A Supervised Machine Learning Classification Algorithm for Research Articles

Leonidas Akritidis
Dpt of Computer & Communication Engineering
University of Thessaly
37 Glavani, Volos, Greece
leoakr@inf.uth.gr

Panayiotis Bozanis
Dept of Computer & Communication Engineering
University of Thessaly
37 Glavani, Volos, Greece
pbozanis@inf.uth.gr

ABSTRACT

The issue of the automatic classification of research articles into one or more fields of science is of primary importance for scientific databases and digital libraries. A sophisticated classification strategy renders searching more effective and assists the users in locating similar relevant items. Although the most publishing services require from the authors to categorize their articles themselves, there are still cases where older documents remain unclassified, or the taxonomy changes over time. In this work we attempt to address this interesting problem by introducing a machine learning algorithm which combines several parameters and meta-data of a research article. In particular, our model exploits the training set to correlate keywords, authors, co-authorship, and publishing journals to a number of labels of the taxonomy. In the sequel, it applies this information to classify the rest of the documents. The experiments we have conducted with a large dataset comprised of about 1,5 million articles, demonstrate that in this specific application, our model outperforms the AdaBoost.MH and SVM methods.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining

Keywords

classification, supervised, machine learning

1. INTRODUCTION

The digital libraries and academic search engines have always been a precious tool for the researchers. Their main functionality is focused on providing search capabilities and further information regarding scientific articles, citations, journals and authors. Multiple such services are in operation on the Web; examples include the ACM Digital Library[1], Google Scholar[2], Microsoft Academic[3], and others.

[1]http://dl.acm.org/
[2]http://scholar.google.com/
[3]http://academic.research.microsoft.com/

SAC'13 March 18-22, 2013, Coimbra, Portugal.

The problem of the automatic classification of research articles is of remarkable importance for these services, since it enables increased functionality and improved performance. For instance, a robust classification strategy allows the user to perform searches by focusing on only a specific portion of the indexed documents, thus increasing both effectiveness and efficiency. Additional potential benefits include similar documents recommendations, collaborative filtering, query expansion facilities, expert identification, and so on.

Several methods have been proposed to address the issue of identifying the research field a scientific article belongs to. These methods include keyword extraction algorithms which attempt to identify repeated textual patterns and extract the most representative keywords from the article. In the sequel, they employ traditional classification approaches such as *k-nearest neighborhood (k-NN)* to identify the research field that best describes the content of the article. Another family of methods adopt citation analysis algorithms which study several citation properties, such as the phenomenon of two or more papers being cited together by multiple articles. These methods have two significant drawbacks: initially, it is not always possible to construct a complete graph of interlinking papers because some nodes and edges are simply not available. At second, a reference to an article does not necessarily reveal thematic affinity.

In this paper we attempt to address these issues by proposing a new algorithm for classifying research papers. More specifically, we introduce a model which has its origins in the traditional k-NN approach however, it also takes into consideration several aspects regarding the particular problem which we examine. These aspects include the authors history, co-authorship information, selection of keywords, and the previous publications of a journal. Our classifier is experimentally compared against two state-of-the-art generic text classification methods, namely support vector machines and AdaBoost.MH. We show that the inclusion of the aforementioned parameters leads to improved classification performance by roughly 6%.

The rest of the paper is organized as follows: In section 2 we refer to some of the most popular works studying the problem of classifying research articles. In section 3 we provide some preliminary information by describing the basic problem components and we describe in details the proposed algorithm. Section 4 contains the experimental evaluation of the algorithm, whereas section 5 concludes this article.

2. RELATED WORK

Document classification is a well-established data mining

problem and the issue of scientific papers classification is a specialization of this problem posing its own challenges.

