paper titles), respectively.

Application Description Our implementation of the search-then-classify framework represents a critical part towards a sustainable CiteSeerX, in that it maintains and augments up-to-date lists of researchers' homepages found on the Web. Given the infeasibility of collecting the entire content on the Web, our search-then-classify framework aims

to minimize the use of network bandwidth and hardware resources by selectively crawling only pages relevant to a (specified) set of topics.

Innovative Use of AI Technology

Convolutional Neural Networks A key component in our framework is a classification module that identifies whether a retrieved webpage is a homepage or not. Inspired by Kim (Kim 2014), we use Convolutional Neural Networks for representation learning of URLs and page content. Convolutional Neural Networks (CNNs) (LeCun et al. 1998) are a special kind of neural networks to process grid-like structured data, including sequence or time series data. CNNs are associated with the idea of a "moving filter." Thus, in our approach, we explore a CNN based classifier for identifying homepages from the retrieved results in both search paths.

The CNN architecture used in our experiments is shown in Figure 2, and is comprised of mainly three layers: a word embedding layer, a convolutional layer followed by max pooling, and a fully connected layer for the classification. The embedding layer for tuning task specific word embeddings is initialized to a random vector corresponding to the input sequence. A convolutional layer consists of multiple filters of different sizes (e.g., sizes 3 and 4, respectively) that generate multiple feature maps (e.g., 300) for each filter size. Pooling is usually used after the convolutional layer to reduce the dimensionality (i.e., number of parameters) and prevent overfitting. The common practice for text is to extract the most important feature within each feature map (Collobert et al. 2011), called 1-max pooling. In our architecture, 1-max pooling is applied over each feature map and the maximum values from each filter are selected. These maximum values are then concatenated and used as input to a fully connected layer for the classification task (homepage versus not-homepage). We minimize a sigmoid (or binary) cross-entropy loss function using Adam optimizer to correctly predict the class label. If y_i is the true label and $p(y_i)$ is the predicted label, then the cross-entropy loss function (L) for N examples is calculated as:

$$L = -\frac{1}{N}\sum_{i=1}^{N} y_i.log(p(y_i)) + (1 - y_i).log(1 - p(y_i)) \quad (1)$$

We investigate two types of representations derived using CNN: (1) word based HTML content; and (2) word based URL. As can be seen from Figure 2, we explore the CNN models individually on either URL or page content, or jointly on both URL and page content. The URLs and corresponding pages (content) are those obtained as the result of our search for author name and paper title queries.

For the word based HTML model, we consider the first 1000 words from the HTML content of each page (given that the average length of HTMLs in the homepage class in our dataset is 982, and most of the homepage characteristics often appear in the beginning of a page). Furthermore, we remove stop words and digits, and consider words appearing in at least 10 documents. For the URL based model, we tokenize the URLs with '/' as a delimiter, and form the vocabulary from all unigrams that appear in WordNet (Miller 1995). Consistent with Gollapalli et al. (2013), for words that do not appear in WordNet, we add URL string patterns to the vocabulary, including underscored or hyphenated words, words with the '∼' sign, alphanumeric words, and long words (i.e., words with more than 30 characters). These patterns can help to filter out course pages, announcements, calendars, etc. For this model, we consider words appearing in at least 3 URLs.

Actual URL	www.cc.gatech.edu/∼mnaik7/pubs/popl16.pdf
Candidate URLs	www.cc.gatech.edu/∼mnaik7/pubs/ www.cc.gatech.edu/∼mnaik7/ www.cc.gatech.edu/

Table 2: Example URLs and candidate URLs.

Semi-supervised Teacher-Student Model As discussed in the Introduction, one major challenge in identifying homepages is that the URLs where homepages can be hosted change over time. In order to reduce the human effort for data annotation, we investigate self-training in a Teacher-Student fashion to utilize the unlabeled data together with already labeled data. The model works in four steps in an iterative manner (Figure 3): (1) labeled data is used to train a teacher model; (2) the teacher model is used on unlabeled data to generate pseudo labels; (3) the student model is trained using both labeled data and pseudo labeled data (unlabeled data); (4) iterate the process by putting back the student as a teacher to generate new pseudo labels for training new student model. We considered examples which are predicted positive or negative with the probability ≥ 0.8 and ≤ 0.2, respectively. In our case, we used *data balancing* while using pseudo labeled data and sampled same number of examples as the training set with equally sampling from each slot of 0.05 range. As an example, for the positively labeled data using the teacher model we sample equally from 0.8 to 0.85, 0.85 to 0.90, and so on. The backbone of our model is the CNN on both page content and URL.

