Journal of Universal Computer Science, vol. 25, no. 13 (2019), 1687-1716
submitted: 25/3/19, accepted: 14/8/19, appeared: 28/12/19 © J.UCS

Improving Multi-Label Classification for Learning Objects Categorization by Taking into Consideration Usage Information

Pedro G. Espejo
(University of Cordoba, Cordoba, Spain
pgonzalez@uco.es)

Eva Gibaja
(University of Cordoba, Cordoba, Spain
egibaja@uco.es)

Víctor H. Menéndez
(Autonomous University of Yucatan, Merida, Mexico
mdoming@correo.uady.mx)

Alfredo Zapata
(Autonomous University of Yucatan, Merida, Mexico
zgonzal@correo.uady.mx)

Cristóbal Romero
(University of Cordoba, Cordoba, Spain
cromero@uco.es)

Abstract: Learning objects are digital resources that can be deployed by means of a web system for supporting teaching. A key advantage is reuse, and this is possible thanks to learning objects repositories that allow learning object search, management and categorization. In this work, we propose a novel approach towards automatically learning object categorization taking into consideration learning object usage information. We use a multi-label learning approach since each learning object might be associated with multiple categories. We have developed a methodology with three main stages allowing us to firstly select the most suitable set of text features from learning objects metadata, secondly selecting how much historical learning object usage information can enhance classification performance, and finally selecting the best multi-label classification algorithms with our data. We have carried out an experimental work using 519 learning objects gathered from the AGORA repository for 8 years. We have compared 13 multi-label classification algorithms over 16 evaluation measures. The results obtained show that usage information about the learning object can improve the classification.

Keywords: multi-label classification, feature selection, usage information, learning object categorization
Categories: H.4.m, J.7

1 Introduction

A Learning Object (LO) is any digital entity with a specific didactic content and an educational goal which can be used, re-used or referenced on technology supported learning [IEEE 2016a]. With the increasing number of LOs and the availability of web-based LO repositories, the possibility of their fast and effective finding and retrieving has become a more critical issue [Zapata 2015]. Several metadata standards such as SCORM (Shareable Content Object Reference Mode), LOM-ES (Learning Object Metadata - Spanish), IEEE-LOM (Institute of Electrical and Electronics Engineers - Learning Object Metadata), etc. have been proposed for characterizing LOs and allowing powerful search [Rodés-Paragarino, 2016]. However, there are so many possible metadata fields to characterize a LO that it can be quite time consuming for users to provide all that information. As a consequence, often users omit most metadata when entering a new LO to a repository, hampering this way an efficient search later, preventing the reuse of LOs. For example, users sometimes don't provide information about what is the subject area, discipline or category that the LO belongs to.

In order to try to resolve this problem, in this paper, we concentrate on how to fill automatically some attributes of the IEEE LOM when a user adds a new LO to a repository, in our case the AGORA repository [Zapata 2013]. In fact, our final goal is to automatically recommend to the user the possible categories of a LO from just provided information about the LO (such as the title, keywords and description). To do it, we propose to use Multi-Label Classification (MLC) for the automatic categorization of LOs in subject areas (discipline field in IEEE LOM, that is, the subject area that the LO belongs to) from the terms or pure text features that characterize these LOs. MLC is a classification paradigm where multiple target labels can be assigned simultaneously to each instance [Gibaja 2014]. Since each LO might be associated with multiple labels or disciplines, a MLC approach is better suited than traditional classification techniques (single-label) where each LO can belong to just a unique class. In this work, we propose an approach for MLC of LOs in subject areas by using not only traditional LO textual metadata, but also social data about the usage of these LOs. We start using all the metadata provided by the users/authors when they add it to a LOs repository. But due to the high number of text features that can be generated from these metadata, we propose to reduce its number by ranking and selecting the best ones. Next, we propose to use usage information about LOs. Repositories incorporate new metadata related to the contributions of the users, the activities carried out and their execution date, which allows developing solutions that facilitate their location, its recovery and its use in e-learning solutions [Lytras 2007]. The relationships between users and LOs allow us to assume that users and resources in the same area of knowledge have a high interaction, some cases being quite significant. For example, a repository may recommend LOs that are similar to those that the user used in the past, considering, for this, other characteristics associated with the resource (use, owner profile, content quality, etc.). [Ochoa 2008] establish a set of metrics to calculate the similarity of LOs based on the activities associated with management. Yen et al. [Yen 2009] establish a collection of similarity metrics based on object downloads, combined with matching terms and the use of the object is weighted more. So, we propose to use information about users' interactions gathered over the years. The usage information of LOs can provide us information about what are the disciplines or areas of the final users of the LOs [Ochoa 2011]. In this way, for example if a LO is mainly used for mathematicians, then independently of its textual metadata or the initial area/category assigned by his/her creator, then this LO can be also associated to this area/category. The usage information available in AGORA for each LO comprises the number of users who have accessed, downloaded and evaluated the LO in the repository. In this way, we try to improve the traditional classification that only uses content information by including usage information. Finally, we propose to find the best MLC algorithm to be used as recommender system in our problem of AGORA LOs categorization. So that, our research tries to answer the following three research questions:

1. RQ1: Can we find the minimal number of LO text features that assure sufficiently good classification performance?
2. RQ2: Can we enhance classification performance of a LO by adding to the textual features historical usage information of the LO?
3. RQ3: Can we select an algorithm or group of algorithms as the best performers for categorization recommendation?

The rest of this paper is organized as follows: section 2 briefly reviews the background to our work; section 3 describes the proposed approach; section 4 describes the data set used in our research and the experimental work that we have carried out; and finally, section 5 provides some concluding remarks and suggests future lines of research.

2 Background

Next, the three main areas related to this work (learning objects, multi-label classification and multi-label classification of learning objects) are introduced.

2.1 Learning Objects

Several definitions and taxonomies have been proposed for LOs [McDonald 2006, Innis-Allen 2008]. We adopt the definition [IEEE 2016a] of LOs as any digital entity which can be used, re-used or referenced during technology supported learning. In this sense, any entity with a specific didactic content and an educational goal is a LO. One of the main benefits of LOs is their potential reusability. The concept of reusable LOs has evolved into a central component within the current context of e-learning [López 2012]. Often, LOs are developed and made available for anyone who wants to use them. But reusability hinges on two important concepts: metadata and repositories. Metadata allow to characterize a LO and hence, allow searching the LOs that are best suited for our educational purposes. A good metadata scheme allows powerful searches based on different criteria. Several metadata standards have been proposed:

- SCORM is a standard proposed by ADL (Advanced Distributed Learning) focused on LO sharing and reusability [ADL 2016].
- IMS-LD, proposed by IMS Global Consortium is a specification for a language aimed at describing learning processes, rather than LOs [IMS 2016].

- IEEE-LOM, proposed by the IEEE, is a standard with a label set allowing the description of LOs [IEEE 2016b]. IEEE-LOM can use different serialization formats for representating the metadata, being XML the most common.
- LOM-ES is the version of IEEE-LOM for the Spanish-speaking educational community. It includes several modifications over the standard IEEE-LOM, mainly in the form of new elements (for example 5.12 cognitive process and 6.4 access), as extensions to the predefined vocabularies (by example the vocabulary of 5.2 LearningResourceType). An important difference is in the nature of the elements defined as mandatory (such as title, description, coverage), recommended (such as keyword, contribution, location) or optional (such as requirements, notes, cost) [AENOR 2014].

In order to be able to perform LO searches, in addition to a language that allows characterizing the LOs (metadata) we need a technological component that allows the search on the real world, and this is the role that learning object repositories (LORs) play. A LOR is a software component that allows the rational storage of LOs (and their corresponding metadata) and their search. Obviously, since E-learning relies on the Internet, LORs are usually web-based systems, allowing to interact (definition and manipulation) with the repository through the web. Some well-known LORs are:

- MERLOT (Multimedia Educational Resource for Learning and Online Teaching), developed by the California State University Center for Distributed Learning (CSU-CDL), is a LO repository that storages only metadata, referencing LOs hosted on different external locations [MERLOT 2016].
- ARIADNE (Alliance of Remote Instructional Authoring & Distribution Networks for Europe), developed by the European Commission's Telematics for Education and Training Program, consists in a hierarchical network of nodes storing both LOs and metadata [ARIADNE 2016].
- MACE (Metadata for Architectural Contents in Europe) is a European initiative to integrate LO repositories distributed over several countries to disseminate digital information about architecture [Stefaner 2007].
- CAREO (Campus Alberta Repository of Educational Objects) is a Canadian LOR that has as its primary goal the creation of a searchable, web-based collection of multidisciplinary teaching materials for educators [Magee 2001].
- ColombiaAprende (ColombiaLearns) is a Colombian initiative for the creation of an institutional bank of objects that organizes and distributes the existing educational material in educational institutions, at the same time increasing the possibilities of its reuse and promoting direct and indirect collaboration [Leal Fonseca 2010].
- AGORA (from a Spanish acronym that means Help for the Management of Reusable Learning Objects) developed by the University of Castilla la Mancha (Spain) and University of Yucatan (Mexico) it is a LO repository that includes metadata and its associated resources [Zapata 2013]. This is the repository used in this work.

