Automatic Subject Classification of Scientific Literature Using Citation Metadata

Abdulhussain E. Mahdi and Arash Joorabchi

Department of Electronic and Computer Engineering, University of Limerick, Ireland
{hussain.mahdi,arash.joorabchi}@ul.ie

Abstract. This paper describes a new method for automatic classification of scientific literature archived in digital libraries and repositories according to a standard library classification scheme. The method is based on identifying all the references cited in the document to be classified and, using the subject classification metadata of extracted references as catalogued in existing conventional libraries, inferring the most probable class for the document itself with the help of a weighting mechanism. We have demonstrated the application of the proposed method and assessed its performance by developing a prototype software system for automatic classification of scientific documents according to the Dewey Decimal Classification (DDC) scheme. A dataset of one thousand research articles, papers, and reports from a well-known scientific digital library, CiteSeer, were used to evaluate the classification performance of the system. Detailed results of this experiment are presented and discussed.

Keywords: Digital library organization, scientific literature classification, library classification schemes, Dewey Decimal Classification (DDC), library Online Public Access Catalogues (OPACs).

1 Introduction

Scientific digital libraries and repositories are a fast-growing concept within research and academic communities. The main aim of these services is to facilitate effective dissemination of research output among researchers by providing efficient centralized access points to large collections of research data in electronic format, mainly available in form of articles, papers, technical reports, thesis, and dissertations. Metadata, generally defined as data about data, plays a critical rule in digital libraries by providing structured data about characteristics of unstructured data resources. It can significantly improve the accessibility of resources by helping to describe, locate, and retrieve them efficiently. Hence, utilizing data mining and knowledge discovery techniques to create, enrich, and harvest metadata has been one of the main efforts of researchers working in the field of digital libraries. The focus of this work is on automatic generation of a specific type of metadata called classification metadata in scientific digital libraries, aimed at defining the content subject of archived resources according to a standard library classification scheme.

The rest of the paper is organized as follows: Section 2 discusses the role of classification metadata in digital libraries and reviews existing Automatic Text

E. Ariwa and E. El-Qawasmeh (Eds.): DEIS 2011, CCIS 194, pp. 545–559, 2011.

Classification (ATC) methods for generating classification metadata in digital libraries. Section 3 provides an outline of our proposed ATC method called Bibliography based ATC (BB-ATC). Section 4 describes a prototype ATC system which has been developed based on the proposed method in order to demonstrate its viability and evaluate its performance in organizing a scientific digital library. Section 5 describes the evaluation process and presents its results. Section 6 provides a conclusion along with a summary account of planned future work.

2 Classification Metadata

Medium to large-scale digital libraries contain tens to hundreds of thousands of items, and therefore require advanced querying and information retrieval techniques to facilitate precision search and discovery of archival materials. In order to deliver highly relevant search results, we need to go beyond the traditional keyword-based search techniques which usually yield a large volume of indiscriminant search results irrespective of their content. Subject classification of materials in digital libraries according to a standard scheme could improve the accuracy of information retrieval significantly and allows users to browse the collection by subject [1]. However, manual subject classification of documents is a tedious and time-consuming task which requires an expert cataloguer in each knowledge domain represented in the collection, and therefore deemed impractical in many cases. Motivated by the ever-increasing number of e-documents and the high cost of manual classification, Automatic Text Classification/Categorization (ATC) - the automatic assignment of natural language text documents to one or more predefined classes/categories according to their contents - has become one of the key methods to enhance the information retrieval and knowledge management of digital textual collections.

Since the early ’90s, with the advances in the field of Machine Learning (ML) and the emergence of relatively inexpensive high performance computing platforms, ML-based approaches have become widely associated with modern ATC systems. A comprehensive review of the application of ML algorithms in ATC, including the widely used Bayesian Model, *k*-Nearest Neighbor, and Support Vector Machine, is given in [2]. In general, an ML-based ATC algorithm uses a corpus of manually classified documents to train a classification function which is then used to predict the classes of unlabelled documents. Applications of such algorithms include spam filtering, cataloguing news articles, and classification of web pages, to name a few. However, although a considerable success has been achieved in above listed applications, the prediction accuracy of ML-based ATC systems depends on a variety of factors, and no single ATC algorithm is adequate for all purposes.

On the other hand, as Golub [3], Yi [4], and Markey [5] discuss, there exits a less investigated approach to ATC that is attributed to the library science community. This approach focuses less on algorithms and more on leveraging comprehensive controlled vocabularies, such as library classification schemes and thesauri which have been developed and used for manual classification of holdings in conventional libraries. A library classification system is a coding system for organizing library materials according to their subjects with the aim of simplifying subject browsing.