Generating Candidate URLs

Note that for each query, a set of URLs are retrieved. We discard responses from a list of 25 domains such as "ResearchGate", "LinkedIn", etc. Candidate homepage URLs for each retrieved URL in Path 2 are generated by first splitting the URL on "/" and then removing the last part of the URL, iteratively, until the domain is reached. In the candidate set of URLs, we keep only the URLs for which we are able to obtain the corresponding HTML. Examples of candidate URLs for a paper title search is shown in Table 2. For Path 1, we use retrieved search results for the candidate URLs.

Datasets

We now describe the datasets that we use in the evaluation of the CNN homepage classifier and our overall approach.

DBLP Dataset

The WebKB dataset collected in 1997 has been previously used for homepage classification (Gollapalli et al. 2013). However, due to continuous changes in the information content on academic websites, this dataset has become outdated. For example, academic websites today contain invited talks,

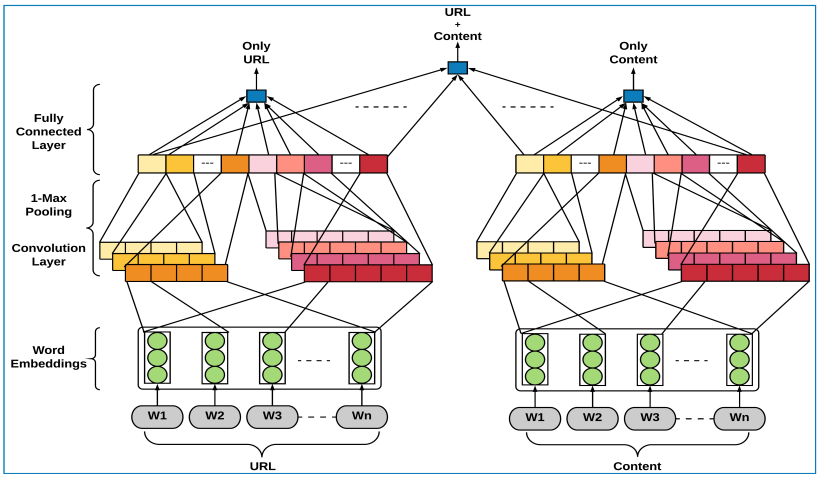

Figure 2: Illustration of our CNN architecture used for homepage classification.

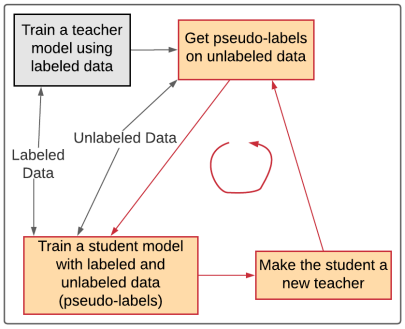

Figure 3: Teacher-Student architecture of self training.

newsletters, job postings, and other events that do not occur in WebKB. In addition, the WebKB dataset does not contain homepages from industry lab researchers. To address the above limitations and in order to enable the exploration of deep learning for author homepage identification, we constructed a labeled dataset for this task as follows.

We obtained a list of author names and their homepages from DBLP in 2015. After data processing, we found that the original DBLP list had a repetition of 21 homepages and contained some URLs that were easily identified as non-homepages, specifically, 56 URLs from Wikipedia, 2 from DBLP and 1 from Springer. After removing repetitions and pages linking to Wikipedia, DBLP, and Springer, we ended up with a list of $13,239$ homepages. We further refined the original DBLP list by removing all outdated URLs, $5,596$ in total (HTTP 4xx client errors, non-valid homepages, or redirects to the default university/company page). We ended up with a list of $7,643$ author names and their homepages. We used the author names as queries to the Bing search API and retrieved the top-10 results for each query, obtaining $76,375$ search responses from Bing in total. From the Bing responses, we filtered out pages from the list of 25 domains such as "ResearchGate," "LinkedIn," "Wikipedia," "YouTube", etc. After removing such pages, and the overlap with the original DBLP set of homepages, we ended up with $20,229$ Bing responses. We manually inspected the set of $27,872$ URLs ($7,643$ DBLP URLs + $20,229$ Bing URLs) for the labeling task, using three Amazon Mechanical Turk (AMT) workers and two undergraduate students, who were trained in an iterative fashion and worked closely with the researchers. Whenever there was agreement between the three AMT workers, we labeled the example accordingly (as homepage or non-homepage). When there was disagreement, the data was labeled further by the undergraduate students, and if a decision could not be reached, the final adjudication was made by one of the researchers. During the labeling task, we removed outdated and non-English pages. At the end of the annotation task, we found $8,529$ positive examples (homepages) and $16,245$ negative examples (non-homepages). After this manual annotation, we found

