Document Type Classification in Online Digital Libraries

Cornelia Caragea[1], **Jian Wu**[2], **Sujatha Das Gollapalli**[3], **and C. Lee Giles**[2]
[1]Department of Computer Science and Engineering, University of North Texas, Denton, TX
[2]College of Information Sciences and Technology, Pennsylvania State University, University Park, PA
[3]Institute for Infocomm Research, A*STAR, Singapore
ccaragea@unt.edu, jxw394@ist.psu.edu, gollapallis@i2r.a-star.edu.sg, giles@ist.psu.edu

Abstract

Online digital libraries make it easier for researchers to search for scientific information. They 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 highly depends on the accuracy of classifiers that identify the types of documents that are crawled from the Web, e.g., as research papers, slides, books, etc., for appropriate indexing. These classifiers in turn depend on the choice of the feature representation. We propose *novel features* that result in high-accuracy classifiers for document type classification. Experimental results on several datasets show that our classifiers outperform models that are employed in current systems.

Introduction

Online digital libraries such as Google Scholar, CiteSeer^x, ACL Anthology, ArnetMiner, and PubMed that store scientific documents or their metadata, are powerful resources for many applications that analyze scientific documents on a Web-wide scale. These applications include: topic classification of research papers (Caragea, Bulgarov, and Mihalcea 2015; Caragea et al. 2011), citation recommendation (Caragea et al. 2013; Küçüktunç et al. 2013; Huang et al. 2012; Kataria, Mitra, and Bhatia 2010), author name disambiguation (Tang et al. 2012; Treeratpituk and Giles 2009), expert search (Gollapalli, Mitra, and Giles 2012), scientific paper summarization (Mei and Zhai 2008; Qazvinian, Radev, and Özgür 2010), paper and slides alignment and generation (Hu and Wan 2013; Kan 2007), and automatic keyphrase extraction from research papers (Caragea et al. 2014a; Gollapalli and Caragea 2014).

To be successful, these applications require *accurate* collections of scientific documents. In systems where the collections are acquired automatically, the accuracy of such collections highly depends on the quality of a classifier that classifies documents crawled from the Web into their types, e.g., research papers, slides, books, etc., for appropriate indexing in digital libraries.

A rule-based system that classifies documents as research papers if they contain any of the words `references` or `bibliography`, as is currently in use by CiteSeer^x (Giles, Bollacker, and Lawrence 1998), will mistakenly classify documents such as curriculum vita or slides as research papers whenever they contain the word `references`, and will fail to identify research papers if they do not contain any of the two words. In contrast, the commonly used "bag of words" (BoW) or *tf-idf* representation for document classification may not capture the specifics of documents, e.g., due to the diversity of topics covered in digital libraries or the diversity of document types. As an example, a paper in Human Computer Interaction may have a different vocabulary space compared with a paper in Information Retrieval, but some essential terms may persist across the papers, e.g., "references" or "abstract." Even when the documents have a similar vocabulary space (e.g., a paper, its corresponding set of slides, and a thesis containing the paper may have similar or same words or word distributions), the BoW does not necessarily distinguish between the document types. In such cases, the number of tokens in documents could be very informative, e.g., the number of tokens in a research paper is generally much higher than in a set of slides, but much smaller than in a PhD thesis. These aspects are ignored by BoW models that only use terms in the documents' content.

Moreover, features extracted from the URLs that link to the crawled documents result in poor performing classifiers due to the weak signal in the URL strings. Although successful for Web applications, e.g., webpage classification and de-duplication (Gollapalli et al. 2013; Kan and Thi 2005; Koppula et al. 2010), the URL based features are not informative for the classification of documents according to their types due to the uncontrolled nature of document names or the lack of any hints or discriminative words in URLs.

Given the above shortcomings, one question that can be raised is: *Can we design features that capture the specifics of documents and result in models that accurately classify documents crawled from the Web into classes such as research papers, theses, books, slides, and curriculum vita?* We specifically address this question in this paper.