Library classification systems are used by expert library cataloguers to classify books and other materials (e.g., serials, audiovisual materials, computer files, maps, manuscripts, realia) in conventional libraries. The two most widely used classification systems in libraries around the world today are the Dewey Decimal Classification (DDC) [6] and the Library of Congress Classification (LCC) [7], which since their introduction in the late 18th century have undergone numerous revisions and updates.

A promising avenue for the application of this approach is the automatic classification of resources archived in digital libraries, where using standard library classification schemes is a natural and usually most suitable choice because of the similarities between conventional and digital libraries. In general, ATC systems that have been developed based on the above library science approach can be divided into two main categories:

1. String matching-based systems: these systems do not rely on ML algorithms to perform the classification task. Instead, they use a method which involves string-to-string matching between words in a term list extracted from library thesauri and classification schemes, and words in the text to be classified. Here, the unlabelled incoming document can be thought of as a search query against the library classification schemes and thesauri, and the result of this search includes the class(es) of the unlabelled document. One of the well-known examples of such systems is the Scorpion project [8] by the Online Computer Library Centre (OCLC) [9]. Scorpion is an ATC system for classifying e-documents according to the DDC scheme. It uses a clustering method based on term frequency to find the most relevant classes to the document to be classified. A similar experiment was conducted by Larson [10] in early 90's, who built normalized clusters for 8,435 classes in the LCC scheme from manually classified records of 30,471 library holdings and experimented with a variety of term representation and matching methods. For more examples of these systems see [11, 12].
2. ML-based systems: these systems utilize ML algorithms to classify e-documents according to library classification schemes such as the DDC and the LCC. They represent a relatively unexplored trend which aims to combine the power of ML-based ATC algorithms with the enormous intellectual effort that has already been put into developing library classification systems over the last century. Chung and Noh [13] built a specialized web directory for the field of economics by classifying web pages into 757 sub-categories of economics category in the DDC scheme using *k*-NN algorithm. Pong et al. [14] developed an ATC system for classifying web pages and digital library holdings based on the LCC scheme. They used both *k*-NN and Naive Bayes (NB) algorithms and compared the results. Frank and Paynter [15] used the linear SVM algorithm to classify over 20,000 scholarly Internet resources based on the LCC scheme.

In this work, we propose a new category of ATC systems within the framework of the library science approach, which we call Bibliography Based ATC (BB-ATC) and is based on utilizing the citation networks among documents. We demonstrate and evaluate the application of the proposed method in the automatic generation of subject classification metadata for documents archived in scientific digital libraries.

3 Outline of Proposed BB-ATC Method

A considerable amount of documents have some form of linkage to other documents. For example, it is a common practice in scientific documents to cite related papers, articles, and books. It is also common practice for documented law cases to refer to other cases, patents to refer to other patents, and webpages to have links to other webpages. Leveraging these networks of citations/links among documents opens a new route for the development of ATC systems, known as collective classification [16]. Our proposed BB-ATC method falls into this route, and aims to develop a new trend of effective ATC systems that are based on leveraging:

1. The intellectual work that has been put into developing and maintaining extensive resources and systems for classifying and organizing the vast amount of materials archived in conventional libraries.
2. The intellectual effort of expert library cataloguers who have used the above classification resources and systems to manually classify and index millions of books and other materials in libraries around the world over the last century.

With the assumption that the majority of materials, such as books and journals, cited in a scientific document belong to the same or closely relevant classification category(ies) as that of the citing document, we can classify the citing document based on the class(es) of its references as identified in existing conventional library catalogues. The proposed BB-ATC method is based on automating this process using three main steps:

1. Identifying and extracting references in the document to be classified.
2. Searching for and retrieving the subject classification metadata of referenced materials from the online public access catalogues (OPACs) of conventional libraries.
3. Inferring and allocating a class(es) to the document based on the retrieved subject classification metadata of referenced materials with the help of a weighting mechanism.

This method of classification is applicable to any document that cites one or more published materials catalogued in at least one of the OPACs searched by the system. Examples of such documents include books, conference and journal articles, learning and teaching materials (e.g., syllabi and lecture notes), theses, and dissertations. In [17] the authors have described an ATC system designed and developed for automatic classification of electronic syllabus documents based on an early version of the BB-ATC method proposed here. Also, in [18] we have applied the underlying idea of the BB-ATC method to the problem of automatic keyphrase indexing of scientific documents which could be viewed as a multi-label text classification problem.