	+ve	-ve ($\theta \geq 50$)	-ve (all)
#Examples	8,529	10,204	16,245
#URLs as a domain	439	1,097	1,156
#URLs with $\sim$ sign	3,974	113	120
#URLs from '.edu'	4,192	1,304	2,307
#URLs from '.com'	464	5,290	8,929
#URLs containing digit	1,034	5,191	9,130
#Pages with 'homepage' word or its synonyms	3,639	2,048	3,759
Max. #characters/URL	158	232	297
Avg. #characters/URL	32	50	52
Max. #words/webpage	86K	530K	530K
Avg. #words/webpage	982	2,011	2,418

Table 3: Datasets characteristics.

that only $5,851$ out of $7,643$ DBLP homepages are valid (with the others being, e.g., moved to a different page). The result of this search validates our intuition that we can obtain homepages for the intended authors, but also additional homepages of related authors (e.g., co-authors).

In the set of negative domains, we observed that there were hundreds of URLs from domains such as "healthgrades.com", "ratemyprofessor.com", etc. Thus, we constructed the final dataset by sampling only 50 negative URLs from a given domain (Threshold $\theta \geq 50$). For domains with less than 50 URLs, we used all URLs. The final dataset used in our experiments contains $18,733$ examples, specifically $8,529$ homepages and $10,204$ non-homepages.

Characteristics of the DBLP Dataset Table 3 contains the characteristics of the DBLP dataset. As we can see, the percentage of URLs containing a $\sim$ sign or being from the '.edu' domain is higher in the positive set as compared to the two negative sets (specifically, $3,974$ out of $8,529$ URLs from the positive set contain the $\sim$ sign). On the other hand, in both negative sets, the percentage of URLs containing a digit or from the '.com' domain is higher as compared with the positive set. Moreover, 43%, 20%, and 23% of webpages contain the keyword 'homepage' or its synonyms such as personal page, personal site, personal website, etc. in the

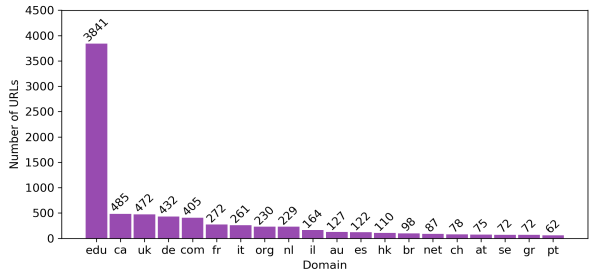

Figure 4: Number of URLs corresponding to homepage from different domains in our DBLP dataset.

positive set, negative set (threshold-50), and negative set (all), respectively. Also the maximum and average #characters per URL and #words per webpage are smaller in the positive set as compared with both negative sets. Figure 4 shows the top-20 domains for the positive set.

CiteSeerX Dataset

Our second dataset, which we used for the large scale evaluation of our overall framework, is compiled from CiteSeerX. Specifically, we extracted papers published in venues related to machine learning, information retrieval, and computational linguistics. Overall, we obtained a random set of $10,000$ paper titles and $14,808$ authors (unique names corresponding to the selected titles) for the evaluation of our search-driven approach on a large scale. These venues along with the number of papers in each venue are as follows: ACL (1193), IJCAI (1167), COLING (1010), ICRA (827), NIPS (650), VLDB (613), ICML (564), AAAI (411), CHI (399), CVPR (371), KDD (366), EACL (333), SIGIR (305), SAC (296), SDM (242), ICDM (236), CIKM (235), WWW (231), LREC (226), HLT-NAACL (209), and EMNLP (116).

Experiments and Results

Next, we describe our experiments and results on homepage classification along with the performance of our overall search-then-classify framework.

Author Homepage Classification

To evaluate the performance of the CNN models on homepage classification, and compare them with previous approaches for this task, we divided our DBLP dataset into train, validation and test sets. The train, validation, and test sets have 60%, 20% and 20% examples, respectively. All the splits are constructed by keeping the original distribution of the URL set. We use the validation set for parameter tuning and model selection. We report precision, recall and F1-score for the positive class and the overall accuracy for each model on the test set (using the model that performed best on the validation set). For CNN models, we run the experiments with three different random initialization of the network weights and we report average values for each measure. The model that achieved the highest performance on the test set was chosen for the large scale experiment.

CNN vs. Supervised Models and Co-training We contrast the performance of CNNs on different input types (URL, HTML content, and the combination) with the performance of several traditional supervised classifiers (Random Forest, Decision Trees, Naïve Bayes, and Support Vector Machines with a linear kernel), as well as with the performance of a semi-supervised co-training classifier for homepage identification as described in (Gollapalli et al. 2013).