The methodologies encountered in the literature can be divided into two wide categories: link-based and text-based categorization. The first category includes works which are mainly based on the document linking and the information extracted out them. For instance, in [4] the authors introduce a statistical framework for modeling link distributions and based on that knowledge, they classify a document according to the category its links belong to. Link-based classification is particularly essential for categorizing graph nodes (i.e. labeling the nodes of a graph, [10], or networked data classification[12]). Furthermore, similar approaches can be also applied on the Web, where the document interlinking can be used for a variety of purposes. An important survey of such methods is provided in [9].

On the other hand, machine learning (ML) text categorization has a gained substantial attention by the data miners; A complete survey on the most effective ML text categorization approaches is provided in [5]. Moreover, [8] and [6] provide detailed evaluations of the primary statistical and machine learning approaches to text categorization. Furthermore, Joachims employed support vector machines (SVM) for the same task [7], whereas [15] introduced AdaBoost.MH, a multi-class expansion of the traditional two-class AdaBoost algorithm.

Nevertheless, none of the aforementioned approaches take into consideration problem-specific information such as the authors of an article, co-authorship and the publishing journal. In this paper we introduce a new method which is based not only on the paper's keywords, but also in the previous activity of both the contributing authors and the publishing journal.

3. RESEARCH PAPER CLASSIFICATION

In this section we present our algorithm for classifying research articles. Initially we provide some background knowledge regarding the problem itself, and in the sequel we describe our solution along with some implementation details.

3.1 Preliminaries

Our analysis begins by introducing a number of fundamental sets that will assist us in establishing a baseline for our algorithm. Initially we define P as the set containing all publications, and J as another set including the journals[4] where the items of P have been published. Note that since each paper is published in exactly one journal, each entry $p \in P$ is mapped to a single element $j \in J$. Moreover, we define A as the set including all the involved authors, and C which consists of the research fields (also mentioned as categories, or labels) where the papers of P belong to. In other words, C represents the given taxonomy structure.

A large number of publications contain *keywords* which are used by the authors to explicitly represent the content of their work. The algorithm we present here exploits these keywords and for this reason, we define a set K which contains all the keywords encountered in all papers of P. In the same set K we also include the keywords extracted from the titles of the articles, since these words represent the documents contents as well.

[4]In this work we use the word journal to collectively refer to journals, magazines, books, and conference proceedings.

Symbol	Meaning
P	The set containing all papers
A	The set containing all authors
C	The set containing all research areas (taxonomy)
J	The set containing all journals
K	The set containing all keywords
A^p	The authors who created p
C^p	The research areas that p belongs to
K^p	The keywords included in p
P^c	The papers belonging to c
P^a	The papers authored by a
$P^{a,c}$	The papers authored by a and belong to c
P^j	The papers published in j
$P^{j,c}$	The papers published in j and belong to c
P^k	The papers containing k
$P^{k,c}$	The papers containing k and belong to c
$\mathcal{T}$	The training set

Table 1: Summary

In addition, we introduce the subset $K^p \subset K$ containing all the keywords of a single article p, the subset $P^k \subset P$ which stores all the publications including the keyword k, and the subset $P^{k,c} \subset P^k$ which contains the publications which both include k, and are mapped to the field c. In Table 1 we summarize all the above notifications.

3.2 Papers Classification Algorithm

All supervised machine learning algorithms are based on a predefined set of labels C, and a training set $\mathcal{T}$ comprised of articles which have been assigned one or more labels from C. Here we present an algorithm used to train our model by using C and $\mathcal{T}$, and in the sequel, we show how to apply this model to classify the unlabeled articles.

3.2.1 Model Training

In this phase we process the training set $\mathcal{T}$ and we construct a classification model with respect to the taxonomy C. This procedure includes three stages where we correlate keywords, authors and journals to one or more labels from C. We also record several frequency values which will be used later by the classification algorithm to effectively determine the labels of the unclassified papers.