4 System Implementation and Functionality

In order to demonstrate the application of the proposed BB-ATC method for automatic generation of subject classification metadata in scientific digital libraries, we have developed a prototype ATC system for categorising the scientific documents archived in CiteSeer digital library [19] according to the DDC scheme. CiteSeer is a scientific literature digital library focusing primarily on the literature in computer science and information technology, and it contains over one million documents. We chose CiteSeer as our experimental platform for two main reasons:

1. CiteSeer is a well-known scientific digital library among the information science and digital library research communities and has been the subject of various studies in the areas of information retrieval and digital libraries.
2. It is an open-access and open-source project providing full access to all of its resources including: metadata records, archived items, and software source codes.

Our developed ATC system is effectively a metadata generator comprising a pre-processing, a data mining, and an inferring unit. The complete collection of CiteSeer's metadata records is freely available on the project's website[1] in the form of dump files. CiteSeer metadata records come in two different types: Open Access Initiative (OAI) records in Dublin Core XML format and bibliographic records in BibTex format. These two types of metadata records associated to each archived document contain a wide range of metadata about the document such as: type, title, authors, abstract, references, publishing date, publisher, source URL, format, language, etc. In order to easily access this large collection of metadata records we first developed a small software component to normalize and convert the CiteSeer BibTex records into XML format. Then the CiteSeer OAI and BibTex records in XML format were loaded into a native XML database called eXist-DB [20] which supports XML query languages, Xquery and Xpath, and facilitates efficient search and retrieval of CiteSeer metadata records.

The initial task of the pre-processing unit is to select a document from the CiteSeer archive for classification and retrieve its metadata from the CiteSeer metadata database for further processing. The selection can be sequential, random, or based on some criteria, such as publishing date, number of references, format, etc. Once a document is selected and its metadata is retrieved, the pre-processing unit compiles a list of titles of all the publications referenced in the document, such as articles, books, reports, etc., as per the list of references provided in the CiteSeer OAI metadata record of the document. The retrieved metadata of the document along with its list of references are then passed to the data mining unit.

The task of the data mining unit is two folds. In the first stage, it uses the Google Books Search (GBS) engine [21] to compile a list of publications that either cite the document to be classified or one of its references. This is done by submitting a number of URL queries to the GBS engine in the following format:

```
http://www.google.com/books/feeds/volumes?max-
results=20&q=%22[title]%2C%22
```

For the first query, the variable *title* in above format is set to the title of the document to be classified, and in the subsequent queries the titles of the references in the document are used consecutively. The parameter *max-results* limits the number of returned matching results to twenty items. This parameter is set empirically to balance the bias in the search results in terms of the number of returned matching publications for different queries. The returned result for each query is an XML file containing the

[1] http://citeseer.ist.psu.edu/

metadata records of matching publications and each record contains a set of metadata elements such as: title, authors, ISBN, etc. At this point, we have a pool of metadata records for the publications that either cite the document to be classified or one of its references. In order to utilise the gathered metadata for inferring the DDC class of the document, we first need to discover the DDC classification numbers of the publications in the pool. This is achieved by the second stage of the data mining process, where the corresponding DDC numbers of publications in the pool are retrieved from the OCLC's WorldCat [22] database. WorldCat is a union catalogue of about 70,000 conventional libraries around the world. The data mining unit performs this task in two steps. First, it processes the metadata records of the publications in the pool to extract their corresponding ISBNs. These ISBNs are then used as unique identifiers for the publications to query the WorldCat database for their corresponding metadata records. The latter process is done by submitting the following URL query to the WorldCat Search API [23] per each ISBN:

```
http://www.worldcat.org/webservices/catalog/content/isb
n/[ISBN]%3Fwskey%3D[key]%3Dfull
```

The returned result for each query is and XML file containing the full bibliographic record of the publication in MARC 21 XML format [24]. Along with other metadata elements, this record contains a DDC classification number assigned to the publication by a professional library cataloguer in one of the 70,000 libraries that have merged their catalogue into the WorldCat catalogue.

The task of the inferring unit is to analyze the pool of metadata gathered by the data mining unit, which contains the DDC numbers potentially related to the document to be classified, and select a DDC number from the pool which is most probable to represent the document's core subject. The inference process is based on a weighting method designed to assign a relevance probability score to each unique DDC number in the pool according to its frequency distribution.