Model	Precision	Recall	F1	Accuracy
CNN-URL	**0.91**	0.75	0.82	85.11%
CNN-Content	0.89	0.90	0.89	90.38%
CNN-Combined	0.90	**0.94**	**0.92**	**92.35%**
Co-training	0.87	0.86	0.87	87.64%
RF-URL	0.89	0.84	0.87	88.10%
RF-Content	0.83	0.92	0.87	87.35%
RF-Combined	0.85	0.91	0.88	88.55%

Table 4: CNN vs. co-training and supervised models.

We used the *tf-idf* vector representations for all input types (URL and content based) for the traditional supervised classifiers. We trained the CNN models using mini-batches of size 64, with a sigmoid cross-entropy loss function and Adam optimizer with a learning rate of 0.0005. After experimenting with a large spectrum of parameters for CNNs on the validation set, the best parameters are as follows: for CNN-URL, 100 embedding size and 100 filters of size 5; for CNN-content, 300 embedding size and 100 filters of size 5. For co-training, we obtained the code and unlabeled data from Gollapalli et al. (2013).[3]

Table 4 shows the performance on the test set of the CNN classifiers compared with co-training and supervised Random Forest (RF) classifiers. The RF classifiers performed the best among all the traditional supervised classifiers (and therefore we only show its results in the table). CNN classifiers are comparable and in many cases outperform their traditional supervised counterparts except the URL based classifier, and also the co-training approach proposed in (Gollapalli et al. 2013). We can also see that the CNN-combined, which uses both word-based content and word-based URL, achieves the highest performance among all models, in terms of recall and F1-score. For example, CNN-combined achieves the highest F1 of 0.92, whereas the RF-combined (URL + content) yields an F1-score of 0.88. We can also observe that CNN-URL achieves a highest precision of 0.91.

The Effect of Self-training To see the effectiveness of self-training, we perform experiments using different portion of the training data along with unlabeled data to train our homepage classifier (CNN-Combined) using self-training. For the unlabeled data used in the self-training of our homepage classifier, we use candidate URLs of Path 1 (author name queries) of our proposed framework. We used $56,339$ HTML pages and URLs as the unlabeled set. We went up to 5 iterations of self-training, iteratively learning a teacher-student model. Table 5 shows the results highlighting the effect of self-training on the CNN-Combined classifier. We can see that the performance of the classifier improves when the labeled data is $\leq 25\%$ (only $2,811$ labeled

[3] https://sites.google.com/site/sujathadas/home/datasets

Labeled Data	Precision	Recall	F1	Accuracy
Without self training				
1%	0.77	0.80	0.78	79.37%
5%	0.88	0.86	0.87	88.33%
10%	0.88	0.89	0.88	89.20%
25%	0.88	0.92	0.90	90.81%
50%	0.89	**0.94**	**0.92**	92.06%
100%	0.90	**0.94**	**0.92**	92.35%
With self training (uses unlabeled data)				
1%	0.83	0.85	0.84	84.96%
5%	0.85	0.91	0.88	88.85%
10%	0.87	0.91	0.89	89.60%
25%	0.87	**0.94**	0.91	91.06%
50%	0.90	0.93	**0.92**	92.14%
100%	**0.91**	0.93	**0.92**	**92.63%**

Table 5: The performance of CNN-Combined model with and without self training.

False Positives	
Example	Confidence
1. http://users.ics.aalto.fi/eugenc	1.0
2. http://www.nott.ac.uk/∼itzbl/	1.0
3. http://www.eng.usf.edu/∼rfrisina/	0.9999
4. http://www.ssrc.ucsc.edu/person/phartel.html	0.9952
False Negatives	
Example	Confidence
5. http://zh-anse.com	0.9995
6. http://www.brookings.edu/experts/yuq	0.9989
7. https://blog.xot.nl/about-2	0.9858
8. http://freudenbergs.de/bert	0.9805

Table 6: Errors made by CNN-Combined model, along with model's confidence values. The blue part in each URL indicates the part of the URL that is used as input to a model.

examples). For example, while using only 1% (112) labeled examples, the performance of the classifier increased by $\approx 8\%$ using self-training from F1 of 0.78 to 0.84. For the error analysis and the large-scale experiments, we used the CNN-Combined classifier trained using self-training with 100% training examples. These results show that self-training can be very useful when we want to deploy/train the homepage classifier for other domains where the labeled data is not easily available, but we can easily collect the unlabeled data relevant to the task. Also, self-training can be very useful in order to reduce the human labeling effort, especially given the changing nature of the URL types/domains where homepages can be located.

