Using ACM DL Paper Metadata as an Auxiliary Source for Building Educational Collections

Yinlin Chen
Department of Computer Science
Virginia Tech
Blacksburg, VA 24061
ylchen@vt.edu

Edward A. Fox
Department of Computer Science
Virginia Tech
Blacksburg, VA 24061
fox@vt.edu

ABSTRACT

Some digital libraries harvest metadata records from multiple content providers to build their collections. However, the quality and quantity of such metadata records are limited by what is harvested. To ensure collection growth, and to expand the scope beyond just what can be harvested, additional content acquisition methods are needed. Accordingly, we discuss how the Ensemble project (a pathway effort in the NSDL) is broadening its collection with the help of machine learning. Since Ensemble aims to aid computing education, we make use of ACM Digital Library records as a resource to help with transfer learning. We have built classifiers that can identify if a potential additional resource is about computing education. We approached this as a cross-domain text classification problem and developed suitable methods for feature extraction and bootstrapping for classifier training. Our experiments on three datasets of computing education metadata records show our approach can enhance the quality and quantity of records being added to Ensemble.

Categories and Subject Descriptors

H.3.7 [**Digital Libraries**]: Collection; I.5.2 [**Design Methodology**]: Classifier design and evaluation

General Terms

Design, Experimentation

Keywords

Digital Library, transfer learning, classification, computing education

1. INTRODUCTION

Some digital libraries harvest metadata records from multiple content providers to build their collections. Ensemble, a multi-university project funded by NSF to add a computing education portal to the NSDL family of STEM pathways [1], is this kind of DL. Currently, Ensemble has 5290 metadata records in 25 computing collections which are harvested from organizations and universities involved in computing education that have agreed to serve as content providers. These resources are collected in the Dublin Core and NDSL_DC formats.

To ensure Ensemble growth, and to expand the scope beyond just what can be harvested, additional content acquisition methods are needed. We are certain that there are large numbers of online educational resources, not yet in Ensemble, that both have good quality and could be of use in learning about computing. Examples of sources for such resources include YouTube and SlideShare.

To find such resources, we employ machine learning, i.e., we develop a suitable classifier. Since it is expensive to manually construct training sets for each target source, e.g., YouTube, we have developed an alternate approach based on the technique called transfer learning [2]. The basic idea is to train a classifier in a source domain that has a large amount of training data, and then to transfer that learning to the target domain.

Thanks to support from ACM, we have been able to use the ACM Digital Library to provide training data. Since we need to support browsing by topic in Ensemble, we also chose the ACM Computing Classification System (CCS) as a source of categories for Ensemble resources.

We considered building one classifier to determine if a potential educational resource was indeed of educational value, and another to determine if a resource was related to computing. However, it was not clear how to build the first classifier. Accordingly we adopted the simpler approach of constructing a single classifier for computing education, using the ACM DL, with positive training examples coming from the part of the ACM DL that relates to computing education, e.g., is associated with SIGCSE or is categorized under "computing education". The remaining challenges included validation of that classifier with the ACM DL, transferring the learning to target domains (e.g., Ensemble and YouTube), and devising bootstrapping methods to gain additional improvements.

Section 2 summarizes related work. Section 3 describes the datasets employed. Section 4 describes our approach and evaluation. Section 5 concludes this paper.

2. RELATED WORK

Transfer learning [2] involves leveraging the knowledge of source domains to help train a classifier for a target domain. There are many real-world applications that use transfer

978-1-4799-5569-5/14/$31.00

learning techniques, such as text classification [3]. The transfer learning approaches are categorized into *instance-transfer*, *feature-representation-transfer*, *parameter-transfer*, and *relational-knowledge-transfer*. The *feature-representation-transfer* approach is to find a "good" feature representation that minimizes the differences between source and target domains [4]. Our study falls into this category.

Our problem setting can be considered as learning with auxiliary data, where the ACM DL is treated as the auxiliary data source. We used traditional machine learning approaches in our previous work [5]. In this study we address the problem as a cross-domain text classification problem. The cross-domain classification is related to transfer learning, where the knowledge built from a given task is used to solve another learning task. [6] used an EM-based Naive Bayes classifier to solve a cross-domain text classification task. Similar to our research, [7] proposed a method to integrate text classification with Wikipedia. They built an auxiliary text classifier which can classify documents, with the most relevant articles of Wikipedia, and then use the bag of words representation with new features corresponding to the titles represented by the Wikipedia articles.