Initially, the weighting method assigns each unique DDC number in the pool three different weights: un-normalized local frequency, normalized local frequency, and global frequency. Each of these weights is designed to measure the relevance probability of a given DDC number in the pool in relation to the document from a unique perspective. We describe these weights and details of the inferring process in the course of the following example which gives a sequential account of how the proposed BB-ATC method is used to classify a sample document from the CiteSeer archive. The document used in this example is a research paper entitled "Statistical Learning, Localization, and Identification of Objects". The core subject of the document is AI-based computer vision and, therefore, it should be classified into the DDC class "Computer science, information & general works\Computer science, knowledge & systems\Special computer methods\Artificial intelligence\Computer vision" represented by the DDC number 00637. The classification process of this sample document would be as follows:

4.1 Pre-processing

The pre-processing unit retrieves the corresponding metadata records for the document to be classified from the CiteSeer metadata database. Table 1 shows some of the retrieved metadata for the sample document.

Table 1. Sample document's metadata

Metadata field	Font size and style
dc:title	Learning, Localization, and Identification of Objects
datestamp	1996-08-06
dc:description	This work describes a statistical approach to deal with learning and recognition problems in the field of computer vision…
dc:identifier	http://citeseer.ist.psu.edu/52.html
oai_citeseer:relation type="References"	<oai_citeseer:uri>oai:CiteSeerPSU:112462</oai_citeseer:uri>

4.2 Data Mining

As described earlier, this process involves compiling a list of publications that either cite the document to be classified or one of its references, and discovering their corresponding DDC numbers. As the last row of Table 1 shows, the document under classification either has only one reference, or the CiteSeer's citation extraction unit, ParsCit [25], which is responsible for extracting citations from the archived documents, has only managed to extract one of the references successfully. Therefore, the title of the document to be classified and the title of its single successfully extracted reference are the only available clues that can be used for mining a list of DDC numbers potentially relevant to the document. Table 2 shows the metadata gathered by the data mining unit for the publications that cite one of these two titles.

Table 2. Data mining results for the sample document

Publications citing the document to be classified titled: "Statistical Learning, Localization, and Identification of Objects"					
ISBN	DDC No.	ISBN	DDC No.	ISBN	DDC No.
0123797721	006.37	3540650806	006.3	0818681845	621.367
0123797772	006.37	3540629092	006.42	3540639314	621.367
3540646132	006.37	3540634606	006.42	0792378504	621.367
0780350987	006.37	389838019X	005.118	3540250468	629.8932
0769501648	Null	1558605835	Null	0780399781	Null
Publications citing the document's reference titled: "Learning Object Recognition Models from Images"					
ISBN	DDC No.	ISBN	DDC No.	ISBN	DDC No.
3540617507	006.37	1586032577	006.3	389838019X	005.118
0195095227	006.37	3540282262	006.3	1848002785	621.367
3540667229	006.37	3540634606	006.42	3540433996	629.892
3540404988	006.37	3540636366	006.7	0818638702	621.399
0120147734	537.56	0780399773	Null		

4.3 Inferring

The inference process starts by deriving the un-normalised local frequency, normalised local frequency, and global frequency weights for each unique DDC number in the pool, as per the following:

- The un-normalised Local Frequency (ULF) of a given DDC number, DDC_i, is defined as the summation of its frequencies in each of the search result sets, R_j, where $j = \{1,\ldots, m\}$ with m being the total number of search result sets:

$$ULF(DDC_i) = \sum_{j=1}^{m} Freq(DDC_{i,j}) \tag{1}$$

  The function $Freq(DDC_{i,j})$ returns the number of times that the DDC number, DDC_i, appears in the search result set j. For a given DDC number, DDC_i, which appears in the pool of search results at least once, $ULF(DDC_i)$ is an integer number greater than or equal to 1. For example, the result of data mining process for the sample document appearing in Table 2 shows that there are 15 publications citing the document to be classified and another 14 publications citing the document's only reference. Among this total of 29 publications, 8 are assigned the DDC number "006.37", and therefore the ULF value for this DDC number is equal to 8.
- In order to prevent a DDC number from unjustifiably biasing the inference result due to its overwhelming high frequency in a single or small number of search result sets, we have adopted a second weight called Normalised Local Frequency (NLF) defined as:

$$NLF(DDC_i) = \sum_{j=1}^{m} \frac{Freq(DDC_{i,j})}{|R_j|} \tag{2}$$

  where, $|R_j|$ represents the total number of DDC numbers in the search result set R_j. For a given DDC number, DDC_i, which appears in the pool of search results at least once, $NLF(DDC_i)$ is a positive real number greater than 0. For example, using the sample data given in Table 2, the NLF value for the DDC number "006.37" is (4/12) + (4/13) = 0.64.
- The third weight, Global Fequency (GF), aims to reflect how common a given DDC number is among all the search result sets irrespective of its frequency inside individual search result sets. The GF for a given DDC number, DDC_i, is defined as the total number of search result sets in which DDC_i appears once or more:

$$GF(DDC_i) = \sum_{j=1}^{m} \left[DDC_i \in R_j \right] \tag{3}$$

  where, $[DDC_i \in R_j]$ returns 1 if DDC_i appears in the search result set R_j at least once, and returns 0 otherwise. For a given DDC number, DDC_i, which appears in the pool of search results at least once, $GF(DDC_i)$ is a positive integer number, with a minimum value of 1 and a maximum value of m, with m being the total number of search result sets. Again, using the sample data given in Table 2, the DDC number "006.37", for example, appears in both R_1 and R_2 search result sets, and therefore its GF is equal to 2.