Error analysis Table 6 shows sampled URLs along with model's confidence value where CNN-Combined model made errors. The blue part of each URL indicates the part of the URL that is used as an input to a deep learning model. Most of the false positive examples are pointing to webpages of a research group/lab or their list of people/members, and to webpages containing basic information regarding a professor or a person. Specifically, URL-1, URL-2 and URL-3 are pointing to a webpage of a member of the research group/lab, while URL-4 is pointing to a webpage containing basic information regarding a professor. Most of the false negative examples are pointing to a homepage containing very little research-related information, or a homepage with very different content than that expected on a usual researcher homepage. URL-5 contains very little HTML content (unlike a typical researcher homepage). URLs 6, 7, and 8 are also homepages with very different content than the content on a usual researcher homepage.

Queries	**URLs**	**Candidate URLs**	CNN-comb. HPs	
			All	**Unique**
Author names	148,042	56,339	12,093	11,016
Paper titles	75,612	51,451	17,685	12,199
Overlapping	-	-	-	2,622
Total	-	-	-	20,593

Table 7: Homepages from $10,000$ title and $14,808$ author search responses in a large-scale experiment.

Large-Scale Experiments

We now evaluate the capability of our search-then-classify approach to discover author homepages in a large-scale experiment, using the CiteSeerX dataset. To this end, we use the $14,808$ author names and the $10,000$ paper titles as search queries on the Bing search API and employ the CNN-combined homepage classifier to identify homepages from the top-10 search results of each query. We used only the top-10 results as they are shown to often be sufficient to retrieve the relevant information (Spink and Jansen 2004).

Overall yield. The total number of Bing URL responses for our author name and paper title queries, the number of resulting candidate URLs (corresponding to the retrieved URLs), and the number of predicted homepages (overall and unique) obtained using the CNN-combined classifier are shown in Table 7. Individually, for the $14,808$ author name queries we obtained $148,042$ Bing URL responses (i.e., URLs pointing to html pages). After filtering out the easy-to-identify non-homepage commercial URLs, we generated $56,339$ candidate URLs (see Table 2 for examples of candidate URLs), out of which $12,093$ are classified as homepages by our CNN-combined classifier. From this set of $12,093$ predicted homepages, we obtained $11,016$ unique author homepages.

For the $10,000$ paper title queries, we obtained $75,612$ Bing URL responses (i.e., URLs pointing to PDF documents). After filtering out the commercial URLs, we generated $51,451$ candidate URLs, out of which $17,685$ are classified as homepages by our CNN-combined classifier. From this set of $17,685$ predicted homepages, we obtained $12,199$ unique author homepages.

Table 7 shows also the overlap in the two sets of unique homepages between author name search and paper title search, which consists of $2,622$ unique homepages. As expected, this small overlap of homepages between author name search and paper title search indicates that research paper titles, formulated as queries, have a great potential to discover new researcher homepages (in addition to the homepages of researchers searched specifically). Precisely, through the paper title search and using the CNN-combined classifier, we were able to find an additional $9,577$ (=

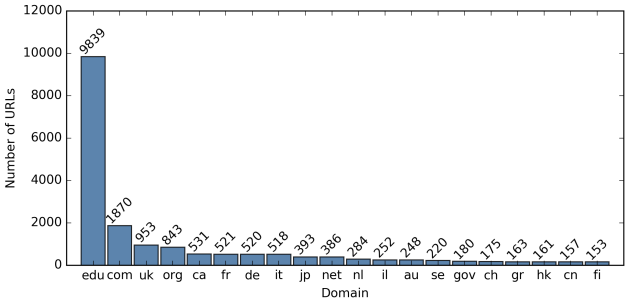

Figure 5: Top-20 domains from author and title searches.

$12,199 - 2,622$) unique homepages (as compared to the author name search). This result also suggests that the two types of searches complement each other and are capable to reach different sections of the Web. The total number of homepages we discover through the author name search and paper title search in our approach is $20,593$ (= $11,016$ + $12,199$ - $2,622$). This total number of homepages showcases the potential of our approach to acquire and maintain large collections of homepages.

To see where the homepages acquired using our approach come from (what parts of the Web), we extracted the number of different domains and ranked them based on the number of URLs in each domain. Figure 5 shows the top-20 domains for the $20,593$ homepages. In total, we found 107 domains. Around 48% of homepages are from the ".edu" domain. The top-20 domains shown in the figure cover $\approx 89\%$ of total homepages that we discovered.

To see for how many authors out of $14,808$ authors we were able to locate the homepages, we used Stanford Named Entity Recognizer. [4] We used the first appearing name from the predicted homepage as the owner of the homepage. To match the author names, we consider a match if we were able to match first and last name. We recovered homepages for $5,815$ and $2,782$ intended authors using author names and paper title queries, respectively. These results also showcase the potential of paper title queries to locate the homepages of other authors than the authors of the queried titles.