Contributions and Organization. We present an approach to classifying unstructured documents crawled from the Web into the above mentioned classes, using a small set of *novel* features, called *structural features*. The result of this classification task will aid indexing of documents in digital libraries, and hence, will lead to improved results in many applications. For example, accurate classification of documents' types is highly needed in retrieval systems, where one might be interested in searching for presentation slides on a particular topic, rather than searching for a research paper on that topic. To enable such typed-searches in a digital library such as CiteSeer^x, it is important to identify the type (thesis/slides/paper/etc.) of a crawled document. Accurate classification of documents' types can also benefit downstream processes. For example, it helps to avoid calculating an author's citation count from the citation mentions in the references lists of presentation slides. A system such as Google Scholar must integrate several components, e.g., paper/book/thesis/etc. classification, header and citation extraction, and author name disambiguation, in order to accurately display an author's citation count.

To our knowledge, the problem of document type classification using features extracted simultaneously from the content and the structure of documents has not been addressed in the literature, for document indexing in digital libraries. Our contributions are as follows:

- We show that classifiers trained using "bag of words" and URL features yield very low performance on the document type classification task.
- We propose novel structural features, which result in high accuracy classifiers. We show experimentally that these classifiers substantially outperform those trained using the "bag of words" and URL features.
- We evaluate the structural features on documents crawled from the Web using CiteSeer^x crawlers.
- Finally, we show experimentally that structural features based classifiers substantially outperform a rule-based learner that is used by current systems such as CiteSeer^x.

The rest of the paper is organized as follows. We first discuss related work, followed by the presentation of our proposed structural features. We then describe our data, present experiments and results and conclude the paper.

Related Work

Text classification is a well-studied problem. Comprehensive reviews of the feature representations, methods, and results on various text classification problems are provided by Sebastiani (2002) and Manning (2008). The "bag of words," *binary*, *tf* or *tf-idf* representations are commonly used as input to machine learning classifiers, e.g., Support Vector Machine and Naïve Bayes Multinomial. In the context of digital libraries such as CiteSeer^x and ArnetMiner, often the classes for text classification are document topics, e.g., "machine learning," "data mining" and "agents" (Lu and Getoor 2003). Different from this task, we are concerned with the classification of text documents crawled from the Web, into one of the following classes: *paper*, *book*, *slides*, *thesis*, *resume/CV*, and *others*, and design structural features that incorporate aspects specific to research documents.

Gosh and Mitra (2008) combine structural and content features for the supervised classification of XML documents according to the source of the documents. The authors show improvement over approaches that use either structural or content features independently. Chagheri et al. (2011) address the problem of identifying technical documentation such as user manuals and manufacturing documents that are available in electronic format and use a combination of terms and the structure of documents (i.e., the tags of XML documents) for classification. However, research documents have features that are not present in standard XML documents. For example, many research documents have both textual as well as citation information, and section boundaries (that are not common across all papers). In our previous work (Caragea et al. 2014b), we present a binary classification of documents crawled from the Web as research papers or not research papers, using only a small set of structural features. This binary classification is in the process to replace the rule-based learner currently available in the CiteSeer^x system (Wu et al. 2014; 2015). In contrast to the binary classification, here, we address a multi-class classification of documents and propose an extended set of structural features. Our long-term goal is to make the multi-class classification of documents as a constituent component of the retrieval system of CiteSeer^x.

Our task has similarities with the webpage classification that involves the identification of the type, genre, or topic of a webpage. Qi and Davison (2009) provide a survey on the content-based term features and HTML structure-based features typically used for classifying webpages. Shen et al. (2004) proposed the use of summarization algorithms to improve the performance of webpage classification. Chekuri et al. (1997) studied webpage classification in order to improve the precision of Web search. Kan and Thi (2005) proposed the use of URLs in performing fast webpage classification. URL features were also found to improve performance of author homepage classification (Gollapalli et al. 2013).

Unlike the above works, we present a supervised approach to classifying a "webpage" or a file from the Web according to its type. This problem faces many challenges including dealing with unstructured text, a large variety of domains, e.g., Computer and Information Sciences, with each domain having its own file templates, an uncontrolled nature of document names, and a high sparsity of URL hints. These challenges give rise to the unique design of our approach.

