Automatic Identification of Research Articles from Crawled Documents

Cornelia Caragea[1,3], Jian Wu[2,4], Kyle Williams[2,4], Sujatha Das G.[1,3], Madian Khabsa[1,4], Pradeep Teregowda[1,4], and C. Lee Giles[1,2,4]
[1]Department of Computer Science and Engineering, [2]College of Information Sciences and Technology
[3]University of North Texas, Denton, TX, [4]The Pennsylvania State University, University Park, PA
ccaragea@unt.edu, jxw394@ist.psu.edu, kwilliams@psu.edu, gsdas@cse.psu.edu, madian@psu.edu, pbt105@psu.edu, giles@ist.psu.edu

ABSTRACT

Online digital libraries that store and index research articles not only make it easier for researchers to search for scientific information, but also have been proven as powerful resources in many data mining, machine learning and information retrieval applications that require high-quality data. The quality of the data available in digital libraries highly depends on the quality of a classifier that identifies *research articles* from a set of crawled documents, which in turn depends, among other things, on the choice of the feature representation. The commonly used "bag of words" representation for document classification can result in prohibitively high dimensional input spaces and may not capture the specifics of research articles. In this paper, we propose *novel features* that result in *effective* and *efficient* classification models for automatic identification of research articles. Experimental results on two datasets compiled from the CiteSeer^x digital library show that our models outperform strong baselines using a significantly smaller number of features.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics

General Terms

Theory

Keywords

Document classification, structural features, research article identification, digital libraries

1. INTRODUCTION

As science advances, scientists around the world continue to produce large numbers of research articles, which provide the technological basis for worldwide dissemination of scientific discoveries. Online digital libraries such as DBLP, CiteSeer^x, Microsoft Academic Search, ArnetMiner, arXiv, ACM Digital Library, Google Scholar, and PubMed that store research articles or their metadata, have become a medium for answering questions such as: how research ideas emerge, evolve, or disappear as a topic; what is a good measure of quality of published works; what are the most promising areas of research; how authors connect and influence each other; who are the experts in a field; and what works are similar. In particular, CiteSeer^x [8] has been proven as a powerful resource in many applications that analyze research articles on a web-wide scale. Such applications include topic classification of research articles [3], document and citation recommendation [13, 15], author name disambiguation [20], expert search [10], topic evolution [11], collaborator recommendation [5], joint modeling of documents' content and authors' interests [17].

To be successful, these applications require *accurate* and *complete* collections of research articles, which highly depend on the quality of a classifier that identifies *research articles* from a set of documents crawled on the Web.

A rule-based system that classifies documents as research articles if they contain any of the words `references` or `bibliography`, as is currently in use by CiteSeer^x [8], will mistakenly classify documents such as curriculum vita or slides as research articles whenever they contain the word `references` in them, and will miss to identify research articles that do not contain any of the two words. In contrast, the commonly used "bag of words" representation for document classification can result in prohibitively high-dimensional input spaces. Machine learning algorithms applied to these input spaces may be intractable due to the large number of dimensions. In addition, the "bag of words" may not capture the specifics of research articles, e.g., due to the diversity of the topics covered in CiteSeer^x. As an example, an article in Human Computer Interaction may have a different vocabulary space compared to a paper in Information Retrieval, but some essential terms may persist across the papers, e.g., "references" or "abstract". The number of tokens could be also very informative, i.e., the number of tokens in a research article is generally much higher than in a set of slides, but much smaller than in a PhD thesis. However, these are not captured by the "bag of words" representation.