Modern repositories, rather than simple spaces to publish digital resources for education, are complete systems where users manage their Learning Objects allowing new models of interaction between them. Within this context the function of metadata extends beyond being simple static descriptors of resources: they become changing entities, recording everything that happens around the Learning Object over time. This makes them an important source of knowledge to propose values, resources and actions to users, streamlining and facilitating their activities and interactions. In this sense, our proposal of search and recommendation of learning objects is enriched by combining the categorization of resources along with the information on their use. The determination of values for a list of categories is a difficult task for the teacher and consequently is prone to failures [Sicilia 2005]. When this activity is carried out manually, significant errors originate when defining the values of these metadata: omission of values, capture errors, selection of incorrect values, misinterpretations or ignorance, among other problems of a subjective nature [Cechinel 2009]. All these errors and deficiencies cause the reduction of the quality of the stored information, since in many cases the metadata is absent, incomplete or poorly constructed [Paulsson 2006] [Currier 2003]. The quality of the metadata has a direct impact on all the management processes of the Learning Objects [Bruce 2004], including the search and recommendation of resources. If the categorization is correct, its interoperability and reuse will be possible to a greater extent.

In this work, we have used IEEE-LOM meta-data standard and AGORA LO repository. We have selected AGORA because we have available all the data from this Ibero-American learning object management system. And we have used IEEE-LOM because it is the standard followed by AGORA. IEEE-LOM defines a hierarchical structure consisting of nine categories (general, lifecycle, metadata, technical, educational, rights, relation, annotation, classification) that contains more than 60 metadata items in total. These items can store values for different elements and types. The LO metadata is stored using a structure that conforms an XML Scheme, allowing a formal description of metadata structure and facilitating the management, search and recovery of the described resources [Prieto Méndez 2014].

2.2 Multi-label Classification

Classification is one of the most studied tasks in machine learning and data mining [Han 2011, Tan 2018] and it consists in predicting the value of a categorical attribute (the class) based on the values of other attributes (predicting attributes). In a classification problem, we have a classification criterion with a fixed set of possible classes for each instance. Normally, classes are mutually exclusive, that is, a specific instance can belong to just a single class. For example, if we are classifying animals by sex, each one could be male or female, but not both. However, there are occasions where classes present overlapping, that is, a specific instance can belong to several classes. For example, if we are classifying pictures by subject, a picture could be classified as mountains and sea simultaneously if it records a landscape of a beach with mountains in the background. This particular type of classification setting is known as multi-label classification (MLC) [Gibaja 2014]. In our work, we are interested in MLC, because a specific LO could belong to several areas of interest, for example education and science, if we have a LO aimed at teaching educational techniques specific to scientific contents.

In the next subsection the three main approaches (problem transformation methods, algorithm adaptation methods and ensembles of multi-label classifiers) have been identified for tackling MLC [Gibaja 2014].

2.2.1 Problem Transformation Methods

Problem transformation methods are based on the idea of transforming the multi-label data set in into one or several single-label datasets in such a way that any single-label classifier could be used. Binary Relevance (BR) [Gibaja 2014] decomposes the multi-label problem into one independent binary problem per label and then a common classifier is generated for each dataset. This method is so popular due to its efficiency and simplicity, nevertheless one of its more criticized features is the fact of not considering label relationships.

Several methods have been proposed to overcome this problem. One of them is Classifier Chains (CC) [Read 2011] which generates a chain of binary datasets, one for each label, in such a way that the feature space of each dataset is extended with the label values of the previous datasets in the chain. Label Powerset (LP) [Boutell 2004] transforms the multi-label dataset into one multi-class dataset in such a way that each combination of labels in the original one is considered as a new class. Then a common classifier is generated. Its complexity is exponential with the number of labels and is not able to predict labelsets which do not appear in the original dataset. To overcome the complexity problem of LP, Pruned Sets (PS) [Read 2008] first performs a pruning process to focus on the most frequent labelsets and then applies LP. Calibrated Label Ranking (CLR) [Fürnkranz 2008] generates a binary dataset for each pair of labels. Therefore, each dataset contains as instances those instances belonging to one of the labels, but not both. Besides, for each label, one binary dataset is generated by considering that if the label is positive, then the pattern is also positive, othercase a virtual label is assigned. This virtual label acts as split point for relevant labels.

2.2.2 Algorithm Adaptation Methods

Algorithm adaptation methods adapt an existing single-label classification algorithm to cope with a multi-label classification problem without applying any previous transformation of the data.