Having computed the ULF, NLF, and GF weights for a given DDC number, *DDC_i*, the formula in Equation 4 is used to derive a single Combined Weight (CW) for it:

$$CW(DDC_i) = GF(DDC_i) \times NLF(DDC_i) \times ULF(DDC_i)^{\frac{depth(DDC_i)}{10}+1} \qquad (4)$$

where, *depth(DDC_i)* returns the vertical position of DDC_i in the classification hierarchy. The formulas for the ULF, NLF, and GF weights of a given DDC number in the pool and the CW formula used to derive a single combined weight from them, have been empirically deduced to give the best inference results based on an extensive analysis of a preliminary dataset. The results of this analysis indicated that the impact of ULF on CW should be kept to a minimum for the DDC numbers at the first level of the DDC hierarchy and it should gradually increase as the depth/level of the given DDC number increases in the hierarchy. The last part of Equation 4 incorporates this condition. Sticking to the DDC number "006.37" in our example and using the data of Table 2, the CW for this DDC number is computed as: $2 \times 0.64 \times 8^{(5/10)+1} = 29.01$.

After computing the above weights for all the DDC numbers in the pool, the inferring unit builds a classification hierarchy tree from all the DDC numbers in the pool and their corresponding weights. This tree is then automatically inspected to find the most probabilistically relevant DDC number to the core subject of the document. The inferring unit uses Java Universal Network/Graph Framework (JUNG) [26], which is an open source software library for graph modelling, analysis, and visualisation, to build, crawl, and visualise the classification tree.

The automatic crawling process aims to find the strongest path in the classification tree based on the CW values of the nodes. It starts from the root/start node and moves to the child node which has the largest CW value as the probabilistically selected most relevant DDC number to the document in the first level of the DDC classification hierarchy. The same selection criterion is then applied to the children of the selected node and so on until a node with no children (i.e. a leaf node) is reached. In cases where all the children of a chosen node have equal CW values, the CWs of its grandchildren are compared and the grandchild which has the largest CW value along with its parent node becomes selected. If all the grandchildren of the chosen node have equal CW values, then the decision will be based on the CW's of its great grandchildren and so on. In rare cases where this selection criterion does not lead to a resolution and all of the descendents of the latest chosen node have equal CW values in their corresponding level of the classification hierarchy, the crawling process stops and the latest selected node becomes the final selected DDC number for the document.

During our preliminary experiments, we noticed some cases where there is a significant decrease in the CW value of a potentially chosen node in relation to its parent's CW value. In majority of studied cases, this sudden drop indicated that either the latest chosen node (i.e., the parent node of current potentially chosen node) is the most appropriate DDC number for the document or there is not enough evidence to confidently conclude otherwise. In these cases, the best policy is to stop the crawling process and output the latest confidently chosen node, i.e., the parent node, as the final selected DDC number for the document. This policy is incorporated into the inference process in the form of a thresholding mechanism which stops the crawling process if a potentially chosen node does not pass the criterion in Equation 5, and outputs the parent of that node as the final selected DDC number for the document.

$$CW(CN) \times depth(CN)^{children(PN)} > CW(PN) \tag{5}$$

where, *CW(CN)* is the *CW* value of the current potentially chosen node, *depth(CN)* is the depth of the current potentially chosen node in the DDC hierarchy, *children(PN)* is the number of the parent node's children (i.e., the number of the current potentially chosen node's siblings added by one), and *CW(PN)* is the *CW* value of the parent node.

The prototype BB-ATC system operates in two modes: unsupervised and semi-supervised/evaluation. In unsupervised mode, classification process of a document ends by adding its final chosen DDC number to its metadata record stored in the CiteSeer metadata database. In the semi-supervised mode, however, first, the built classification hierarchy tree and the inference result for the document are visualized and presented; and then the user is required to either confirm the DDC number suggested by the system for the document as the most appropriate, or enter the correct DDC number manually. Once the results are confirmed/corrected, both the DDC numbers chosen by the inferring unit and the user are added to the metadata record of the document stored in the CiteSeer metadata database. In parallel to that, when operating in evaluation mode, the system creates a HTML log file for each classified document containing its original metadata, data mining results, and manual and automatic generated subject classification metadata.