Moreover, the number of crawled documents can be in the order of millions. For example, Figure 1 shows the increase in both the number of crawled documents as well as the

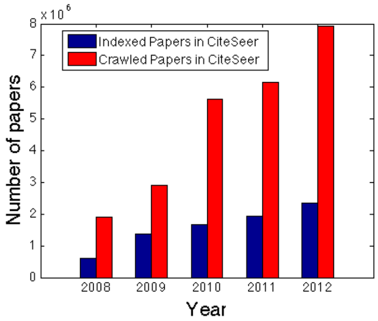

Figure 1: The growth in the number of crawled documents as well as in the number of research papers indexed by CiteSeer^x between 2008 and 2012.

number of research articles indexed by CiteSeer^x between 2008 and 2012. As can be seen from the figure, the number of crawled documents has increased from less than two million to almost eight million, whereas the number of indexed documents has increased from less than one million to more than two million. The "bag of words" approach applied to such data are not necessarily very efficient. Therefore, algorithms that can process data into features fast at runtime and result in high-accuracy classifiers are greatly needed.

Against this background, one question that can be raised is: *Can we design features that capture the specifics of research articles and result in classification models that accurately and efficiently identify the research articles from a collection of documents crawled on the Web?* The research that we describe in this paper addresses specifically this question.

Contributions. We describe novel features, called structural features, extracted from the content and the structure of crawled documents. We evaluate these features on documents crawled using CiteSeer^x crawlers and experimentally show that they can be successfully used to identify research articles. The structural features based models substantially outperform the "bag of words" models and a rule-based learner that uses the existence of the words "references" or "bibliography" to identify research articles.

2. RELATED WORK

We formulate the research article identification problem as the problem of identifying *research articles* from a collection of crawled documents. In this work, we also refer to research articles as research papers or simply papers. This problem can be cast as a document or webpage classification, where a webpage is in fact a text document (a pdf or ps file) that will be classified as research paper or non-research paper.

Many approaches to *webpage classification* on the Web and in the context of digital libraries such as CiteSeer^x [14] and ArnetMiner [19], have been studied in the literature. For example, Qi and Davison [16] used content-based term features and HTML structure-based features for classifying webpages. Shen et al. [18] proposed the use of summarization algorithms to improve the performance of webpage classification. Chekuri et al. [4] studied webpage classification in order to improve the precision of web search. Kan and Thi [12] proposed the use of URLs in performing fast webpage classification.

Craven et al. [7] introduced the WebKB dataset that contains webpages collected from computer science departments of four universities. The webpages are classified into seven categories: *student*, *faculty*, *staff*, *department*, *course*, *project*, and *other*. The goal of creating this web knowledge base was to enable more effective retrieval of information on the Web. Blum and Mitchell [1] used WebKB to identify *course* webpages from the entire collection. They proposed co-training, an approach for semi-supervised learning that makes use of unlimited amounts of unlabeled data when the number of labeled examples available for training is limited.

The problem of researcher homepage classification is an instance of webpage classification. Tang et al. [19] studied homepage acquisition from search engine results using researcher names as queries. Gollapalli et al. [9] proposed the use of co-training on URL and content-based features for homepage classification and sought to apply focused crawling using a seed list of academic websites (where researchers' homepages are typically hosted) to acquire such a collection. Focused crawling, which was first proposed by Bra et al. is a rich area of research on the Web [2, 6].

In contrast to these works, we present a supervised machine learning approach to identifying the functionality of a "webpage" or a file on the Web as being research paper or non-research paper. Specifically, we propose a novel set of structural features that result in fast identification of research papers. We show that these features outperform the commonly used "bag of words" for text classification. To our knowledge, this is the first attempt to formulate the problem of identifying research papers on the Web. Currently, CiteSeer^x uses the above mentioned rule-based learner for identifying research papers, i.e., pdf or ps files on the Web are valid research documents if a "references" or "bibliography" section occurs in them [8].

3. FEATURES FOR RESEARCH ARTICLE IDENTIFICATION

We propose novel features for the identification of research articles from a crawled collection of documents. These features are general enough and, with small refinement, can be applied to other tasks such as the classification of documents into several classes: papers, slides, curriculum vita, theses, handouts, newsletters, etc. The proposed features are listed in Table 1. We identify four types of features, depending on their scope: file specific features, text specific features, section specific features, and containment features.