Many classification algorithms have been adapted to the multi-label setting within the algorithm adaptation approach. Multi-Label k-Nearest Neighbour (ML-kNN) [Zhang 2007] is an adaptation of the well-known kNN algorithm. It counts the neighbours belonging to each label and, for an unseen instance, the maximum a posteriori (MAP) principle is used to determine the set of associated labels. Instance-based Logistic Regression (IBLR) [Cheng 2009] combines instance-based learning and logistic regression by using the labels of neighbours as extra attributes in a logistic regression scheme. Another worthy method is AdaBoost.MH [Schapire 2000] which is the adaptation of AdaBoost [Freund 1995], an algorithm which iteratively generates a set of weak classifiers in such a way that each classifier can focus in those instances more difficult to predict by the previous ones by weighting instances. Adaboost.MH works similarly but maintaining weights over both instances and labels. More recently, a deep-learning based method has been proposed [Fu 2018].

2.2.3 Ensembles of Multi-label Classifiers

Finally, a third category, so-called ensembles of multi-label classifiers can be cited. Note that despite the fact that some of the proposals cited above involve the combination of several single-label classifiers (e.g. BR and CLR), only those methods whose base classifiers are multi-label are considered into this group. A complete review about this issue can be found in [Moyano 2018].

As the order of the chain may affect the performance of the CC classifier, an Ensemble of Classifier Chains (ECC) [Read 2011] that follows a bagging scheme and uses a different chain for each base classifier has been proposed.

Multi-Label Stacking (MLS) [Tsoumakas 2009] is based on BR and it applies BR twice. It first trains a base-level consisting of an independent binary classifier for each label. Then MLS learns a meta-level of binary classifiers following a stacking approach [Wolpert 1992] that allows combining the predictions of the classifiers in the base level, using predictions of previous classifiers as features and the true labels as outputs. Hierarchy Of Multi-label classifiERs (HOMER) [Tsoumakas 2008] is another algorithm based on BR, but specially designed to deal with domains in which the number of labels is high. It transforms a multi-label classification problem into a tree-shaped hierarchy of simpler multi-label problems containing each one a smaller number of labels. As PS is based on LP, it cannot predict labelsets which are not present in the training set. To overcome this problem Ensemble of Pruned Sets (EPS) [Read 2008] was proposed. EPS trains a set of PS, each over a subset without replacement of the instances of the original training set. Then the predictions are combined by a voting scheme using a prediction threshold. Finally, Random-k-LabelSets (RAkEL) [Tsoumakas 2011] randomly breaks the set of labels into several small-sized sets and then an LP classifier is trained over each one. Outputs are combined in a multi-label prediction by majority voting.

2.3 Multi-label Classification of Learning Objects

The specific application of MLC techniques to LOs categorization has being only proposed and explored in a couple of previous works. In [López 2012] the authors apply only two multi-label algorithms from MULAN library (namely RAkEL and MLkNN) to two different private data sets, one data set with 253 LO instances and another one with 1000 LO instances. They apply multi-label classification based on LO metadata with the goal of finding the corresponding class labels associated with the topics covered by the LO. Both algorithms are compared based on six performance metrics, showing that the RAkEL algorithm tends to present better results than MLkNN.

In [Aldrees 2016] the authors present a comparison of only four MLC algorithms (ECC, RAkEL, EPS and MLkNN) on the basis of 16 performance metrics. They use a data set containing 658 LO instances extracted from the ARIADNE repository. Their objective is to find the best multi-label classification algorithm for categorizing multi-labeled LOs. They have also used MULAN library and the results obtained show that ECC prevails over the other three algorithms.

Although these previous works have proposed the application of MLC for LO categorization, both have considered a reduced number of MLC algorithms and their results diverge on the best algorithm identified. In our work we carry out a deeper experimentation comparing most MLC algorithms available and applying statistical test in order to check statistically significant differences. Besides, we propose a novel contribution: we analyse the possibility of enhancing classification performance by means of using information about LO usage. See [Tab. 1] for a concise comparison of previous works and ours.

Ref.	Repository	#LO	#features	#labels	#algo-rithms	#metrics	Information used
[López 2012]	Private repository	253	1442	38	2	6	IEEE LOM features
	Lornet and Merlot	1000	1442	-			
[Aldrees 2016]	ARIADNE	658	3500	30	4	16	IEEE LOM features
Our work	AGORA	519	1336	5	13	16	IEEE LOM features and usage data

Table 1: Comparative of our approach with previous research

3 Proposed Approach

Our proposed approach towards automatically recommending the categories (subjects areas or disciplines) that a LO belongs to when a user adds it to a repository is shown in [Fig. 1].