The majority of the research articles includes a set of keywords placed between the abstract and the first section. Moreover, the words occurring in the title are also considered representatives of the document's content and can also be used in our model. Now consider a paper $p \in P$ drawn from the training set $\mathcal{T}$ which includes the keywords K^p and is categorized into one or more research fields $C^p \subset C$. Our objective is to create correlations between each keyword of p and each of the research fields of C^p. Since a keyword k may appear in multiple papers belonging to different research areas, we adopt a strategy similar to the k-NN; that is, we examine how frequently this keyword has been mapped to each field c. This is achieved by the construction of (k, c) pairs which we store in a *relevance description* vector $\mathcal{K}$ [13]. We also compute two frequency values $|P^k|$ and $|P^{k,c}|$: The former represents the number of papers including the keyword k, whereas the latter signifies the number of papers which both include k and are mapped to the field c.

Algorithm 1 shows the steps required to train $\mathcal{K}$. For each paper p of the training set we initially identify all the research fields C^p and the keywords K^p. For each research field $c \in C^p$ we create a (k, c) pair and we search for it within

$\mathcal{K}$. If the search is not successful, we insert the pair and we set $|P^{k,c}| = 1$; otherwise we merely increase $|P^{k,c}|$ by one. This procedure leads to the vector $\mathcal{K}$, which contains all the keywords of the papers accompanied by a global frequency value $|P^k|$ and a list of the corresponding research fields.

The previous activity of the authors who contribute to a research paper can also provide an indication of the research field the paper discusses. Learning the areas of expertise of a scholar is important since it can be exploited to classify his/her unlabeled articles. However, a scientist usually conducts research in multiple areas of science. For instance, consider an author X who has published articles in the areas of *databases* and *data mining* and for the production of these articles has co-operated with Y and Z respectively. Intuitively, an article authored by both X and Z should be labeled as a *data mining* paper.

To capture these intuitions we create a vector $\mathcal{A}$ which for each author a accommodates a list of all the encountered co-authors a' (*AA list*). Each co-author entry is accompanied by an array with the research fields of the articles authored by both a and a'. Hence, in case a and a' are encountered again in an unlabeled article, we retrieve the research fields of their common articles from the aforementioned list. Furthermore, for each author a we also create one more list (*AP list*) which stores all the research fields of all the papers of a. This record will be used to classify articles authored by multiple authors, but no previous co-authorship information between a and the co-authors is available.

Similarly to the previous stage, each research field is accompanied by two frequency values $|P^a|$ and $|P^{a,c}|$. The former represents the number of publications of a, whereas the latter denotes the total number of publications of a which are mapped to the research area c. The steps 14–30 of Algorithm 1 describe the construction process of $\mathcal{A}$. For each author a, the correlations (a, c) are all inserted in the AP list (steps 18–23). In the sequel, we iterate through each co-author an we create (a, a', c) tuples which correlate the author, his/her co-authors, and each field (steps 24–30).

Finally, the publishing journal can provide an indication about the research area that a paper belongs to. This is due to the fact that journals are also categorized and do not publish articles which deal subjects that are foreign to their interest area. For instance, a journal which is focused on *Data Engineering* would not publish a paper which discusses a problem related to *Systems Security*.

The aforementioned notifications lead to the enhancement of our trainer with its third part (Algorithm 1, steps 31–39). Here we identify the research areas that the majority of the papers published in a journal j belong to. Compared to the other two stages this one is slightly simplified because a paper is only published in one journal and hence, we do not have to iterate through multiple journals. The outcome of this process is the $\mathcal{J}$ vector, which contains for each journal, a list of research fields accompanied by their corresponding frequency values $|P^{j,c}|$. These values indicate how many times a journal j has published papers belonging to c.