File specific features refer to the characteristics of the file, i.e., the size of the file in kilobytes and the number of pages in a document. The intuition is that generally, research articles are smaller in size and have smaller number of pages than other documents, e.g., theses, patents, etc.

Text specific features refer to the specifics of the text of a document and include the length of the text in characters, the number of words and number of lines in a document, the average numbers of words and lines per page, the percentage of references and reference mentions throughout a document, the percentage of space and non-alphanumeric characters, the length of the shortest line divided by the length of the longest line, the number of lines that start with uppercase letters or non-alphanumeric characters. For example, papers have more lines per page on average than slides.

Section specific features refer to the section names and determine if specific sections such as "abstract", "in-

Table 1: The list of structural features.

Feature Name	Description
File Specific Features	
FileSize	The size of the file in kilobytes
PageCount	The number of pages of the document
Text Specific Features	
DocLength	Length of the document in characters
NumWords	. . . in the number of words
NumLines	The number of lines in the document
NumWordsPg	The average number of words per page
NumLinesPg	. . . lines per page
RefRatio	The number of references and reference mentions throughout a document divided by the total number of tokens in a document
SpcRatio	The percentage of the space characters
SymbolRatio	. . . of words that start with non-alphanumeric characters
LnRatio	Length of shortest line divided by length of longest line in the document
UcaseStart	The number of lines that start with uppercase letters
SymbolStart	. . . with non-alphanumeric characters
Section Specific Features	
Abstract	Document has section "abstract"
Introduction	. . . "introduction" or "motivation"
Conclusion	. . . "conclusion"
Acknowledge	. . . "acknowledgement" or "acknowledgment"
References	. . . "references" or "bibliography"
Chapter	. . . "chapter"
Containment Features	
ThisPaper	Document contains "this paper"
ThisBook	. . . "this book"
ThisReport	. . . "this report"
ThisThesis	. . . "this thesis"
ThisManual	. . . "this manual"
ThisStudy	. . . "this study"
ThisSection	. . . "this section"
TechRep	. . . "technical report" or "tr-NUMBER"

troduction", "conclusion", "acknowledgements", "references" and "chapter" appear in a document.

Containment features refer to the containment of specific words (or keywords) such as "this paper", "this book", "this report", "this thesis", "this manual", "this study", "this section", and "technical report", in a document.

4. EVALUATION OF THE PROPOSED FEATURES

To evaluate the proposed features, we randomly sampled two independent sets of documents from CiteSeer^{x}: one of 1000 documents sampled directly from the crawled documents, which we refer to as **Crawl**, and another one of 1500 documents sampled from CiteSeer^{x} that passed the "references" or "bibliography" filter, which we refer to as **CiteSeer^{x}**. The CiteSeer^{x} crawler automatically downloads pdf and ps files from the Web using a dedicated focused crawler. The crawler starts from a list of seed URLs that are selected from a *whitelist* [21]. We describe the manual labeling process first and then present the experimental design and results.

Table 2: Datasets description.

Dataset	Number of Docs	NumDocs with Text	Positive Exp	Negative Exp
Crawl	1000	833	352	481
CiteSeer^{x}	1500	1409	811	598

We manually labeled the documents in each of the two datasets using the following labeling scheme: *positive* (+1 or P) corresponds to documents that are research articles including papers in conference proceeding, journal articles, press releases, book chapters, and technical reports; *negative* (-1 or N) corresponds to all other documents including books, theses, long technical documentation of more than 50 pages, slides, posters, incomplete papers/books (e.g., a references list, preface, table, abstract), brochures (e.g., a company introduction, circular, ad, product manual, government report, meeting notes, policy, form instruction, code, installation guide), handouts, homework, schedule, agenda, news, form, flyer, syllabus, class notes, letters, curriculum vita, resumes, memos, speeches.