Structural Features for Classification

We identify four types of features (described below), 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 (*FileSize*) and the number of pages of a document (*PageCount*). The intuition is that generally scientific documents vary in size and number of pages, e.g., research papers are smaller in size and have smaller number of pages compared to documents such as theses and books.
- **Text (or document) specific features** refer to the specifics of the text of a document and include: the length of the text in characters (*DocLength*), the number of words (*NumWords*) and number of lines (*NumLines*) in a document, the average numbers of words and lines per

Dataset	Documents	Docs with Text	Books	Slides	Theses	Papers	CVs	Others
Train	1000	960	13	48	9	472	2	416
Test	1000	959	22	40	8	461	4	424
Train+	3284	3223	511	824	500	472	500	416

Table 1: Datasets description.

page (*NumWordsPg* and *NumLinesPg*, respectively), the average number of words per line (*NumWordsLn*), the percentage of references and reference mentions throughout a document (*RefRatio*) as well as their counts (*RefCount*), the percentage of spaces (*SpcRatio*), of words that start with capital letters (*UcaseRatio*), and those that start with non-alphanumeric characters (*SymbolRatio*), the length of the shortest line divided by the length of the longest line in a document (*LnRatio*), the number of lines that start with uppercase letters (*UcaseStart*) or non-alphanumeric characters (*SymbolStart*), and the number of words that appear before the References section (*TokBeforeRef*). These features encode our intuition about the crawled document templates. For example, papers have usually more lines per page on average than theses, which in turn have more lines per page on average than slides. A file that contains only a list of references has no words before the occurrence of the word "references," whereas a research paper contains a body of text before the word "references," which describes a scientific or technical problem.

- **Section specific features** refer to section names and their position in documents and determine if sections such as "abstract", "introduction", "conclusion", "acknowledgements", "references" and "chapter" appear in a document (denoted as *Abstract*, *Introduction*, *Conclusion*, *Acknowledgements*, *References*, and *Chapter*, respectively). We also consider the position of these sections in documents (*PosAbstract*, *PosIntroduction*, *PosConclusion*, *PosAcknowledgements*, and *PosReferences*) and the position of "acknowledgment" before or after "introduction" (*AckBeforeIntro*, *AckAfterIntro*). These features capture our intuition about the structure of the crawled documents in terms of their constituent sections and the sections' positions. For example, in a paper, the "acknowledgments" section generally occurs at the end, whereas, in a thesis, it occurs in the beginning, and may have no occurrence in a curriculum vitae or a set of slides.
- **Containment features** refer to the containment of specific words / phrases in a document. These include "this paper," "this book," "this thesis," "this chapter," "this document," "this section," "research interests," "research experience," "education," and "publications" (denoted as *ThisPaper*, *ThisBook*, *ThisThesis*, *ThisChapter*, *ThisDocument*, *ThisSection*, *ResInterests*, *ResExperience*, *Education*, *Publications*, respectively), as well as the position of "this paper," "this book," and "this thesis" in a document (*PosThisPaper*, *PosThisBook*, *PosThisThesis*). The intuition is that authors of scientific documents usually use "in this paper, . . ." or "this book, . . ." to describe what the document is about, with the occurrence of these words generally very early in the content of documents.

Datasets and Evaluation Measures

To evaluate the proposed features, we randomly sampled two independent sets of 1000 documents from our crawl data, collected using a dedicated CiteSeer^x focused crawler. The crawler starts with a list of preselected seed URLs, performs a breadth-first crawl and saves open-access PDF documents. We refer to these sets as Train(Crawl) and Test(Crawl), or short, **Train** and **Test**.

We manually labeled the documents from each set into one of the following classes: *Paper*, *Book*, *Thesis*, *Slides*, *Resume/CV*, and *Others*. For labeling, we used two annotators from our research labs. Whenever there was a disagreement between them, a third annotator was asked to select between the two annotations done by the two original annotators.

To extract the text from the PDF documents, we used PDFBox[1]. The scanned documents and other documents for which the text was not correctly extracted were ignored. The statistics of **Train** and **Test** are shown in Table 1.