5 System Evaluation and Experimental Results

Evaluating the performance of the developed prototype BB-ATC system was the most challenging and time consuming part of this work. To start with, the CiteSeer digital library, used as the test platform in this work, does not provide any subject classification metadata for its archived items. In fact, to the best of our knowledge, there exit no digital library of scientific literature which classifies its collection according to a standard library classification scheme, such as the DDC or the LCC. This fact, as discussed in Section 2, can be attributed to two main obstacles: the first is the high cost of manual classification, and the second is the inefficiency of common ML-based ATC systems to cope with the sheer size of library classification schemes, containing thousands of classes. Therefore, in the absence of any suitable third-party test corpus, we had no option but to create our own.

To perform the evaluation, the pre-processing unit of the system was set to randomly retrieve the metadata records of one thousand documents from the CiteSeer metadata database to be automatically classified and manually examined by a group of five postgraduate students in our research group. The students were given access to the WebDewey[2] which is part of the Online Computer Library Centre (OCLC) [9] suite of cataloguing and metadata services and enables full browsing of the latest version of the DDC online. The students were first familiarized with the DDC scheme and its hierarchical nature, and then each were assigned a set of documents (as defined below) to examine and classify.

[2] `http://www.oclc.org/dewey/versions/webdewey`

In order to measure the effect of the number of references successfully extracted from documents on the classification performance of our system, the pre-processing unit was set to build the test corpus from five different subsets of documents grouped according to their number of references. Each subset is made up of 200 documents with equal number of references. The first subset contains the documents that have no references successfully extracted from them. The second, third, forth, and fifth subsets contain documents with 4, 8, 16, and 32 references, respectively. Also, we set the inferring unit of the system to work in the semi-supervised/evaluation mode, which requires the user to either verify or rectify the DDC number automatically assigned to the document, and logs all the data produced during the classification process of the document in a dedicated HTML log file, as explained previously in Section 5. The HTML log files for all of the 1000 test documents used in this experiment may be viewed online on our webpage[3].

We used the standard measures of Precision (*Pr*), Recall (*Re*), and *F1* to evaluate the classification performance of our system. Micro-average and macro-average are the two wildly used measures to evaluate the overall prediction performance of ATC systems. In micro-averaging, the above target performance measures (i.e. *Pr*, *Re*, and *F1*) are computed globally over all classes. Whereas, in macro-averaging, the performance measures are computed for each individual class locally and then the average over all classes is taken. Micro-averaging gives equal weight to each document, whereas, macro-averaging gives equal weight to each class. Due to the high subject sparsity of our test corpus, there is a substantial number of classes which contain only one or two documents, and that could result in biased performance measures if macro-averaging is used. Therefore, in order to obtain a true objective evaluation of the classification performance of our system, we adopted the micro-average measure which gives equal weight to each document regardless of its class. The overall micro-averaged precision, recall, and F1 for all the one thousand documents in the test corpus regardless of their number of references are 0.84, 0.78, and 0.81 respectively. In order to show the effect of the number of references in the documents on the classification performance of the system, Table 3 shows the micro-averaged performance measures for each of the five document subsets in the test corpus individually.

Table 3. Performance measures for each of the five document subsets in the test corpus

Subset	# of References	# of Docs	Micro-Avg. Precision	Micro-Avg. Recall	Micro-Avg. F1
1	0	200	0.72	0.52	0.61
2	4	200	0.84	0.82	0.83
3	8	200	0.84	0.83	0.84
4	16	200	0.88	0.86	0.87
5	32	200	0.89	0.88	0.89
Overall	0-32	1000	0.84	0.78	0.81

[3] http://www.csn.ul.ie/~arash/BB-ATC1/HTML/index.html

As a common practice in developing a new ATC method or system, it is always desired to compare its performance with that of others. However, it was not possible for us to conduct a true objective comparison between the performance of our system and that of other reported ATC systems due to the following:

1. To the best of our knowledge there has been no previous attempt to automatically classify a collection of digital scientific literature according to a standard library classification scheme.
2. Unlike our system which utilizes the full DDC scheme, other relatively similar reported ATC systems, due to their limitations, either adopt only one of the main classes in the DDC/LCC along with its subclasses as their classification scheme, or use an abridged version of the DDC/LCC by limiting the depth of the classification hierarchy to second or third level.
3. Some of the similar works had reported the performance of their system using measures other than the standard performance measures of precision, recall, and F1 used in this work.