We used the PDFBox[1] to extract the text from the pdf documents. We ignored the scanned documents and other documents for which the text was not correctly extracted. The statistics of the datasets are shown in Table 2. After completing the manual labeling of all documents in the two samples, we found 352 positive examples and 481 negative examples in the **Crawl** dataset and 811 positive examples and 598 negative examples in the **CiteSeer^{x}** dataset. As seen, a significant fraction of non-papers were indexed in CiteSeer^{x} by simply using the "references" or "bibliography" filter.

4.1 Experimental Design

Our experiments are designed around the following questions. *How does the performance of classifiers trained using the proposed structural features compare with that of "bag of words" classifiers and the "references" rule-based learner?* For "bag of words" (BoW), we first preprocess the data by removing punctuation, stop words, and performing stemming. In addition, we filtered out words with document frequency (*df*) less than 10. We experimented with several values for *df* and found *df* = 10 to give best results for BoW. The features used in each case are the following:

- The 27 structural features, denoted by Str.
- A bag of 7443 and 15248 words (*tf-idf*) on **Crawl** and **CiteSeer^{x}**, respectively, denoted by BoW.
- The feature *References* used as input to a rule-based learner that identifies a document as paper based on the existence of the feature in the document.

We experimented with several classifiers: Support Vector Machine (SVM), Naïve Bayes (NB), Naïve Bayes Multinomial (NBM), Logistic Regression (LR), Decision Trees (DT), and Random Forest (RF), trained on the above features, and compared their performance on each dataset *independently* using 10-fold cross-validation. We used the Weka[2] implementation of these classifiers with default parameters.

[1] http://pdfbox.apache.org/

[2] http://www.cs.waikato.ac.nz/ml/weka/

The next question is: *Do the classifiers trained on the structural features generalize well on new unseen data?* We trained our classifiers on one dataset and tested them on the other dataset. Since these datasets are independent of each other, we expect to provide an accurate estimate of the performance of structural-based classifiers on documents in CiteSeer^x to improve data quality.

The last question is: *Among the structural features, what are those that are most informative in identifying research articles from the crawled documents?* We ranked features using Information Gain.

To evaluate the performance of our models, we report Precision, Recall, and F-Measure for the positive class since we are mainly interested in the correct identification of papers, as well as the overall accuracy. Precision gives the fraction of papers correctly identified by a classifier among all papers identified by the algorithm, whereas Recall gives the fraction of papers correctly identified by the algorithm among all actual papers. F-Measure gives the harmonic mean of Precision and Recall. Accuracy gives the fraction of correctly classified examples from each set. Precision, Recall, and F-Measure are given as: $Precision = \frac{TP}{TP+FP}$, $Recall = \frac{TP}{TP+FN}$, $F\text{-}Measure = \frac{2 \cdot Precision \cdot Recall}{Prec + Recall}$, where TP, FP, FN represent the true positives, false positives, and false negatives, respectively. Given the sets of labeled documents, we can easily compute TP, FP, FN as follows: TP are documents identified as papers, which are indeed papers; FP are documents identified as papers, which in fact are not papers; FN are documents that are papers but are not identified by the algorithm as papers.

4.2 Results

Performance of classifiers trained on structural features. Table 3 compares the performance of various classifiers that use structural features with that of "bag of words" classifiers and a rule-based learner that uses the existence of the words "references" or "bibliography" to identify papers, on the **Crawl** dataset. Table 4 shows similar results on the **CiteSeer^x** dataset. As can be seen from Table 3, on **Crawl**, SVM using structural features outperforms the other classifiers in terms of Precision, F-Measure and Accuracy. For example, SVM(Str) achieves an F-Measure of 0.854 as compared to 0.845 achieved by LR, the next best performing classifier. Although NB(Str) has a higher Recall compared with SVM(Str), it achieves a much lower Precision of 0.703. SVM(Str) substantially outperforms the "references" rule-based learner in terms of all compared measures. For example, SVM(Str) achieves a Precision of 0.889 as compared to 0.764 achieved by the rule learner. In addition, SVM(Str) substantially outperforms both SVM(BoW) and NBM(BoW) in terms of Precision, at the expense of a lower Recall. However, Precision of both SVM(BoW) and NBM(BoW) is unacceptable in the context of digital libraries, resulting in many non-papers indexed in the database.