Algorithm 1 Model training

```
1.  initialize K, A, J
2.  for each paper p ∈ T
3.      C^p ← ExtractResearchAreas(p)
        Phase 1: Processing of the keywords
4.      K^p ← ExtractKeywords(p)
5.      for each keyword k ∈ K^p
6.          |P^k| ← |P^k| + 1
7.          for each research area c ∈ C^p
8.              Create pair (k, c)
9.              if K.search(k, c) =false
10.                 K.insert(k, c)
11.                 |P^{k,c}| ← 1
12.             else
13.                 |P^{k,c}| ← |P^{k,c}| + 1
        Phase 2: Processing of the authors
14.     A^p ← ExtractAuthors(p)
15.     for each author a ∈ A^p
16.         |P^a| ← |P^a| + 1
17.         for each research area c ∈ C^p
18.             Create pair (a, c)
19.             if A.AP.search(a, c) =false
20.                 A.AP.insert(a, c)
21.                 |P^{a,c}_{AP}| ← 1
22.             else
23.                 |P^{a,c}_{AP}| ← |P^{a,c}_{AP}| + 1
24.             for each author a' ∈ A^p
25.                 Create tuple (a, a', c)
26.                 if A.AA.search(a, a', c) =false
27.                     A.AA.insert(a, a', c)
28.                     |P^{a,c}_{AA}| ← 1
29.                 else
30.                     |P^{a,c}_{AA}| ← |P^{a,c}| + 1
        Phase 3: Processing of the journals
31.     j ← ExtractJournal(p)
32.     |P^j| ← |P^j| + 1
33.     for each research area c ∈ C^p
34.         Create pair (j, c)
35.         if J.search(j, c) =false
36.             J.insert(j, c)
37.             |P^{j,c}| ← 1
38.         else
39.             |P^{j,c}| ← |P^{j,c}| + 1
```

3.2.2 Articles Classification

We can now employ the trained model to classify an unlabeled article $p \in P$. Similarly to the training phase, classification is also conducted in three phases. During each phase, the involved research fields are assigned scores according to their correlation with the keywords, authors, and journal of p. In Algorithm 2 we describe these procedures.

In the first phase (steps 2–6), we initially extract the paper's keywords K^p and for each retrieved keyword k we perform a search in the relevance description vector $\mathcal{K}$. In case this search is successful, we retrieve the list of the associated research areas along with the respective $|P^{k,c}|$ values. Then, for each research field c we compute a score $\mathcal{S}^c_k$ according to a scoring function $\mathcal{S}^c_k = F_k(P^k, P^{k,c})$. This function can implement simple schemes such as the traditional *idf* (i.e. $|P^{k,c}|/|P^k|$), or more complex ones. The steps 2–6 of Algorithm 2 illustrate the exact process.

Classification is further enhanced by taking into account the information regarding the authors of p. However, in this case the process is more complex, since we must consider the co-authorship data and retrieve the correct record from the vector $\mathcal{A}$. Initially, we identify the set of authors A^p of p. In the sequel, we search among the AA co-authorship records and in case a correlation between two authors is found, each corresponding research field is assigned a score $\mathcal{S}^c_a = F_a(P^a_{AA}, P^{a,c}_{AA})$ (steps 10–14). If such a correlation is not present in $\mathcal{A}$, we search in the AP list and we use the associated AP scores (steps 15–18).

Moreover, our classifier takes into consideration the re-

Algorithm 2 Paper Classification

```
    Classify(p_i, F, K, A, J)
1.  for each unlabeled article p
    Phase 1: Keyword-based classification
2.      K^p ← ExtractKeywords(p)
3.      for each keyword k ∈ K^p
4.          if k ∈ K
5.              for each (k, c) ∈ K
6.                  S_k^c ← F_k(P^k, P^{k,c})
    Phase 2: Author-based classification
7.      A^p ← ExtractAuthors(p)
8.      for each author a ∈ A^p
9.          coauthor ← false
10.         for each author a' ∈ A^p
11.             if (a, a') ∈ A.AA
12.                 coauthor ← true
13.                 for each (a, a', c) ∈ A.AA
14.                     S_a^c ← F_a(P_AA^a, P_AA^{a,c})
15.         if coauthor = false
16.             if a ∈ A.AP
17.                 for each (a, c) ∈ A.AP
18.                     S_a^c ← F_a(P_AP^a, P_AP^{a,c})
    Phase 3: Journal-based classification
19.     j ← ExtractJournal(p)
20.     if j ∈ J
21.         for each (j, c) ∈ J
22.             S_j^c ← F_j(P^j, P^{j,c})
```