Despite above, it is possible to provide a relative comparison between the performance of our system and those of similar systems reported in the literature. For example, Pong and co-workers [14] used both NB and *k*-NN algorithms to classify 254 documents according to a refined version of the LCC scheme which consisted of only 67 categories. They reported the values of 0.802, 0.825, and 0.781 as the best figures for micro-averaged F1, recall, and precision, respectively, achieved by their system. Also, Chung and Noh [13] reported the development of a specialized economics web directory by classifying a collection of webpages, belonging to the field of economics, into 575 subclasses of the DDC main class of economics. Their unsupervised string-matching based classifier achieved an average precision of 0.77 and their supervised ML-based classifier achieved an average precision and recall of 0.963 and 0.901, respectively. in [17] we used an early version of the BB-ATC method to automatically classify a collection of 200 computer science related syllabus documents archived in the Irish national syllabus repository according to the full DDC scheme. The achieved micro-averaged performance measures of precision, recall, and F1 were 0.917, 0.889, and 0.902, respectively.

6 Conclusions and Future Work

In this paper we proposed a new category of ATC systems based on a new route for leveraging conventional library classification systems and resources, which we refer to as the Bibliography Based ATC (BB-ATC) approach. Our proposed approach solely relies on the subject classification metadata of the publications citing either the document to be classified or one of its references, as catalogued in the OPACs of conventional libraries, in order to probabilistically infer the most appropriate class for the document. In order to demonstrate the application and evaluate the classification performance of the proposed BB-ATC approach, we developed a prototype ATC system for automatic classification of scientific literature archived in the CiteSeer digital library. The developed ATC system was evaluated using a test corpus of one thousand scientific documents and the classification results were presented and analysed with the aim of quantifying the prediction performance of the system and identifying the factors influencing its performance. We reported micro-averaged values of 0.84, 0.78, and 0.81 for the overall precision, recall, and F1 measures of our system, respectively, and provided a relative comparison between the performance of our system and those of similar reported systems.

Based on above, we believe that we have developed a new robust approach for automatic classification of scientific literature in digital libraries and repositories according to a standard library classification scheme, which offers a prediction performance competitive to that achieved by the ML-based and string matching-based system. As for future work, we have identified a number of enhancements that could potentially improve the prediction performance of our method:

- As discussed in section 4, in the first stage of the data mining process carried out by the data mining unit of our system, the GBS engine is used to gather the corresponding metadata of all the publications that either cite the document to be classified or one of its references. GBS enables the full-text search of books, journals, and other materials that Google and its library and publisher partners scan, OCR, and index. In October 2009, Google announced that they had over 10 million items searchable through GBS [27]. Google does not provide public access to the full content of the majority of these items due to copyright restrictions. However, the metadata record of each archived item includes a so called “word cloud” which contains a set of key terms that have been identified as statistically significant within the full textual content of the item. The majority of these key terms are domain-specific, semantically rich, and directly related to the core subject of the book, and we have already proved their application in automatic keyphrase extraction from scientific documents [18]. These key terms could be used to measure the relevance of a publication, which cites either the document to be classified or one of its references, to the document. Thus, we are currently working on an enhanced version of the BB-ATC system which searches the content of the document to be classified for these key terms and based on their total number and frequency in the document derives a relevance weight, which measures the subject similarity of the citing publications to the document. Incorporating this new weight into the inference process should eliminate or at least limit the negative effect of a minor number of citing publications, whose main subject does not match the subject of the document to be classified.
- As discussed in section 5, the number of references in the documents to be classified has a large impact on the prediction performance of the proposed method. The references are used as indicative clues which collectively point to the right class for the document and, therefore, the larger the number of the clues the more reliable and accurate the classification results. Based on this, we can expect our method to yield its best performance when applied to documents which have a large number of references, such as Electronic Thesis and Dissertations (ETDs). Therefore, the next version of the BB-ATC system incorporating the enhancements described above will be deployed and evaluated for classification of a large collection of ETD documents archived in a digital library, such as the Networked Digital Library of Thesis and Dissertations (NDLTD) [28].