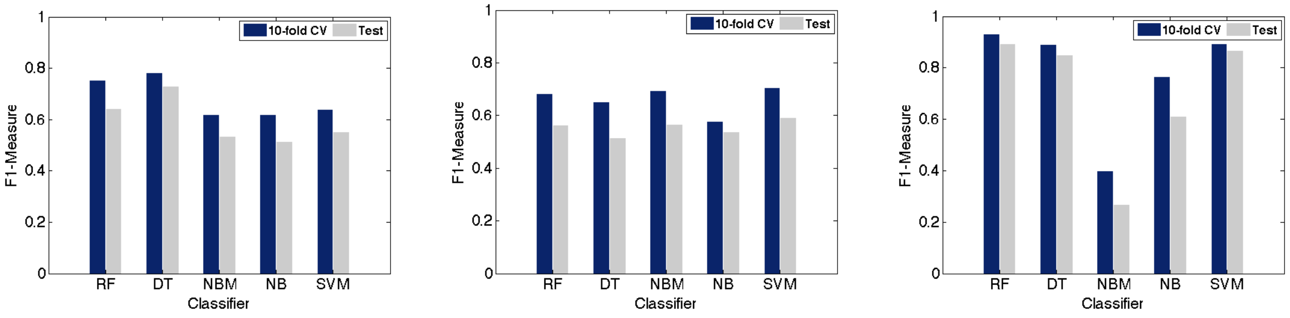
As can be seen from the table, all sets are highly unbalanced, with the majority of documents belonging to *paper* or *others* categories, whereas the categories *book*, *slides*, *thesis*, and *CV* are under-represented. To overcome this difficulty, we supplemented the **Train** set with ≈ 500 documents for each of the following categories: *book*, *thesis*, and *CV*, and ≈ 700 documents for the *slides* category. We refer to this set as **Train+**, which will be the training set in all of our experiments (see Table 1 for data statistics).

The sample of books is selected from the open-book search engine introduced by Wu et al. (2013). The sample of theses is selected from our crawl data, Google, and Bing, by querying "in partial fulfillment of the requirements." This query returns a large volume of online-theses, from which we randomly selected 500. The documents in the *CV* category are selected from the crawl data by searching "resume", "cv", or "curriculum vitae" in the title field, whereas the sample of slides is selected from Bing by searching for ppt documents. All these additional documents were manually inspected to ensure accurate labeling.

Results and Observations

How does the performance of classifiers trained using the proposed structural features compare with that of classifiers trained on "bag of words" extracted from the content of documents and features extracted from the URLs of the documents? For "bag of words" (BoW) from textual content, we first constructed a vocabulary from the terms in the training documents, and then, represented each document as a normalized term frequency (*tf*) or term frequency - inverse document frequency (*tf-idf*) vector. We preprocessed

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



(a) BoW(Content)

(b) URL features

(c) Str features

Figure 1: The weighted F1-Measure obtained with various classifiers on **Train+** (denoted 10-fold CV), and on **Test**, using (a) "bag of words" (BoW), (b) URL and (c) structural (Str) features.

the data to remove punctuation and stop words, and performed stemming. In addition, we kept only words that occurred in at least 5 documents (i.e., document frequency $df \geq 5$). We experimented with different *df* values, and found $df \geq 5$ to give the best results for BoW.

For URL features, we used a similar feature extraction as in (Gollapalli et al. 2013). We extracted features from the entire URL, including the domain name of a page. We used "/" as the delimiter for tokenization, and constructed the term vocabulary from all unigrams and bigrams that occurred at least twice in the URL strings from the training set. We also used regular expressions to capture URL string patterns, e.g., *alpha-characters_year*, which could cover the name of a venue and the year of publication, hyphenated or underscore words, and the presence of words such as *publications*, *pub*, *pubs*, *publ*, *papers*, *slides*.

The features used for training classifiers are as follows:

- The 43 structural features, denoted by Str.
- A bag of 61,655 words (*tf-idf*) extracted from the textual content of documents in **Train+**, denoted by BoW.
- A set of 2,692 features extracted from the URL strings of documents in **Train+**, denoted by URL.

We experimented with several classifiers: Random Forest (RF), Decision Trees (DT), Naïve Bayes Multinomial (NBM), Naïve Bayes (NB), and Support Vector Machines with a linear kernel (SVM), trained on the above features. We used the Weka implementation of these classifiers. We tuned model hyper-parameters in 10-fold cross-validation experiments on **Train+**, whenever applicable (e.g., the C parameter in SVM and the number of trees in RF).

Figures 1 (a), (b), and (c) show the F1-Measure obtained by the above classifiers on **Train+** (in 10-fold cross-validation experiments), and on **Test** (training on **Train+**), using BoW, URL, and Str features, respectively. For each classifier, the best parameter setting obtained on **Train+** (in 10-fold cross-validation) was used on **Test**. For example, the number of trees in RF that gave the best results on **Train+** using 10-fold cross-validation with Str features was 21. Hence, 21 trees in RF were used on **Test** with Str.