search fields of specialization of a journal. Hence, in case the publishing journal j of an unlabeled paper p has been encountered during the training phase, we can exploit the correlated research fields (stored within the vector $\mathcal{J}$) to classify p. The third part of Algorithm 2 describes our approach. After the identification of the journal j of p, we perform a look-up in $\mathcal{J}$. In case searching is successful, we retrieve the research fields that the published articles of j belong to, along with their respective frequency values. For each of these fields we calculate a score given by a third function $\mathcal{S}_j^c = F_j(P^j, P^{j,c})$.

The total score assigned to a research field c is a linear combination of the three aforementioned scores:

$$\mathcal{S}^c = w_k \mathcal{S}_k^c + w_a \mathcal{S}_a^c + w_j \mathcal{S}_j^c \quad (1)$$

where w_k, w_a, and w_j are constant parameters used to regulate the contribution of the keywords, the authors, and the publishing journal of an article. These three constants are tuned with the aim of satisfying the following limitation:

$$w_k + w_a + w_j = 1 \quad (2)$$

To identify the research field a paper belongs to, we merely have to sort the $\mathcal{S}^c$ scores in descending order and select the first entry (the one received the highest score, $max(\mathcal{S})$). In this way, each article is mapped to only one research field. We can raise this limitation and classify an article into multiple research areas, by introducing a coefficient $\epsilon \in (0, 1]$. Then, each paper is assigned additional research areas if the scores of these areas satisfy the following condition:

$$\mathcal{S}^c \geq \epsilon \; max(\mathcal{S}) \quad (3)$$

The value of ϵ determines how strict this condition can become. For $\epsilon = 1$ we tolerate no fields with scores lower than the maximum and an article is mapped to the research area (or areas) which received the highest score.

4. EXPERIMENTAL EVALUATION

In this section we measure the effectiveness of our proposed algorithm. Initially we describe the employed dataset and the taxonomy structure, and in the sequel we present the results of our performance measurements.

4.1 Dataset and Taxonomy Characteristics

To experimentally attest the effectiveness of our classification approach it is required that we firstly determine the set of labels which will be used by the classifier (i.e. the taxonomy), and a dataset comprised of an adequate number of articles. In addition, a subset of these articles must support the selected taxonomy, that is, the items of this subset must be mapped to at least one of the labels of the taxonomy.

The strict data protection policies applied by the digital libraries renders the collection of bibliometric data a rather challenging task. Since the crawling of such repositories is forbidden, we are limited in using only open access document collections. The largest among these collections is the CiteSeerX [1] dataset, an open repository comprised of approximately 1.8 million scientific articles. These articles are related to the wider fields of computer and communications engineering, and a significant portion of them are mapped to the local taxonomies employed by their publishers. Of course, each publisher applies its own taxonomy structure; consequently, the first issue we need to address is to determine a unique taxonomy which will be used to classify the rest of the articles.

Since our primary goal is to build a training set comprised of largest possible number of articles, we simply scanned our dataset to identify the organization which published the most documents. Our analysis proved that the 63% of the articles of the CiteSeerX dataset has been published by either ACM or IEEE. These publishers employ a common categorization policy; they classify their published articles into a taxonomy[5] of research fields which mainly includes areas and sub-areas from the Computer Science, Engineering, Communications, and Mathematics. The structure consists of three levels of categorization, i.e. 11 first-level research fields divided into 81 second-level and 276 third-level classes. To achieve extensive and unbiased measurements of the effectiveness of our algorithm, we employed each level as a different taxonomy and we gave the names $C11$, $C81$ and $C276$.

After the selection of the taxonomy, the training set is immediately identified. All 1,159,634 articles which are mapped to one ore more labels of the ACM/IEEE taxonomy, are automatically becoming members of the training set. Our goal now is to assign labels to the rest 684,638 articles.