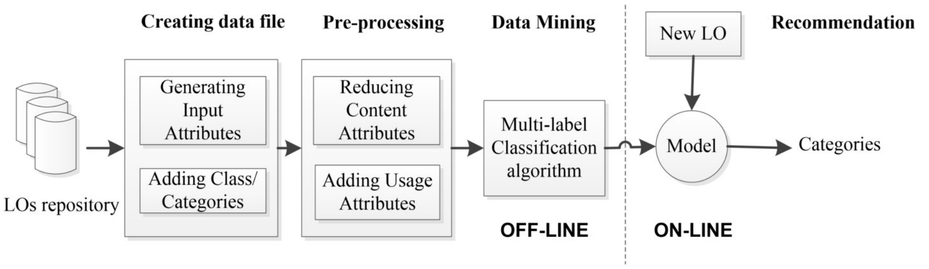


Figure 1: LO recommendation approach

Our approach has two stages, as customary in machine learning and data mining solutions: an off-line one and an on-line one [see Fig. 1].

The off-line stage has several steps. The first step consists in creating a data file from the LOs metadata. The terms or pure text features that characterize LOs are extracted from the LOR database and transformed to a suitable format. From each LO title, key words and description, significant and meaningful terms are extracted. Then, we count the number of instances of each term inside each LO and this count is represented in matrix form. Next a data pre-processing is carried out. Usually, the number of content attributes describing LOs tends to be very high, and this implies very long processing times. This is the reason for performing an attribute selection. Besides, we also try to improve classification performance by adding usage attributes conveying information related to past usage of the LOs. The number of downloads (to save LO to your computer), the number of visualizations (to show the content of LO using a viewer) and the number of evaluations (to evaluate the quality of learning objects using a questionnaire) of the LO provide valuable information that can be a hint of the potential utility of a new LO for future users. The off-line processing finishes with a data mining step in which several multi-label classification algorithms are compared over different performance metrics in order to select the most suitable algorithm and hence to obtain an optimal model for the on-line classification of new LOs.

The on-.line stage is able to categorize/classify an unlabelled LO, effectively by discovering the disciplines or subject areas to which it could be useful. In this way, when a user adds a new LO to the repository, we can recommend automatically some disciplines or subject areas that the LO could belongs to, due to this is not compulsory to introduce them. We propose to use a metric that calculates the similarity degree (*Sim*) between two LOs that it based solely on quantifying the co-occurrence of values in the metadata to determine their importance. This context similarity value (between 0 and 1) of an object (FOx_{CS}) is calculated using the following equation [Zapata 2013]:

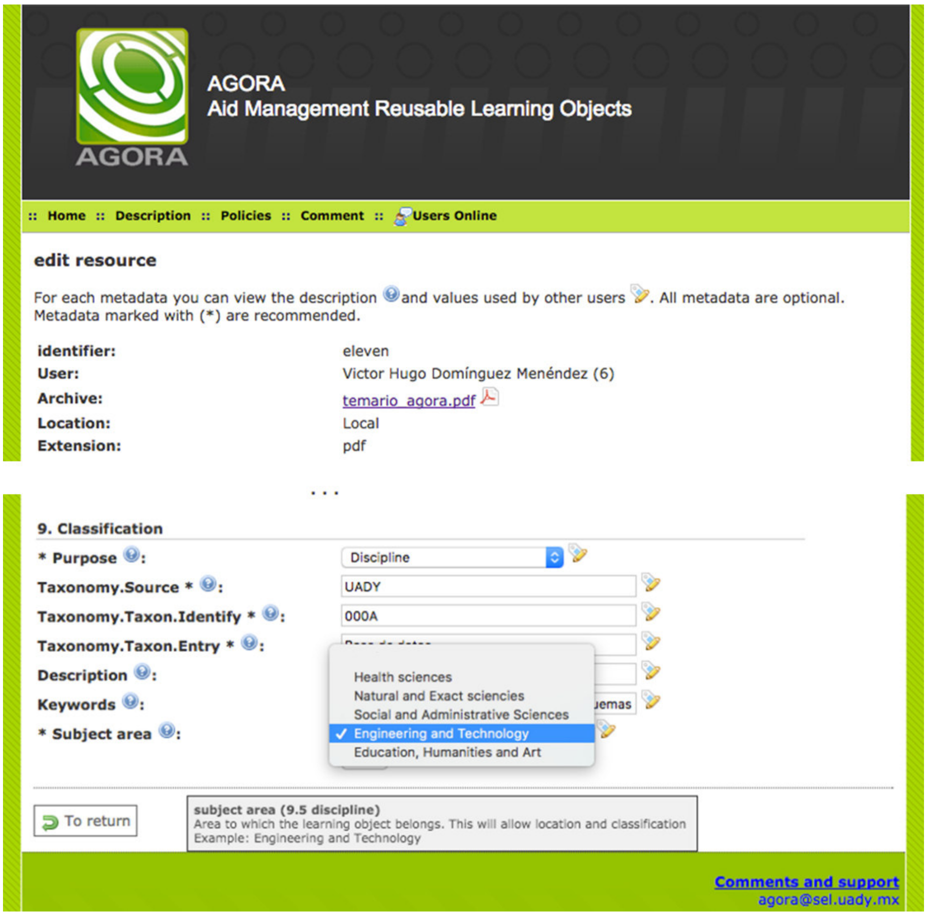
$$FOx_{CS} = Sim(O_x, O_y) = \frac{\sum_{m \in M} simMeta(m_x, m_y)}{|M|}$$