As shown in Table 4, on **CiteSeer^x**, using structural features, RF outperforms the other classifiers in terms of F-Measure and Accuracy, whereas SVM has a slightly higher Recall. Again, NB achieves a higher Recall, at the expense of a much lower Precision. Since the **CiteSeer^x** dataset contains documents from CiteSeer^x that have already passed the "references" filter, it is not surprising that the "references" rule learner achieves such a higher Recall of 0.942. However, the low values of Precision and Accuracy suggest that the number of false positives if very high. By inspecting the confusion matrix, we found that $TP = 764$, $FP = 506$, $TN = 92$, and $FN = 47$. Note that a naive classifier that would classify everything in the majority class has an accuracy of 57.55%. The performance of "bag of words" classifiers is substantially worse than that of structural features based classifiers in terms of all measures compared.

Table 3: Results on the Crawl dataset.

Feature/Classifier	Precision	Recall	F-Measure	Accuracy
Str/SVM	**0.889**	0.821	**0.854**	**88.11%**
Str/LR	0.880	0.813	0.845	87.39%
Str/NB	0.703	**0.886**	0.784	79.35%
Str/DT	0.853	0.807	0.829	85.95%
Str/RF	0.844	0.815	0.829	85.83%
BoW/SVM	0.59	0.912	0.717	69.50%
BoW/NBM	0.668	0.852	0.749	75.87%
References/Rule	0.764	0.79	0.777	80.79%

Table 4: Results on the CiteSeer^x dataset.

Feature/Classifier	Precision	Recall	F-Measure	Accuracy
Str/SVM	0.837	0.872	0.854	82.82%
Str/LR	0.830	0.877	0.853	82.54%
Str/NB	0.701	0.936	0.801	73.31%
Str/DT	0.829	0.864	0.846	81.90%
Str/RF	0.829	0.899	0.863	83.53%
BoW/SVM	0.713	0.650	0.680	64.79%
BoW/NBM	0.727	0.822	0.772	72.03%
References/Rule	0.602	0.942	0.734	60.75%

Figure 2 (left and middle plots) shows the Precision-Recall curves for **Crawl** and **CiteSeer^x**, respectively. As shown, SVM(Str) offers a higher Precision compared to NBM(BoW) and "references" learner for values of Recall larger than 0.8.

We conclude that the proposed features result in better performing models compared with BoW and "references" learner. Moreover, the number of structural features is significantly smaller than the number of words in BoW, i.e., 27 structural features and 7443 and 15248 words for **Crawl** and **CiteSeer^x**, respectively. The extraction of structural features took less than a minute for each dataset, whereas the extraction of dictionaries and feature encoding for BoW took 30 and 100 minutes on **Crawl** and **CiteSeer^x**, respectively, (potentially due to longer files), which is infeasible given the current size of CiteSeer^x. Next, we test the generalization performance of classifiers trained using our features.

Generalization performance of structural features based classifiers. Table 5 shows the performance of classifiers trained on **Crawl** and evaluated on unseen data from **CiteSeer^x**, obtained using a different sampling procedure. As shown in the table, LR outperforms the other classifiers in terms of Precision, F-Measure and Accuracy, but has a small drop in Recall compared with RF. NB has a much higher Recall, but Precision is about 10% lower than that of LR. It is worth noting that the performance of classifiers trained on **Crawl** and evaluated on **CiteSeer^x** (Table 5) is slightly worse than that of classifiers evaluated in a cross-validation setting on **CiteSeer^x** (Table 4). Figure 2 (right plot) shows the Precision-Recall curves for SVM and NB trained on **Crawl** and evaluated on **CiteSeer^x**, and for SVM evaluated on **CiteSeer^x** using cross-validation (CV). As shown, SVM-CV offers a slightly higher Precision compared to SVM and NB trained on **Crawl** and evaluated on