Overlap with DBLP author homepages One interesting question that can be raised is the following: "Is our approach able to discover homepages that are not already available in some online resource?" That is, starting with our sets of author names and paper titles, which are independent from the author names in the DBLP list of homepages, how many homepages can we discover that are not already in the DBLP list? To answer this question, we compare our overall $15,203$ homepages predicted by the CNN-combined classifier with the list of known DBLP homepages ($5,851$ homepages). The overlap between the $15,203$ predicted homepages in our approach and $5,851$ DBLP homepages is $1,882$. Hence, remarkably, overall, we are able to discover $18,711$ = $20,593$ - $1,882$ homepages that are not present in our DBLP dataset.

[4] https://nlp.stanford.edu/software/CRF-NER.html

Human Assessment and Validation. Key to our large scale experiment is to ensure data quality of the discovered homepages. In order to analyze the impact of the system effectiveness, we performed human assessment and validation. Specifically, we sampled 600 CNN-predicted homepages for human assessment and validation. We asked a human annotator (the first author of this paper) to determine if each page provided in the set is a homepage or not. The human annotator labeled 496 out of 600 URLs as homepages. In other words, 82.67% URLs from the sampled set were identified as true homepage. Close inspection of the remaining 104 URLs revealed that, most of those URLs are pointing to a group or lab or course page. Furthermore, we noticed that 266 out of 600 URLs contain a $\sim$ sign, and 246 URLs with a $\sim$ sign were marked as a homepage by the human annotator.

Development and Deployment

Although CiteSeerX utilizes open source software packages, many core components are not directly available from open source repositories and require extensive programming and testing. The current CiteSeerX codebase inherited little from its predecessor's (CiteSeer) for stability and consistency. The core part of the main web apps were written by Dr. Isaac Councill and Juan Pablo Fernández-Ramírez and many components were developed by other graduate students, postdocs and software engineers, which took at least 3-4 years. Different components of CiteSeerX are built using several languages such as JAVA, Python, Perl, Scala, etc. The homepage classifier component is developed using Python 2.7. We have used the Amazon AWS service for training the deep learning-based homepage classifier. The AWS instance that was used for training the classifier has Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz processor, 64GB RAM, and Tesla V100 SXM2 16GB GPU.

To collect documents, CiteSeerX crawls researchers' homepages, URLs from the Microsoft Academic Graph and Google Scholar, and maintains whitelists and blacklists for crawling. Our framework is integrated in CiteSeerX to continuously augment and maintain the URLs collection used for crawling (in order to preserve network bandwidth and hardware). CiteSeerX also directly incorporates PDFs from PubMed, arXiv, and digital repositories in a diverse spectrum of disciplines such as mathematics, physics, and medical science. These crawled documents are passed to multiple AI modules such as document classification, document deduplication and citation graph, metadata extraction, header extraction, citation extraction, etc. The ingestion module writes all metadata into the database. The PDF documents are renamed under a document ID (csxDOI) and saved to the production repository with XML metadata files. The index data are also updated. During the application development, we have learned that identifying homepages "in the wild" is very challenging since they have very diverse structures and content. Moreover, based on the human annotation task, identifying author homepages is difficult sometimes even for human annotators.

Maintenance

The researcher homepage classifier is developed and maintained by one graduate student, whereas the web-crawler component in CiteSeerX is developed and maintained by several graduate students. The homepages finding project received partial financial support from the National Science Foundation aimed at a sustainable CiteSeerX. While collecting the documents from homepages (seed URLs), multiple types of documents can be found on homepages such as CVs, slides, syllabus, homeworks, etc. which should not be included in CiteSeerX. Thus we found that a classifier that distinguishes research articles from other types of documents, as described in (Caragea et al. 2016), should be used on the crawled documents. During maintenance, as new homepages emerge and also existing authors may change affiliations or the homepage may get outdated (4XX error), periodically we need to automatically update the list of homepages as well as remove the outdated homepages without the human effort. The maintenance work includes, but is not limited to fixing bugs, updating the list of URLs including researcher homepages, periodically checking the system health, and running the web-crawlers. CiteSeerX data is updated regularly. The crawling rate varies from 50,000 to 100,000 PDF documents per day. Of the crawled documents, about 40% are eventually identified as being academic and ingested into the database.