Where $|M|$ is the total number of metadata to compare and simMeta(m_x, m_y) is the semantic distance between one LO metadata m(O_x) and other m(O_y) considering the average metadata similarity. We then ranking all the LO based on their context similarity value with the new LO and select the first one as the most similar. Finally, we add this usage information of the most similar LO to the context features of the new LO, in order to the previously selected multi-label classification algorithm recommend us the disciplines or subjects areas of the new LO. It is important to notice that this paper only show the experimental part of the off-line stage in order to answer to the three research questions.

4 Experimental Work

4.1 Data Description

The data used in this work has been obtained from AGORA repository [Zapata 2013] from 2009 to 2016. When users add new LOs to AGORA, they must provide all its metadata such as the title, keywords, description and other related data such by filling a specific form [see Fig. 2]. Users can also specify the subject areas (one or several) to which the LO belongs to from five academic disciplines. The categories used in our work are generic and based on the areas of knowledge that are defined by the University of Yucatán in Mexico. Our five specific subject areas are: Engineering and Technology; Natural and Exact Science; Social and Administrative Science; Education, Humanities and Art; and Health Science. These five categories are defined by the university using AGORA and correspond to the labels that our system is able to output. The distribution (in percentages) of LOs corresponding to each of the categories in our data set is as follows: Engineering and Technology (31%); Education, Humanities and Art (27%); Natural and Exact Science (18%); and Health Science (14%); and Social and Administrative Science (10%). And about a half of these LOs have two or more categories assigned in AGORA (54% only one category, 39% two categories and 7% three categories). So, multi labelling is really relevant in this particular context.


AGORA
AGORA
Aid Management Reusable Learning Objects
:: Home :: Description :: Policies :: Comment :: Users Online

edit resource

For each metadata you can view the description and values used by other users. All metadata are optional. Metadata marked with (*) are recommended.

identifier:	eleven
User:	Victor Hugo Domínguez Menéndez (6)
Archive:	temario_agora.pdf
Location:	Local
Extension:	pdf

. . .

9. Classification

* Purpose:	Discipline
Taxonomy.Source *:	UADY
Taxonomy.Taxon.Identify *:	000A
Taxonomy.Taxon.Entry *:	Base de datos
Description:	
Keywords:	uemas
* Subject area:	Engineering and Technology

Health sciences
Natural and Exact sciencies
Social and Administrative Sciences
✓ Engineering and Technology
Education, Humanities and Art
To return

subject area (9.5 discipline)
Area to which the learning object belongs. This will allow location and classification
Example: Engineering and Technology

Figure 2: LO recommendation approach

In this work we have used data from 519 LOs. From the title, keywords and description of all the LOs we extracted 1336 terms (content features), after removing stop words and stemming (to reduce the terms to their roots). Next, we compute the frequency of these roots for the LO at issue obtaining their term frequency (TF) representation [Ochoa 2008]. So, we obtained a LO-term matrix, in which each element represents how many times a term appears in an example. We normalized the count of term frequency to measure the importance of a term and we add this content information about 1336 attributes [see Tab. 2].