3. DATASETS

We use three datasets in this study, with metadata from the ACM DL, Ensemble, and YouTube. This section summarizes key characteristics of those datasets, and how we worked with each.

3.1 ACM DL

Our copy of the ACM DL has conference papers and journal articles from 1954 to 2013. There are 1,761,956 metadata records, classified according to the 2012 ACM Computing Classification System (2012 CCS), a six-level hierarchical topic tree. Each paper in the ACM DL was manually classified by its author(s) according to CCS.

ACM provided metadata in support of our research, which we loaded into Solr. We constructed a training set by issuing category queries to the Solr server. We chose "CCS->Social and professional topics->Professional topics->Computing education" (computing education) to build a classifier to identify computing education records. We only used metadata records with both title and abstract information for our classification task. Ultimately, we had 11,422 papers in the computing education category as positive examples. We randomly selected 11,400 other papers from the collection as negative examples to train our classifier.

3.2 Ensemble

The Ensemble portal has 25 collections which are either harvested from content providers or from user contributed resources. There are three collections that are all about computing education. The CSTA collection is about K-12 Computer Science Teaching and Learning Materials. The Computing and Information Technology Interactive Digital Educational Library (CITIDEL) collection also contains computing education records. The Nifty collection contains assignments, handouts, starter code, and projects for CS educators. We combined these three computing education collections into a ground truth dataset with 590 records to test the classifier which was trained using ACM DL metadata.

Table 1: Evaluation of computing education classifiers using ACM DL metadata

Classifier	Precision	Recall	F1 measure
NB	0.89	0.928	0.909
NB + SW	0.896	0.922	0.909
NB + SM	0.89	0.928	0.909
NB + SW + SM	0.896	0.922	0.909
NB + SW + IG	0.907	0.913	0.91
NB + SW + CS	0.904	0.914	0.909

3.3 YouTube

YouTube.com contains million of videos, so we sought to examine our classifier's ability to find educational videos about computing. We implemented a crawler using the Google YouTube data API. We collected 660 records from the YouTube education channel, each including the title and description.

4. PILOT CLASSIFICATION STUDY

In this study, we are particularly interested in how we can use transfer learning to identify new resources using the existing dataset we have. We examined our approach to utilize the knowledge (learning model) which was built using resources from one source domain (ACM DL metadata), and applied it to the target domain (Ensemble / YouTube). In this section, we describe in detail our classification process, common feature selection, and bootstrapping process.

4.1 Classifier Building

We used ACM metadata records under the "Computing education" category in our training dataset to build the computing education classifier. We extracted distinct term vectors over the combination of title and abstract. We removed stopwords using the Stanford stopword list (SW) and stemmed the remaining terms using Porter's Snowball algorithm (SM) [8]. We used two feature selection methods, Information Gain (IG) [9] and Chi-squared (CS) [10], and used the first 200 significant words. We used the Naive Bayes (NB) algorithm [11] to train the classifier. Each classifier was validated via 10-fold cross-validation. Table 1 shows the performance of these classifiers using the "Computing education" metadata set as input. The classifier with best performance used Naive Bayes, stop word removal, and information gain.

After the best performing classifier was identified, we used Ensemble collection metadata as the test set to evaluate the classifier. The classifier's accuracy decreased from 91.27% to 76.55%. In order to increase the classifier accuracy on the target domain (Ensemble collection), we used the *feature-representation-transfer* transfer learning approach to find a "good" feature representation that reduces differences between source and target domain.

4.2 Common Features Selection

In transfer learning settings, we assume that there is a relationship between source and target domains. In this study, our assumption is that both source and target domains share some similarities since all are computing education resources. In order to find the similar features between ACM DL metadata and Ensemble records, and ACM DL metadata and YouTube records, we first used Information

Table 2: Top 10 words for each of ACM DL, Ensemble, and YouTube

ACM DL	Ensemble (1)	YouTube (2)
students	paper	lecture
computer	students	paper
science	professional	science
teaching	instructional	computer
courses	materials	videos
education	proposed	learn
curriculum	show	model
learning	development	algorithm
university	operating	programming
programming	performance	present

Table 3: Significant words in common for both ACM DL and Ensemble or YouTube

ACM DL & Ensemble	ACM DL & YouTube
algorithm	algorithm
computer	computer
courses	courses
curriculum	development
education	engineering
introductory	learning
performance	linear
programming	method
project	performance
school	programming