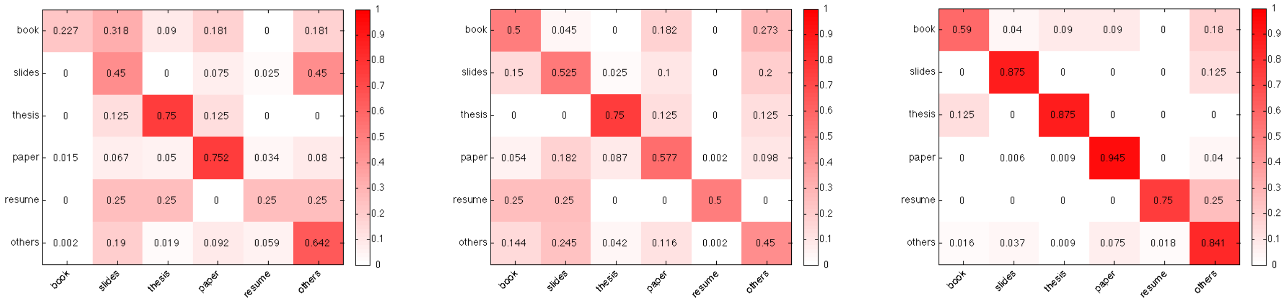
Table 2 shows the performance (Precision, Recall, F1-Measure and Accuracy) on both **Train+** and **Test** for each feature type, BoW, URL, and Str, with the classifiers that give the best results for the corresponding feature type.

Feature/Classifier	Precision	Recall	F1-Measure	Accuracy
Train+ (10-fold CV)				
BoW/DT	0.781	0.782	0.781	78.21%
URL/SVM	0.706	0.708	0.704	70.77%
Str/RF	**0.928**	**0.928**	**0.928**	**92.83%**
Test				
BoW/DT	0.801	0.677	0.726	67.67%
URL/SVM	0.741	0.518	0.590	51.82%
Str/RF	**0.901**	**0.887**	**0.891**	**88.73%**

Table 2: Results on **Train+** and **Test** with best classifiers for each feature type.

As can be seen from Figure 1 and Table 2, DT performs best on BoW(Content), SVM performs best on URL features, whereas RF performs best on Str features. As expected, the performance of classifiers on **Test** is slightly worse than that of classifiers on **Train+** (in 10-fold cross-validation experiments). For example, Str with RF achieves 0.928 F1-Measure on **Train+** and 0.891 F1-Measure on **Test**. However, the results on **Test** provide better estimates of classifiers' performance given its construction, which reflects the natural distribution of the data, i.e., data that will be encountered at test time in a real-world scenario. Note that **Train+** contains additional files for categories *books*, *theses*, *CVs* and *slides*. Thus, since **Test** is a randomly sampled set from the crawled documents, it is likely to reflect the average properties of the entire crawl.

Our proposed structural features, Str with RF, have the highest overall performance, among all feature types and all classifiers, on both **Train+** and **Test**. BoW, which is commonly used for text/topic classification, achieves a much worse performance compared with that of Str. For example, BoW achieves only 0.726 F1-Measure on **Test**, whereas Str achieves 0.891 F1-Measure on the same set. This can be explained by the fact that BoW contains significant noise from the point of view of document type classification. Str features effectively harness the information that is relevant for our classification task, resulting in 18.5% relative improve-



(a) Bag of Words (Content)

(b) URL features

(c) Structural features

Figure 2: Confusion matrices for: (a) BoW with Decision Trees (DT), (b) URL with Support Vector Machines (SVM), and (c) Str with Random Forest (RF), obtained on the Test dataset.

ment in F1-Measure over BoW.

The URL features have the worst overall performance. For example, the highest F1-Measure achieved by URL features is 0.590 on **Test**, using SVM, whereas the highest F1-Measure achieved by Str features is 0.891 on the same set, using RF, which gives 33.7% relative improvement in F1-Measure from URL to Str. The poor performance of URL features could be explained by the weak signal in the URL strings with respect to the type of documents that are hosted at these URLs. We took a closer look at the URLs in our datasets. Several URL examples from our data along with the category of the hosted documents are shown in Table 3.