Next, we have added usage information attributes about LOs on a yearly basis from 2009 to 2016. We have added information about the frequency or percentage in the number of downloads, the number of visualizations and the number of evaluations of the LOs on each academic discipline or category. In order to obtain the Learning Objects frequency of use per category (a value between 0 and 1) we have used three different equations. For example, the download frequency of a leaning object (FOxUsage) is calculated using the following equation:

$$FOx_{Usage} = \frac{\sum_{I=1}^{N} DOx_i}{MaxDOy}$$

where DOxi is the number of downloads of a Learning Object (Oxi) and MaxDOy is the maximum number of downloads that a learning object has (Oy) in the repository. In the same way, the visualization and evaluation frequencies of a learning object are calculated using similar equations. We have in total 15 usage attributes (3 usage information * 5 categories) for each LO [see Tab. 2]. We include [Fig. 3] showing the cumulative figures corresponding to past usage of the LOs used in our work. On the one hand, we can see that the LOs usage starts with high values as a result of the repository's novelty, but then the increment decreases until it becomes normal in a range of values. In our case, there is a higher increment in the usage of LOs during the three four years (from 2009 to 2012) than in the last five years (from 2012 to 2016). On the other hand, we can see that the number of visualiztion is much higher than the number of downloads and the last is the number of evaluations. A in deep analysis of the management activity of the AGORA repository can be consulted in [Menendez 2012].

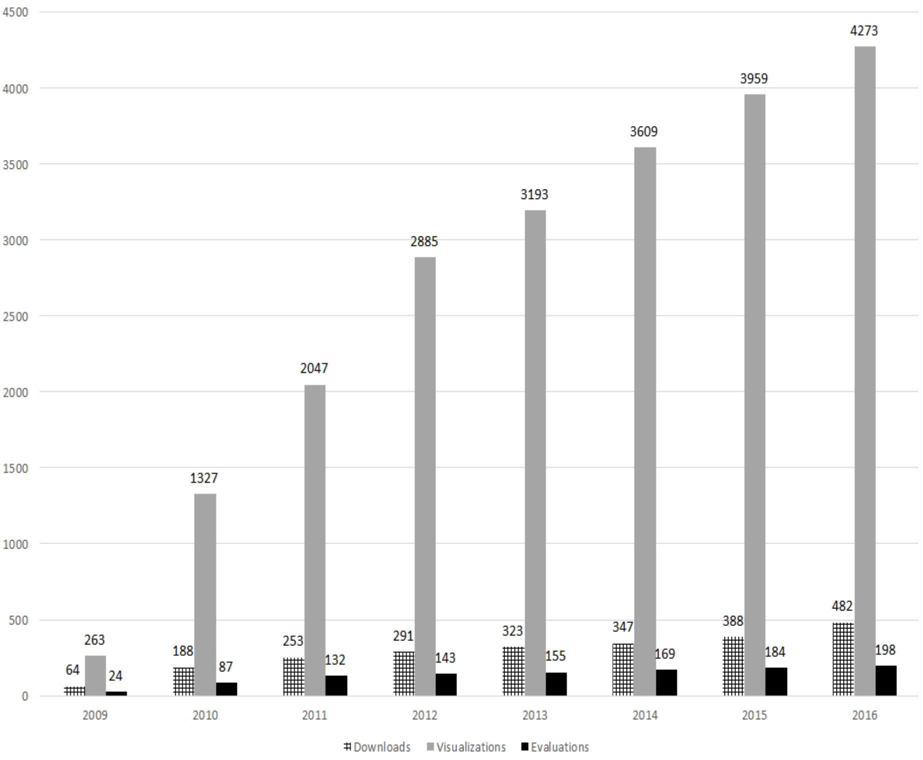


Figure 3: LO usage over the years

Finally, we added 5 class labels (in binary format) to each LO as possible classes to predict [see Tab. 2].

Attribute number	Description
1 to 1336	Content information about the terms that appears in the LO (real number between 0 and 1 - normalized count of the number of times that a term appears in the title / key words / description of a LO).
1337 to 1351	Usage information of the LO during previous years (real number between 0 and 1 – normalized count of the number of downloads, visualizations and evaluations for a LO. Since an LO can belong to several categories, these values are obtained for each discipline.
1352 to 1356	Label about the class information (Boolean 0 or 1 indicating the subject areas that the LO belongs -1- or not -0-).

Table 2: Attribute description

4.2 Experimental Setting

We have used the MLDA framework [Moyano 2018a] for pre-processing and preparing data for doing multi-label classification. MLDA is a tool for the exploration and analysis of multi-label datasets. It comprises a GUI and a Java API, providing the user with a wide set of charts and metrics about datasets, methods for transforming and preprocessing multi-label data, as well as functionalities for the comparison of several datasets. The MULAN library for MLC [Tsoumakas 2011b] has been used for running all the MLC algorithms. A set of 13 different MLC algorithms have been applied to the data set, 3 algorithm adaptation methods: AdaBoost.MH, Multi-Label k-Nearest Neighbour (MLkNN) and Instance-based Logistic Regression (IBLR) and 10 problem transformation algorithms in which the J48 implementation of the C4.5 decision tree algorithm has been used as base classifier: Binary Relevance (BR), Classifier Chains (CC), Calibrated Label Ranking (CLR), Label Powerset, Pruned Sets (PS), Ensemble of Pruned Sets (EPS), Ensemble of Classifier Chains (ECC), Random-k-LabelSets (RAkEL), Hierarchy Of Multi-label classifiERs (HOMER) and Multi-Label Stacking (MLS). These algorithms have been selected as they can be considered de state-of-art in multi-label classification as we can see in the background section.