4.2 Model Training

The model training process includes two separate phases: initially, we construct the relevance description vectors $\mathcal{K}$, $\mathcal{A}$, and $\mathcal{J}$ and in the sequel, we attempt to evaluate the w_k, w_a, and w_j parameters of equation 1 which maximize the performance of our classifier.

For this reason, we applied a cross validation strategy according to which the training set was split in three equally-sized parts. The first two thirds were used for building $\mathcal{K}$, $\mathcal{A}$, and $\mathcal{J}$. In the sequel, we used the last third of the training set to measure the classification performance for all the pos-

[5]http://www.acm.org/about/class/ccs98-html

$\|\mathcal{T}\|$	C	$\{w_k, w_a, w_j\}$	**Acc.**	**SVM**	**Ada**
10,000	$C11$	{0.3, 0.1, 0.6}	94.0%	88.2%	88.8%
	$C81$	{0.2, 0.1, 0.7}	87.5%	82.9%	83.4%
	$C276$	{0.2, 0.1, 0.7}	80.7%	78.4%	80.1%
100,000	$C11$	{0.3, 0.2, 0.5}	95.1%	89.6%	-
	$C81$	{0.3, 0.1, 0.6}	88.2%	84.3%	-
	$C276$	{0.2, 0.2, 0.6}	80.9%	79.0%	-
1,159,634	$C11$	{0.3, 0.2, 0.5}	95.9%	94.1%	-
	$C81$	{0.3, 0.2, 0.5}	89.0%	87.9%	-
	$C276$	{0.3, 0.1, 0.6}	81.3%	80.8%	-

Table 2: Optimal tuning of the w_k, w_a, and w_j parameters for the three employed taxonomy structures and for training sets of different sizes.

sible combinations of w_k, w_a, and w_j. In particular, we continuously modified the values of all three parameters in the range $[0.0, 1.0]$ (with respect to equation 2) and we recorded the number of papers for which our classifier assigned correct labels. To verify the correctness of our results and to eliminate any random effects, we experimented with different training set sizes. More specifically, we repeated our measurements by using training sets comprised of 10,000, 100,000, and all the 1.16 million articles.

In Table 2 we report the results of this experiment. The first column denotes the training set sizes, whereas in the second column we show which taxonomy structure is employed. In the third column we record the values of the w_k, w_a, and w_j for which our classifier achieved maximum performance, whereas in the last three columns we report the accuracy of our algorithm against two methods based on support vector machines (SVM) and AdaBoost.MH (Ada).

Notice that our proposed classifier exhibited equally high effectiveness for multiple combinations of w_k, w_a, and w_j. From Table 2 we conclude that among the three exploited types of data (i.e. keywords, authors, and journal), the publishing journal is the most important indication of the research field an article belongs to. On the other hand, the previous works (i.e. the history) of the authors is the weakest one. A second conclusion is that our algorithm is relatively insensitive to the training set size; its performance degrades by only a small percentage (i.e. 1-2%) when the training set size is decreased by a factor of 10. In all cases, a satisfactory setting for the three constants is $w_k \in [0.2, 0.3], w_a \in [0.1, 0.2]$ and $w_j \in [0.5, 0.6]$.

In addition, we observe that performance decreases as the number of the available labels increases. This is expected since as the size of the taxonomy increases, the possibility of an erroneous prediction also increases (the classifier has more available choices). However, the possibility is not proportional to the taxonomy size; Although $C81$ includes about 8 times more labels than $C11$, the effectiveness degrades by only a percentage of 9-10%. Finally, we point out the remarkable 94% of successful labeling in the case of our small $C11$ taxonomy.

Now let us compare our approach against the state-of-the-art method based on SVMs. Since the available label sets consist of multiple entries, it is required that we use multi-class SVMs. In particular, we use the *one-against-all* strategy which given a set of y labels, it requires the construction of one binary classifier per label. To decide which labels to assign to each article, we take the classes that present the largest margins. The features we selected for SVM training were identical to the ones of our proposed algorithm, i.e. keywords, authors and journals. The one-against-all strategy is far preferable than the other existing approach, *one-against-one*, which performs pairwise classifications and requires the construction of y^2 classifiers.