Conclusion and Future Work

In this paper, we presented a novel search-then-classify approach to discover researcher homepages using author names and paper titles as queries in order to augment and maintain the URL lists for document crawling in CiteSeerX. To our knowledge, we are the first to interleave Web search and deep learning for researcher homepage identification to build an efficient author homepage acquisition approach. This is a useful component in CiteSeerX, which crawls researchers' homepages to collect research papers for inclusion in the library. Moreover, we show that self-training can be very useful to train deep learning based researcher homepage classifiers using small amount of labeled data along with unlabeled data. Since data annotation is very expensive, we show that human effort can be reduced through self-training, which could be useful when deploying this into another system in future. Our results showcase the potential of our approach. More interestingly, we discovered $12,199$ researcher homepages using $10,000$ paper title queries. This shows the capability of research paper titles for finding researcher homepages. We show the integration of our framework in CiteSeerX for collecting URLs for crawling scientific documents. The new datasets that we created are made available online to foster research in this area.

Acknowledgments

We thank our anonymous reviewers for their constructive comments and feedback, which helped improve our paper. This research is supported in part by NSF. Any opinions, findings, and conclusions expressed here are those of the authors and do not necessarily reflect the views of NSF

References

Alzaidy, R.; Caragea, C.; and Giles, C. L. 2019. Bi-LSTM-CRF Sequence Labeling for Keyphrase Extraction from Scholarly Documents. In Liu, L.; White, R. W.; Mantrach, A.; Silvestri, F.; McAuley, J. J.; Baeza-Yates, R.; and Zia, L., eds., *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, 2551–2557. ACM.

Balog, K.; and De Rijke, M. 2007. Determining Expert Profiles (with an Application to Expert Finding). In *IJCAI*.

Broder, A. 2002. A Taxonomy of Web Search. *SIGIR Forum* 36(2).

Caragea, C.; Bulgarov, F.; and Mihalcea, R. 2015. Co-Training for Topic Classification of Scholarly Data. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2357–2366. Lisbon, Portugal: Association for Computational Linguistics. doi:10.18653/v1/D15-1283. URL https://www.aclweb.org/anthology/D15-1283.

Caragea, C.; Bulgarov, F. A.; Godea, A.; and Das Gollapalli, S. 2014. Citation-Enhanced Keyphrase Extraction from Research Papers: A Supervised Approach. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1435–1446. Doha, Qatar: Association for Computational Linguistics. doi:10.3115/v1/D14-1150. URL https://www.aclweb.org/anthology/D14-1150.

Caragea, C.; Wu, J.; Gollapalli, S. D.; and Giles, C. L. 2016. Document Type Classification in Online Digital Libraries. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, 3997–4002. AAAI Press.

Chakrabarti, S.; van den Berg, M.; and Dom, B. 1999. Focused crawling: a new approach to topic-specific Web resource discovery. In *WWW*.

Chen, H.-H.; Gou, L.; Zhang, X.; and Giles, C. L. 2011. Collabseer: a search engine for collaboration discovery. In *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries*, 231–240.

Christoudias, C.; Urtasun, R.; and Darrell, T. 2012. Multi-view learning in the presence of view disagreement. *arXiv preprint arXiv:1206.3242* .

Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; and Kuksa, P. 2011. Natural language processing (almost) from scratch. *JMLR* 12(Aug): 2493–2537.

Florescu, C.; and Caragea, C. 2017. PositionRank: An Unsupervised Approach to Keyphrase Extraction from Scholarly Documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1105–1115. Vancouver, Canada: Association for Computational Linguistics. doi:10.18653/v1/P17-1102. URL https://www.aclweb.org/anthology/P17-1102.

Garcia, C. M.; Pereira, A. H.; and Pereira, D. A. 2017. A framework to collect and extract publication lists of a given researcher from the web. *JWET* 12(3): 234–252.

Giles, C. L.; Bollacker, K. D.; and Lawrence, S. 1998. CiteSeer: An automatic citation indexing system. In *JCDL*, 89–98.

Gollapalli, S. D.; and Caragea, C. 2014. Extracting Keyphrases from Research Papers Using Citation Networks. In Brodley, C. E.; and Stone, P., eds., *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada*, 1629–1635. AAAI Press.

Gollapalli, S. D.; Caragea, C.; Mitra, P.; and Giles, C. L. 2013. Researcher homepage classification using unlabeled data. In *WWW*, 471–482. ACM.

Gollapalli, S. D.; Caragea, C.; Mitra, P.; and Giles, C. L. 2015. Using unlabeled data to improve researcher homepage classification. In *TWEB*.

Gollapalli, S. D.; Mitra, P.; and Giles, C. L. 2011. Learning to Rank Homepages For Researcher Name Queries. In *EOS Workshop at SIGIR*.

Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. *Deep Learning*. The MIT Press. ISBN 0262035618, 9780262035613.