Although some URLs contain very good hints for the type of the hosted documents, e.g., paper, thesis, and slides (examples [1] - [4] in the table), we found that a large fraction of URLs are rather impossible to discern even for a human (examples [5] - [8] in the table), having no obvious hints about the type of documents that these URLs link to. For example, in [5], someone with knowledge in academic search can confidently guess that www.cs.tau.ac.il/∼azar/ is a researcher's homepage, but would have difficulties to infer that the file "node.pdf" linked from the homepage refers to a research paper, and not to a set of slides or a reference manual. In addition to the above examples, we found many URLs that have similar URL string patterns, but belong to different categories (see examples [9] - [12] in Table 3).

We continued our data analysis by investigating the confusion matrices of the best classifiers for each feature type. Figures 2 (a), (b), and (c) show these matrices for BoW with DT, URL with SVM and Str with RF, respectively, on **Test**. The darker the red in an entry on the main diagonal, the more correctly classified examples in the category corresponding to that entry. The off diagonal red entries reflect misclassified examples, with darker red showing more misclassifications. Figure 2 also shows that Str features yield best classification results compared with BoW and URL features.

URLs hinting to the hosted document type
paper
[1] homes.cs.washington.edu/∼pedrod/papers/uai11c.pdf
[2] www.cs.berkeley.edu/∼krste/papers/fame-isca2010.pdf
slides
[3] cs.cmu.edu/∼ggordon/10601/slides/Lec04_GM_annot.pdf
[4] usenix.org/legacy/events/slaml10/tech/slides/schneider.pdf
URLs with no clear hints
paper
[5] www.cs.tau.ac.il/∼azar/node.pdf
[6] www.cslab.ece.ntua.gr/∼dtsouma/index_files/swim2012.pdf
slides
[7] www.dtic.mil/ndia/2012CMMI/W14923_Beckett.pdf
[8] www.ecb.int/paym/groups/pdf/fxcg/icap_ecb_240610.pdf
URLs with similar surface patterns, but different categories
paper
[9] www.comp.nus.edu.sg/∼nght/pubs/www03.pdf
slides
[10] www.ece.msstate.edu/∼sherif/pubs/DRE.pdf
others (homework assignment)
[11] www.cs.vu.nl/∼vdvorst/pde2013a6.pdf
paper
[12] www.public.asu.edu/∼afrieden/ecta5602.pdf

Table 3: Example URLs from our datasets.

On the task of identifying research papers, how do structural features based classifiers compare with a rule-based learner, which is used by current systems such as CiteSeer^x? We investigate the performance of our features in comparison with the rule-based learner for identifying research papers from another sample of 1000 documents from the CiteSeer^x crawl data. Each document in this sample contains at least one occurrence of either "references" or "bibliography." We refer to this set as **Test_References**. We manually annotated this sample as before. The number of documents by class in **Test_References** are as follows: 7 books, 8 slides, 26 theses, 831 papers, 0 CVs, and 128 *others*.

Table 4 compares the performance of RF trained using Str features with the rule-based learner on **Test_References**. The results for RF with Str features are shown for the category *paper* (from the multi-class classification). As can be seen from the table, both methods have a high Recall (note that the Recall for the rule-based learner is less than 1 because the words "references" and "bibliography" are not correctly extracted from the PDF of a few documents by PDFBox). RF with Str achieves a much higher Precision, i.e., almost 10% boost in Precision, compared with the rule-based learner, suggesting that the structural features are able to substantially reduce the number of false positives.

Feature/Classifier	Precision	Recall	F1-Measure
Str/RF	0.925	0.970	0.947
References/Rule	0.842	0.974	0.903

Table 4: Results on **Test_References**.

Conclusion

We addressed the classification of documents, crawled from the Web, according to their document type, e.g., *paper*, *slides*, *book*, *thesis*, *resume/CV*, and *others*. We proposed novel structural, text density, and layout features that are designed to incorporate aspects specific to research documents. Experimental results showed that the proposed features outperform the "bag of words" and URL features for document type classification. Compared with a rule-based learner that classifies documents as research papers if they contain any of the words "references" or "bibliography" (as currently employed in CiteSeer^{x}), the classifiers trained on the proposed structural features result in a higher precision. Thus, our high-accuracy classifiers represent a substantial improvement over the state-of-the-art approaches used in existing systems such as CiteSeer^{x} and will aid information search and retrieval.