Vector	**Records**	**Most Frequent**	**Articles**
$\mathcal{K}$	475,308	system	144,295
$\mathcal{A}$	497,604	Philip S. Yu	654
$\mathcal{J}$	3,915	Theor. Computer Science	13,295

Table 3: Trained Model Statistics

To construct each binary SVM classifier, we created an equal number of training sets. For instance, for the experiments with the $C11$ label set we created 11 training sets. Each training set comprised of all the papers mapped to this specific research field (positive examples), followed by the rest of the articles which were declared as negative examples. We then created one binary classifier for each label, by using the SVMLight Program [14]. Finally, we scanned the outputs of these classifiers and each article was assigned the label which presented the largest margin. The precision results are recorded in the last column of Table 2.

Our approach outperformed the SVM-based approach in all of the examined cases. The results of Table 2 reveal that the performance gap between the two methods increases as the training set size becomes smaller. More specifically, in case the training set consists of 10 thousand articles, the precision of our proposed method is higher than the precision of the SVMs by a margin ranging between 2% and 6%. On the other hand, the smallest performance gap was observed in the scenario where the employed label set is $C276$ and the training set consisted of all the 1.16 million articles.

We also compared our algorithm against another high-performing classification algorithm, AdaBoost.MH. For our small training set comprised of 10,000 articles, our approach achieved better performance by a percentage ranging between 1% and 6%. In the other two larger training sets, AdaBoost.MH consumed all the available memory (12GB) of our machine and the experiments failed to complete. On the contrary, the in-memory data structures of our approach occupied roughly 1.1 GB for the largest training set.

In Table 3 we present some interesting statistics of our model. The keywords relevance description vector is comprised of about 475 thousand keywords, and the most frequent keyword is *system*. Furthermore, the most frequent author and journal were *Philip S. Yu* and *Theoretical Computer Science*, which authored and published 654 and 13,295 articles respectively.

4.3 Classification Results

Now we employ the trained model of the previous phase to classify the 684,638 unlabeled articles of our dataset. As previously, we conducted the same experiment three times and each time we employed one of the taxonomy structures $C11$, $C81$, and $C276$. In Tables 4 and 5 we illustrate the five most popular research fields for each of our employed taxonomy structures for $\epsilon = 1$ and $\epsilon = 0.85$ respectively.

In the case of $\epsilon = 1$ the unlabeled articles are assigned only one research field (the one which received the highest score according to equation 1). Regarding the $C11$ taxonomy, the most popular research field was *Computing Methodologies*; the 40.6% of the articles were classified to this category. Furthermore, *Artificial Intelligence* and *General* were the

	Label	Articles
1	Comp. Methodologies	278,179
2	Inform. Systems	99,958
3	Systems Organization	71,635
4	Math. of Computing	69,140
5	Software	66,391

	Label	Articles
1	Artificial Intelligence	174,272
2	Comp.-Comm. Networks	83,410
3	Numerical Analysis	62,211
4	Software Engineering	47,962
5	Interfaces & Presentation	29,617

	Label	Articles
1	General	185,718
2	Net. Archit.-Design	49,423
3	Nonnum. Algorithms	43,153
4	Applications	20,161
5	Types-Design Styles	19,218

Table 4: Classification results for the three experimental taxonomy structures, for $\epsilon = 1$: (i) Left: $C11$, (ii) Center: $C81$, and (iii) Right: $C276$

	Label	Articles
1	Comp. Methodologies	347,739
2	Inform. Systems	144,668
3	Math. of Computing	97,514
4	Systems Organization	92,136
5	Software	91,331

	Label	Articles
1	Artificial Intelligence	224,183
2	Comp.-Comm. Networks	101,391
3	Numerical Analysis	85,764
4	Software Engineering	65,010
5	Interfaces & Presentation	41,545

	Label	Articles
1	General	237,761
2	Nonnum. Algorithms	64,204
3	Net. Archit.-Design	60,994
4	Applications	31,826
5	Design Methodology	29,369

Table 5: Classification results for the three experimental taxonomy structures for $\epsilon = 0.85$: (i) Left: $C11$, (ii) Center: $C81$, and (iii) Right: $C276$

most popular research areas of the $C81$ and $C276$ taxonomy structures respectively. In particular, the former label was assigned to the 25.4% of the unlabeled articles, whereas the latter gathered the 27.1% of the articles.