Jing, X.-Y.; Wu, F.; Dong, X.; Shan, S.; and Chen, S. 2017. Semi-supervised multi-view correlation feature learning with application to webpage classification. In *AAAI*.

Kang, I.-S.; Kim, P.; Lee, S.; Jung, H.; and You, B.-J. 2011. Construction of a large-scale test set for author disambiguation. *Information Processing & Management* 47(3): 452–465.

Khabsa, M.; Treeratpituk, P.; and Giles, C. L. 2014. Large scale author name disambiguation in digital libraries. In *2014 IEEE International Conference on Big Data, Big Data 2014, Washington, DC, USA, October 27-30, 2014*, 41–42. IEEE Computer Society.

Kim, K.; Sefid, A.; Weinberg, B. A.; and Giles, C. L. 2018. A Web Service for Author Name Disambiguation in Scholarly Databases. In *2018 IEEE International Conference on Web Services, ICWS 2018, San Francisco, CA, USA, July 2-7, 2018*, 265–273. IEEE.

Kim, Y. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1746–1751. Doha, Qatar: Association for Computational Linguistics. doi:10.3115/v1/D14-1181. URL https://www.aclweb.org/anthology/D14-1181.

Lawrence, S. 2001. Free online availability substantially increases a paper's impact. In *Nature*, 411 (6837), 521–521.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11): 2278–2324.

Li, H.; Councill, I. G.; Bolelli, L.; Zhou, D.; Song, Y.; Lee, W.-C.; Sivasubramaniam, A.; and Giles, C. L. 2006. Citeseer^x: a scalable autonomous scientific digital library. *InfoScale* .

Litvak, M.; and Last, M. 2008. Graph-Based Keyword Extraction for Single-Document Summarization. In *MMIES '08*, 17–24. ISBN 978-1-905593-51-4.

Long, B.; Yu, P. S.; and Zhang, Z. 2008. A general model for multiple view unsupervised learning. In *SIAM*, 822–833.

Lu, Q.; and Getoor, L. 2003. Link-Based Classification. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, 496–503. AAAI Press. ISBN 1577351894.

McClosky, D.; Charniak, E.; and Johnson, M. 2006. Effective self-training for parsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, 152–159.

Miller, G. A. 1995. WordNet: a lexical database for English. *Communications of the ACM* 38(11): 39–41.

Patel, K.; and Caragea, C. 2019. Exploring Word Embeddings in CRF-based Keyphrase Extraction from Research Papers. In Kejriwal, M.; Szekely, P. A.; and Troncy, R., eds., *Proceedings of the 10th International Conference on Knowledge Capture, K-CAP 2019, Marina Del Rey, CA, USA, November 19-21, 2019*, 37–44. ACM.

Patel, K.; Caragea, C.; Wu, J.; and Giles, C. L. 2020. Keyphrase Extraction in Scholarly Digital Library Search Engines. In *International Conference on Web Services*, 179–196. Springer.

Qi, X.; and Davison, B. D. 2009. Web Page Classification: Features and Algorithms. *ACM Comput. Surv.* 41(2).

Spink, A.; and Jansen, B. J. 2004. *Web search: Public searching of the Web*, volume 6. Springer Science & Business Media. ISBN 9781402022692.

Tang, J.; Zhang, D.; and Yao, L. 2007. Social Network Extraction of Academic Researchers. In *ICDM*.

Tang, J.; Zhang, J.; Yao, L.; Li, J.; Zhang, L.; and Su, Z. 2008. Arnetminer: extraction and mining of academic social networks. *KDD* .

Upstill, T.; Craswell, N.; and Hawking, D. 2003. Query-independent Evidence in Home Page Finding. *TOIS* .

Wang, Y.; and Oyama, K. 2006. Web Page Classification Exploiting Contents of Surrounding Pages for Building a High-Quality Homepage Collection. *ICADL* 4312: 515–518.

Xi, W.; Fox, E.; Tan, R.; and Shu, J. 2002. Machine learning approach for homepage finding task. In *SPIRE*, 169–174. Springer.

Ye, J.; Qian, Y.; and Zheng, Q. 2012. PLIDMiner: A Quality Based Approach for Researcher's Homepage Discovery. In *Asia Information Retrieval Symposium*, 199–210. Springer.

Zhao, J.; Liu, T.; and Shi, J. 2019. Improving Academic Homepage Identification from the Web Using Neural Networks. In *ICCS*, 551–558. Springer.

Zhuang, Z.; Wagle, R.; and Giles, C. L. 2005. What's there and what's not?: focused crawling for missing documents in digital libraries. In *JCDL*, 301–310. IEEE.