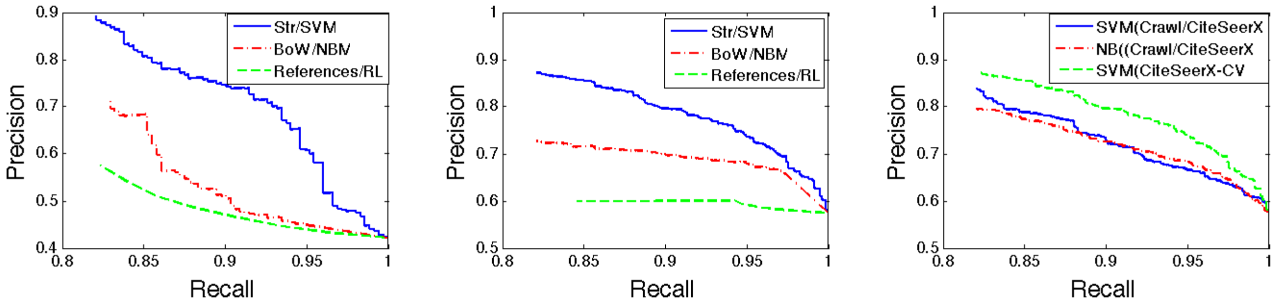

Figure 2: Precision-Recall curves.

Table 5: Performance of classifiers trained on Crawl and evaluated on CiteSeer^{x}.

Method	Precision	Recall	F-Measure	Accuracy
Str(SVM)	0.801	0.837	0.819	78.63%
Str(NB)	0.733	0.891	0.805	75.08%
Str(LR)	0.822	0.837	0.830	80.19%
Str(RF)	0.799	0.846	0.822	78.85%

Table 6: Top 15 ranked features by InfoGain.

	Crawl		CiteSeer^{x}	
Rank	IG Score	Feature Name	IG Score	Feature Name
1	0.296	RefRatio	0.2167	PageCount
2	0.283	References	0.1816	NumWords
3	0.283	DocLength	0.1771	DocLength
4	0.278	NumWords	0.1427	NumWordsPg
5	0.262	ThisPaper	0.1319	RefRatio
6	0.240	Abstract	0.1311	NumLines
7	0.213	NumLines	0.0943	FileSize
8	0.174	PageCount	0.0849	ThisPaper
9	0.163	NumWordsPg	0.0843	NumLinesPg
10	0.162	Introduction	0.0829	ThisManual
11	0.141	UcaseStart	0.0669	ThisThesis
12	0.135	Conclusion	0.0637	Chapter
13	0.125	NumLinesPg	0.0359	LnRatio
14	0.092	ThisSection	0.0329	ThisBook
15	0.085	FileSize	0.0308	ThisReport

CiteSeer^{x} for values of Recall larger than 0.8.

It is also interesting to note that the performance of all classifiers trained on **Crawl** and evaluated on **CiteSeer^{x}** (Table 5) outperform BoW evaluated in a cross-validation setting on **CiteSeer^{x}** (Table 4), suggesting the "goodness" of the proposed features in identifying papers, if deployed in a real-world scenario. We further investigate what are the most informative structural features for our problem.

Most Informative Features for Paper Identification. Table 6 shows the top 15 ranked structural features using Information Gain (IG) along with the IG score of each feature. As can be seen from the table, the most informative features on **Crawl** are the reference ratio, followed by "references", the length of the document in characters, the number of words and the containment of "this paper" in the document. The features Abstract, Introduction and Conclusion are also among the top 15 ranked features.

Table 7: Performance of classifiers on Crawl, using various sets of features.