On the other hand, in the case of $\epsilon = 0.85$ the scientific documents can be assigned more than one labels, if the scores of the additional labels obey to the limitation of equation 2. Therefore, it is anticipated that each research field is assigned to more articles than in the previous case. For this reason, the orderings of Table 5 are slightly modified compared to the ones of Table 4.

5. CONCLUSIONS

In this paper we introduced a supervised machine learning algorithm for classifying research articles. The problem we examine is particularly useful for academic search engines and digital libraries, since a robust solution can provide improved functionality and performance benefits. Our algorithm operates by using a predefined list of labels (i.e. taxonomy structure) and a training set comprised of labeled articles. The training process we proposed is based on the creation of three vectors which correlate each article keyword, author, and journal with a number of labels of our given taxonomy. During the classification process of the unlabeled articles, we use the data stored within these vectors to compute scores for each label. Our experimental evaluation on a large set of 1.5 million research articles demonstrated that our approach achieved successful classification by a percentage above 90%.

6. REFERENCES

[1] CiteSeerX. *http://csxstatic.ist.psu.edu/about/data.*

[2] S. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. *Third Text REtrieval Conference, Gaithersburg, USA*, 1994.

[3] S. Robertson, H. Zaragoza, and M. Taylor. Simple BM25 Extension to Multiple Weighted Fields. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 42–49, 2004.

[4] Q. Lu, and L. Getoor Link-based Classification, *Advanced Methods for Knowledge Discovery from Complex Data, 2005*

[5] F. Sebastiani. Machine Learning in Automated Text Categorization *ACM computing surveys (CSUR), 2002* vol. 34, issue 1, pp. 1–47

[6] Y. Yang. An evaluation of statistical approaches to text categorization *Information Retrieval, 1999* vol. 1, issue 1, pp. 69–90

[7] T. Joachims. Text categorization with support vector machines: Learning with many relevant features, *Machine Learning: ECML 1998*, pp. 137–142

[8] Y. Yang and J.O. Pedersen. A comparative study on feature selection in text categorization, In *Proceedings of the International Conference on Machine Learning (ICML'97), 1997*, pp. 412–420.

[9] X. Qi and B.D. Davidson. Web page classification: Features and algorithms, *ACM Computing Surveys (CSUR), 2009*, vol. 41, issue 2, pp. 1–31

[10] M. Bilgic, and G.M. Namata, and L. Getoor. Combining collective classification and link prediction, In *Proceedings of Workshop on Mining Graphs and Complex Structures at the IEEE International Conference on Data Mining, 2007*, pp. 381–386

[11] L. Getoor and C.P. Diehl. Link mining: a survey, *ACM SIGKDD Explorations Newsletter, 2005*, vol. 7, issue 2, pp. 3–12

[12] S.A. Macskassy and F. Provost. Classification in networked data: A toolkit and a univariate case study, *The Journal of Machine Learning Research, 2007*, vol. 8, pp. 935–983

[13] N. Fuhr. A probabilistic model of dictionary-based automatic indexing. In *Proceedings of RIAO-85, 1st International Conference "Recherche d'Information Assistee par Ordinateur", 1985,* pp. 207–216.

[14] T. Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the International Conference on Machine Learning (ICML'99), 1999*, pp. 200–209.

[15] Zhu, J. and Rosset, S. and Zou, H. and Hastie, T. Multi-class adaboost. In *Ann Arbor, 2006* vol. 1001, issue 1, pp. 1612–1631.