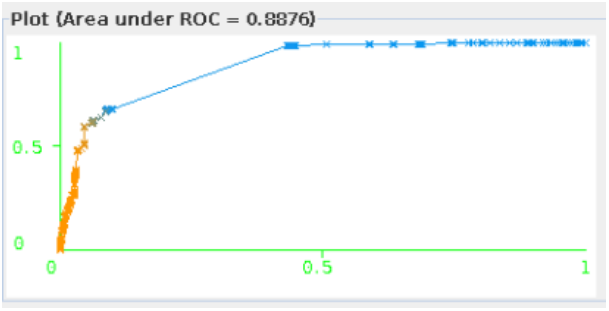

Figure 1: ROC curve for computing education classifier with features that are significant words in common, trained on ACM DL and applied to Ensemble records

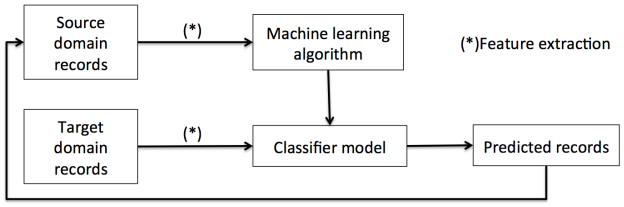

Figure 2: Bootstrapping in the target domain to improve the classifier for adding Ensemble educational records

Gain (IG) and Chi-squared (CS) to get ranked lists from each dataset, and implemented a program to find common words in these lists. Finally, we used these common features to train a classifier and evaluated the classifier's performance. Table 2 shows the top 10 word matches between ACM and the (1) Ensemble list and (2) the YouTube list. Table 3 shows the top 10 common significant words between ACM DL and Ensemble, and ACM DL and YouTube.

After the common significant words list was generated, we used the top 20 words as our features when training with ACM DL metadata and used Ensemble records as the test set to evaluate the classifier's performance. The classifier's performance increased from 76.55% to 77.75% with 0.907 precision and 0.734 F-Measure. Figure 1 shows the area under an ROC curve; the classifier's ROC curve is .8876, which would be considered "good". We used the same procedure to classify YouTube records; the classifier's performance is 78.36% with 0.917 precision and 0.763 F-Measure. We examined the confusion matrix. Although precision is acceptable, the recall is low. In order to increase the performance, we proposed bootstrapping, discussed in the following section.

4.3 Bootstrapping Approach

According to the results of our experiments, the classifier's performance will be affected by the difference between source domain and target domain. In order to reduce the difference, we used a bootstrapping process to iteratively retrain the classifier by adding predicted target domain records into source domain records. Figure 2 shows this bootstrapping process.

We first used source domain records to build a classifier using the training algorithm. Second, we used the built classifier to predict target domain records and get a new set of predicted records. We added these new predicted records into the original source domain records and used feature selection methods to compute new feature sets and a new classifier was trained. During each iteration, the classifier's performance was increased and new predicted records were the new records to be added into the DL repository.

To test this proposed bootstrapping approach, we used ACM DL metadata as source domain records and YouTube metadata as target domain records. To prepare the YouTube metadata records, we used our YouTube crawler tool and used the category terms under "CCS->Social and professional topics->Professional topics->Computing education" as search terms to search YouTube. Figure 3 shows the search terms we used to generate a set of queries to search on YouTube in order to get video metadata records. An example of one of these queries is "Computing education computational thinking".

In each iteration, we used a classifier to predict 50 YouTube records and used the YouTube metadata set we used in section 4.2 as test set to evaluate the classifier's performance. In each iteration, we added new predicted positive records to source domain records and built a new classifier for the next interation. We applied the classifier trained in each iteration to the test set, and used 10-fold cross-validation to compute the classifier's accuracy. The experiment result appears in Table 4.

Our experiments shows some interesting results. During the common feature selection, we found that authors tend to use similar words to describe their education records. Thus, the training dataset with lots of labeling data can give us lots of positive features to train classifiers with high performance.