Method	Precision	Recall	F-Measure	Accuracy
File specific	0	0	0	57.74%
Text specific	0.770	0.713	0.740	78.87%
Containment	0.839	0.696	0.761	81.51%
Section specific	0.779	0.790	0.784	81.63%
Containment+Sect.	0.910	0.719	0.803	85.11%
Text+ Section	0.858	0.804	0.830	86.07%
Containment+Text	0.832	0.719	0.771	81.99%
Containment+Text +Section	0.895	0.821	0.856	88.35%

We also analyzed the four types of features independently and in combination with each other. These results are shown in Table 7. The Section specific features result in highest F-Measure compared to the other features, whereas the combination of Text specific and Section specific features results in an increase in the performance of independent feature classifiers. The combination of Containment, Text specific and Section specific features results in the highest performance, suggesting that features extracted from all these scopes of a document are informative for identifying research articles.

5. ERROR ANALYSIS, DISCUSSION, AND FUTURE DIRECTIONS

We performed an error analysis and found that many papers are predicted as non-papers (FNs) when a document is in a foreign (non-english) language, which is a limitation of our models. Our goal is to focus only on english language documents. We experimented with a simple heuristic for filtering out foreign documents. Specifically, if a document contains stopwords such as "the", "a", "of", then it is considered as an english document. However, we found that many foreign language documents have one or more paragraphs in english, and hence, they passed the filter. Further investigation will be considered as part of future work.

Documents that are non-papers are generally predicted as papers (FPs) especially in the case of theses which are about 50 pages in lengths. Similar to papers, theses contain sections such as "abstract", "acknowledgements", "introduction", "conclusion", "references". In addition, theses have other particularities of research papers such as contain references throughout the text and may have similar average number of words per page. During the manual inspection of errors, we also found that training guides that have a similar structure as papers are mistakenly identified as research articles. Also, documents for which the text or the section names were not correctly extracted by PDFBox were wrongly identified as papers.

Another source of errors is for documents that contain only a list of references without the body of the document. Given that the features "reference ratio" and "references" that are ranked very high by IG , occur in the documents, it is not surprising that such documents are labeled as papers by our classifiers. In future, we will consider refining the set of structural features to clearly distinguish between papers, theses, reports, training guides, among other classes. In fact, a multi-class classification task that will classify documents based on their type would be interesting from a searching point of view, i.e., for improving searching of document types. In addition to the multi-class classification, we plan to evaluate the proposed structural features on larger samples of labeled data from both CiteSeer^x and on documents sampled directly from the crawl. We will deploy our models on CiteSeer^x data as well as on data crawled from the Web.

Since CiteSeer^x stores the URLs of the downloaded pdf or ps files, we plan to explore the use of information available in URLs to improve the task of identifying research articles from crawled documents. A preliminary analysis of the URLs corresponding to the documents in our datasets revealed that URLs can provide additional evidence for identifying research articles. For example, the occurrence of "pubs" or "papers" in the URLs from Table 8 is a clear indication that these URLs link to research articles. We plan to investigate semi-supervised approaches such as co-training that make use of unlabeled data readily available from the crawl. Specifically, we will design novel URL features and use them in conjunction with structural features as complementary views to improve the performance of classifiers on the task of identifying research articles.

Table 8: Example URLs pointing to research papers.

http://www.eecs.harvard.edu/ellard/pubs/ellard2004-disp.pdf
http://www.comp.nus.edu.sg/∼nght/pubs/www03.pdf
http://www.cs.berkeley.edu/∼krste/papers/fame-isca2010.pdf
http://tangra.si.umich.edu/∼radev/papers/167.pdf

6. CONCLUSION

We presented an approach to identifying research articles from a set of documents drawled on the Web to improve the quality of data stored and indexed in digital libraries, and used the CiteSeer^x digital library as a case study. Results of our experiments showed that the proposed features perform substantially better compared with the "bag of words" approach and a rule-based learner in cross-validation experiments as well as when tested on an independent set of documents from CiteSeer^x.