References

[1] Avancini, H., Rauber, A., Sebastiani, F.: Organizing digital libraries by automated text categorization. In: Heery, R., Lyon, L. (eds.) ECDL 2004. LNCS, vol. 3232. Springer, Heidelberg (2004)

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

[3] Golub, K.: Automated subject classification of textual Web pages, based on a controlled vocabulary: Challenges and recommendations. New Review of Hypermedia and Multimedia 12(1), 11–27 (2006)

[4] Yi, K.: Automated text classification using library classification schemes: trends, issues, and challenges. International Cataloguing and Bibliographic Control (ICBC) 36(4), 78–82 (2007)

[5] Markey, K.: Forty years of classification online: final chapter or future unlimited? Cataloging & Classification Quarterly 42(3), 1–63 (2006)

[6] Dewey, M.: Dewey Decimal Classification (DDC). (Online Computer Library Center (OCLC), 1876-2010) (cited February 2011), `http://www.oclc.org/us/en/dewey`

[7] Putnam, H.: Library of Congress Classification (LCC). (Library of Congress, Cataloging Policy and Support Office, 1897-2010) (cited February 2011), `http://www.loc.gov/catdir/cpso/lcc.html`

[8] Scorpion (OCLC Online Computer Library Center, Inc., 2002) (cited (February 2011), `http://www.oclc.org/research/software/scorpion/`

[9] OCLC (Online Computer Library Center (cited February 2011), `http://www.oclc.org/`

[10] Larson, R.R.: Experiments in automatic Library of Congress Classification. Journal of the American Society for Information Science 43(2), 130–148 (1992)

[11] Jenkins, C., Jackson, M., Burden, P., Wallis, J.: Automatic classification of Web resources using Java and Dewey Decimal Classification. Computer Networks and ISDN Systems 30(1-7), 646–648 (1998)

[12] Dolin, R., Agrawal, D., Abbadi, E.E.: Scalable collection summarization and selection. In: Proceedings of the Fourth ACM Conference on Digital Libraries. ACM, Berkeley (1999)

[13] Chung, Y.-M., Noh, Y.-H.: Developing a specialized directory system by automatically classifying Web documents. Journal of Information Science 29(2), 117–126 (2003)

[14] Pong, J.Y.-H., Kwok, R.C.-W., Lau, R.Y.-K., Hao, J.-X., Wong, P.C.-C.: A comparative study of two automatic document classification methods in a library setting. Journal of Information Science 34(2), 213–230 (2008)

[15] Frank, E., Paynter, G.W.: Predicting Library of Congress classifications from Library of Congress subject headings. Journal of the American Society for Information Science and Technology 55(3), 214–227 (2004)

[16] Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., Eliassi-Rad, T.: Collective classification in network data. AI Magazine 29(3) (2008)

[17] Joorabchi, A., Mahdi, A.E.: Leveraging the legacy of conventional libraries for organizing digital libraries. In: Agosti, M., Borbinha, J., Kapidakis, S., Papatheodorou, C., Tsakonas, G. (eds.) ECDL 2009. LNCS, vol. 5714, pp. 3–14. Springer, Heidelberg (2009)

[18] Mahdi, A.E., Joorabchi, A.: A Citation-based approach to automatic topical indexing of scientific literature. Journal of Information Science 36(6), 798–811 (2010)

[19] Giles, C.L., Kurt, D.B., Steve, L.: CiteSeer: an automatic citation indexing system. In: Proceedings of the Third ACM Conference on Digital Libraries. ACM, Pittsburgh (1998)

[20] Meier, W.: eXist-DB. (exist-db.org, Released under the open source GPL licence, 2009) (cited February 2011), `http://exist.sourceforge.net/`

[21] Google Books Search (GBS) engine. (Google, 2004) (cited February 2011), `http://books.google.com/`

[22] WorldCat (Online Computer Library Center (OCLC), 2001-2010 2008) (cited (February 2011), `http://www.oclc.org/worldcat/default.htm`

[23] WorldCat Search API (OCLC - WorldCat, 2009) (cited (February 2011), `http://worldcat.org/devnet/wiki/SearchAPIDetails`

[24] MARC standards (Library of Congress Network Development and MARC Standards Office, 1999 December 5, 2007) (cited February 2011), `http://www.loc.gov/marc/`

[25] Councill, I.G., Giles, C.L., Kan, M.Y.: ParsCit: An open-source CRF reference string parsing package. In: Proceedings of the Language Resources and Evaluation Conference (LREC 2008), Marrakesh, Morrocco (May 2008)

[26] O'Madadhain, J., Fisher, D., Nelson, T., White, S., Boey, Y.-B.: JUNG 2.0. (Released under the open source GPL licence, 2009) (cited February 2011), `http://jung.sourceforge.net/index.html`

[27] Brin, S.: A Library to Last Forever (The New York Times, October 8, 2009) (cited June 2010), `http://www.nytimes.com/2009/10/09/opinion/09brin.html?_r=1`

[28] Networked Digital Library of Thesis and Dissertations (NDLTD, 1996-2010) (cited February 2011), `http://www.ndltd.org`