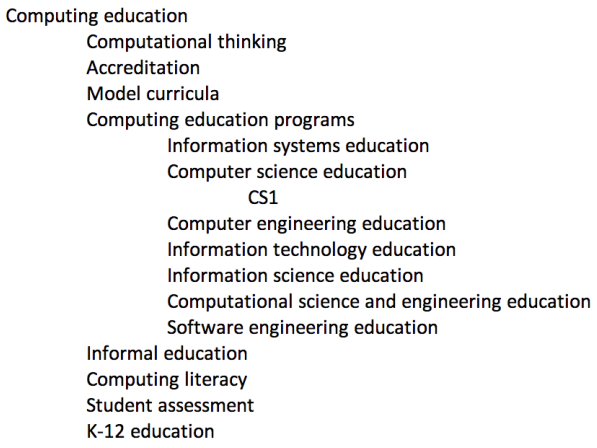

Figure 3: Search terms from ACM CCS used to get metadata records from YouTube

Table 4: Bootstrapping iterations showing improved classifier accuracy

Iteration	Classifier accuracy	New records
1	78.36%	34
2	78.74%	34
3	79.12%	36

Choosing common features between source and target domain records will affect the original classifier's performance due to some significant words not being selected during the training process. The newly trained classifier's performance on predicting target domain records is increasing and it is able to predict new unseen data. The bootstrapping experiment result shows that the classifier's performance was increased through each iteration and new records are also identified.

5. CONCLUSIONS

In this study, we show how we use ACM DL metadata as auxiliary source and adapt the transfer learning approach to build a useful classifier for predicting computing education records from different domains of resources. Our bootstrapping approach shows a performance improvement of the classifier, and new computing education records are identified. Our preliminary experiments provided us with valuable insights for moving forward.

We intend to further verify this approach and build more topic-specific classifiers. Currently we are building classifiers for ACM categories under "Security and privacy" and "Design and analysis of algorithms". We use an offline learning algorithm which trains the classifier in batch mode; we intend to explore and use an online learning algorithm and perform online learning. Our application interacts with a backend Solr server to obtain training data for classification building, thus we can easily deploy our application into our Hadoop cluster environment in order to process big data collections harvested from multiple resources. Finally, we will integrate our classifiers with harvesting tools and automate the overall collection building process.

6. ACKNOWLEDGMENTS

This research is supported by NSF Grants DUE-0840713, 0840715, 0840719, 0840721, 0840668, 0840597, 0836940, and 0937863. Our thanks go to ACM for providing us the ACM DL metadata to use in this research.

7. REFERENCES

[1] E. A. Fox, Y. Chen, M. Akbar, C. A. Shaffer, S. H. Edwards, P. Brusilovsky, D. Garcia, L. Delcambre, F. Decker, D. Archer, R. Furuta, F. Shipman, S. Carpenter, and L. Cassel, "Ensemble PDP-8: Eight principles for distributed portals," in *JCDL '10*, pp. 341–344, ACM, 2010.

[2] S. J. Pan and Q. Yang, "A survey on transfer learning," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 22, pp. 1345–1359, Oct 2010.

[3] P. Wang, C. Domeniconi, and J. Hu, "Using wikipedia for co-clustering based cross-domain text classification," in *Data Mining, 2008. ICDM '08. Eighth IEEE International Conference on*, pp. 1085–1090, Dec 2008.

[4] R. Gupta and L. Ratinov, "Text categorization with knowledge transfer from heterogeneous data sources," in *Proceedings of the 23rd National Conference on Artificial Intelligence*, pp. 842–847, 2008.

[5] Y. Chen, P. L. Bogen, II, H. Hsieh, E. A. Fox, and L. N. Cassel, "Categorization of computing education resources with utilization of crowdsourcing," in *JCDL '12*, pp. 121–124, ACM, 2012.

[6] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu, "Transferring naive bayes classifiers for text classification," in *Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 1*, AAAI'07, pp. 540–545, 2007.

[7] E. Gabrilovich and S. Markovitch, "Feature generation for text categorization using world knowledge," in *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, IJCAI'05, pp. 1048–1053, 2005.

[8] M. F. Porter, "Readings in information retrieval," ch. An Algorithm for Suffix Stripping, pp. 313–316, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997.

[9] M. Dash and H. Liu, "Feature selection for classification," *Intelligent Data Analysis*, vol. 1, no. 1 - 4, pp. 131 – 156, 1997.

[10] H. T. Ng, W. B. Goh, and K. L. Low, "Feature selection, perceptron learning, and a usability case study for text categorization," *SIGIR Forum*, vol. 31, pp. 67–73, July 1997.

[11] G. H. John and P. Langley, "Estimating continuous distributions in bayesian classifiers," in *Eleventh Conference on Uncertainty in Artificial Intelligence*, (San Mateo), pp. 338–345, Morgan Kaufmann, 1995.