In future, it would be interesting to explore the document type classification, where the document types are organized in a hierarchy. For example, slides could be further classified as slides corresponding to a conference paper, slides corresponding to an invited talk, and course or lecture slides.

Acknowledgments

We thank our anonymous reviewers for their constructive feedback. This research is supported in part by a collaborative grant from the National Science Foundation.

References

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

Caragea, C.; Silvescu, A.; Mitra, P.; and Giles, C. L. 2013. Can't see the forest for the trees? a citation recommendation system. In *Proceedings of JCDL '13*.

Caragea, C.; Bulgarov, F.; Godea, A.; and Gollapalli, S. D. 2014a. Citation-enhanced keyphrase extraction from research papers: A supervised approach. In *Proc. of EMNLP '14*.

Caragea, C.; Wu, J.; Williams, K.; Gollapalli, S. D.; Khabsa, M.; Teregowda, P.; and Giles, C. L. 2014b. Automatic identification of research articles from crawled documents. In *Web-Scale Classification Workshop, co-located with WSDM (WSC 2014)*.

Caragea, C.; Bulgarov, F.; and Mihalcea, R. 2015. Co-training for topic classification of scholarly data. In *Proc. of EMNLP '15*.

Chagheri, S.; Calabretto, S.; Roussey, C.; and Dumoulin, C. 2011. Document classification: combining structure and content. In *ICEIS*.

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

Ghosh, S., and Mitra, P. 2008. Combining content and structure similarity for xml document classification using composite svm kernels. In *Proc. of ICPR'08*, 1–4.

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

Gollapalli, S. D., and Caragea, C. 2014. Extracting keyphrases from research papers using citation networks. In *AAAI '14*.

Gollapalli, S. D.; Caragea, C.; Mitra, P.; and Giles, C. L. 2013. Researcher homepage classification using unlabeled data. In *Proc. of WWW'13*, 471–482.

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

Hu, Y., and Wan, X. 2013. Ppsgen: Learning to generate presentation slides for academic papers. In *Proc. of IJCAI'13*.

Huang, W.; Kataria, S.; Caragea, C.; Mitra, P.; Giles, C. L.; and Rokach, L. 2012. Recommending citations: translating papers into references. In *Proceedings of CIKM '12*.

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

Kan, M.-Y. 2007. Slideseer: a digital library of aligned document and presentation pairs. In *Proc. of JCDL '07*.

Kataria, S.; Mitra, P.; and Bhatia, S. 2010. Utilizing context in generative bayesian models for linked corpus. In *AAAI'10*.

Koppula, H. S.; Leela, K. P.; Agarwal, A.; Chitrapura, K. P.; Garg, S.; and Sasturkar, A. 2010. Learning url patterns for webpage de-duplication. In *Proc. of WSDM'10*, 381–390.

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

Lu, Q., and Getoor, L. 2003. Link-based classification. In *ICML*.

Manning, C. D.; Raghavan, P.; and Schütze, H. 2008. *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press.

Mei, Q., and Zhai, C. 2008. Generating impact-based summaries for scientific literature. In *Proceedings of ACL-08: HLT*, 816–824.

Qazvinian, V.; Radev, D. R.; and Özgür, A. 2010. Citation summarization through keyphrase extraction. In *Proc. of COLING'10*.

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

Sebastiani, F. 2002. Machine learning in automated text categorization. *ACM Computing Surveys* 34(1):1–47.

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

Tang, J.; Fong, A.; Wang, B.; and Zhang, J. 2012. A unified probabilistic framework for name disambiguation in digital library. *IEEE TKDE* 24(6):975–987.

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

Wu, J.; Williams, K.; Chen, H.-H.; Khabsa, M.; Caragea, C.; Ororbia, A.; Jordan, D.; and Giles, C. L. 2014. Citeseerx: Ai in a digital library search engine. In *IAAI 2014, co-located with AAAI 2014*.

Wu, J.; Killian, J.; Yang, H.; Williams, K.; Choudhury, S. R.; Tuarob, S.; Caragea, C.; and Giles, C. L. 2015. Pdfmef: A multi-entity knowledge extraction framework for scholarly documents and semantic search. In *K-Cap 2015*.

Wu, Z.; Mitra, P.; and Giles, C. L. 2013. Table of contents recognition and extraction for heterogeneous book documents. In *ICDAR*, 1205–1209.