Among the proposed features, we found the top three most informative ones to be reference ratio, the occurrence of "references" or "bibliography" section, and the length of the text of a document in characters, as ranked by Information Gain on a set of documents crawled on the Web.

7. REFERENCES

[1] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT*, 1998.

[2] P. D. Bra, G. jan Houben, Y. Kornatzky, and R. Post. Information retrieval in distributed hypertexts. In *In RIAO*, 1994.

[3] C. Caragea, A. Silvescu, S. Kataria, D. Caragea, and P. Mitra. Classifying scientific publications using abstract features. In *SARA*, 2011.

[4] C. Chekuri, M. Goldwasser, P. Raghavan, , and E. Upfal. Web search using automated classification. In *Proc. of WWW*, 1997.

[5] H.-H. Chen, L. Gou, X. Zhang, and C. L. Giles. Collabseer: a search engine for collaboration discovery. In *Proceedings of JCDL '11*, 2011.

[6] J. Cho, H. Garcia-Molina, and L. Page. Efficient crawling through url ordering. In *WWW*, 1998.

[7] M. Craven, D. Dipasquo, D. Freitag, A. McCallum, T. Mitchell, and K. Nigam. Learning to extract symbolic knowledge from the world wide web. In *AAAI-98*, pages 509–516. AAAI Press, 1998.

[8] C. L. Giles, K. D. Bollacker, and S. Lawrence. Citeseer: An automatic citation indexing system. In *Proceedings of the Third ACM Conference on Digital Libraries*, DL '98, pages 89–98, 1998.

[9] S. D. Gollapalli, C. Caragea, P. Mitra, and C. L. Giles. Researcher homepage classification using unlabeled data. In *Proc. of WWW*, pages 471–482, 2013.

[10] S. D. Gollapalli, P. Mitra, and C. L. Giles. Similar researcher search in academic environments. In *JCDL*, 2012.

[11] Q. He, B. Chen, J. Pei, B. Qiu, P. Mitra, and C. L. Giles. Detecting topic evolution in scientific literature: how can citations help? In *CIKM*, 2009.

[12] M.-Y. Kan and H. O. N. Thi. Fast webpage classification using url features. In *CIKM*, 2005.

[13] O. Küçüktunç, E. Saule, K. Kaya, and Ü. V. Çatalyürek. Diversified recommendation on graphs: pitfalls, measures, and algorithms. In *WWW*, 2013.

[14] H. Li, I. G. Councill, L. Bolelli, D. Zhou, Y. Song, W.-C. Lee, A. Sivasubramaniam, and C. L. Giles. Citeseerx: a scalable autonomous scientific digital library. In *InfoScale*, 2006.

[15] R. M. Nallapati, A. Ahmed, E. P. Xing, and W. W. Cohen. Joint latent topic models for text and citations. In *Proceedings of KDD '08*, 2008.

[16] X. Qi and B. D. Davison. Web page classification: Features and algorithms. *ACM Comput. Surv.*, 41(2), Feb. 2009.

[17] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *Proceedings of UAI '04*, 2004.

[18] D. Shen, Z. Chen, Q. Yang, H.-J. Zeng, B. Zhang, Y. Lu, and W.-Y. Ma. Web-page classification through summarization. In *Proc. of ACM SIGIR*, SIGIR '04, pages 242–249, 2004.

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

[20] P. Treeratpituk and C. L. Giles. Disambiguating authors in academic publications using random forests. In *JCDL '09*, 2009.

[21] J. Wu, P. Teregowda, J. P. F. Ramírez, P. Mitra, S. Zheng, and C. L. Giles. The evolution of a crawling strategy for an academic document search engine: whitelists and blacklists. In *Proc. of the 3rd Annual ACM Web Science*, WebSci '12, pages 340–343, 2012.