We have configured the following parameters in the MLC algorithms. AdaBoost.MH used decision stump as base learner. BR, LP, CC, PS and CLR transformations were run with the classical J48 as a base algorithm. IBLR used k=10 nearest neighbours. HOMER was run with a BR with J48 base classifier and 3 clusters. RAkEL was run with its default parameter setting, that is, an LP with a J48 base classifier, subset size equal to 3, number of models equal to twice the number of labels and 0.5 as threshold value. MLkNN was configured with k=10 nearest neighbours and a smoothing factor of 1.0. MLS used J48 as base classifier. ECC used also its default configuration with J48 as base classifier, 10 models, using confidences and sampling with replacement. EPS used the recommended MULAN's configuration consisting of 10 models in the ensemble, strategy A (keeping the top b = 2 ranked subsets), 66% of data to sample, J48, a threshold of 0.5 and pruning label sets occurring less than p=3 times. In all the executions we used a 10-fold cross validation with 10 seeds.

Finally, we have used the 16 evaluation measures for assessing classification performance. We have selected these metrics because they are the state-of-art in MLC evaluation metrics and they have been used in similar recent works [Aldrees 2016]. They can be categorized into example-based and label-based:

- Example-based metrics are computed for each instance and then these values are averaged. Example-based metrics include metrics to evaluate bipartitions (Hamming loss, subset accuracy, precision, recall, F–measure and accuracy) and metrics to evaluate rankings (average precision, coverage one error and ranking-loss).
- Label-based metrics compute any binary evaluation metric by considering the number of true positives (tp), true negatives (tn), false positives (fp) and false negatives (fn). As a contingency table is computed for each label, two different strategies can be applied to average the values of the metric. The macro-average approach first computes the metric for each label and then averages these values. The micro-average approach first aggregates the values of all the contingency tables into one single table and then computes the metric. This way, macro and micro-averaged

precision, recall and f-measure can be considered. The definition of these metrics can be found in [Gibaja 2015].

4.3 Experiments

We have carried out a series of experiments that allow us to study different factors affecting classification performance. Firstly, we have applied an attribute selection technique in order to analyse the influence of the number of attributes on classification performance. Then, we have also studied if classification performance can be enhanced taking into consideration historic information about past usage of the LO. Besides, we carried out a comparison of classification performance of different MLC algorithms for selecting the best with our data. These experiments are described below.

4.3.1 RQ1: Can we find the minimal number of LO text features that assure sufficiently good classification performance?

We have carried out an first experiment using only content features and labels for testing what is the effect of reducing the number of attributes in the MLC performance. Overall, the time employed by a MLC algorithm in order to generate a model is proportional to the number of training instances and the number of attributes describing each instance. Our hypothesis is that, if we reduce the number of attributes then the computational cost will be reduced as well. However, a reduction of the number of attributes could discard relevant information and hence the induced model could perform poorly. Therefore, we have performed an attribute selection with different reduction levels in order to find the minimal set of fatures that assure sufficiently good performance. We want to reduce the number of attributes in the dataset with the goal of enhancing training and classification times and removing noisy and irrelevant attributes, which can have a negative impact on performance. Feature selection has been performed as suggested in [Tsoumakas 2011a]. First, the $\chi2$ feature ranking method was separately applied to each label. Thus, for each label, the worth of each attribute is estimated by computing the statistic with respect to the label to determine its independence. The rationale behind is that if an attribute is independent on a class, this attribute could be removed. This way we obtain for each label a score for each feature according to the $\chi2$ statistic. The top n features, where n is set by the user, are selected based on their maximum score over all labels.

Our original data set contains 519 LO instances, each one characterized by 1336 textual attributes. From these, we have selected 1000, 750, 500, 250, 150, 100 and 50 attributes with highest ranking to create different datasets. Next, we have applied 13 MLC algorithms to each different version of the data set, in order to know if there are significant differences in computational costs and performance by checking execution time and 16 evaluation measures.

Regarding execution time, we have found a significant reduction of computational costs (algorithms training time) as the number of features decreases from using all attributes (1336) to only the best 50 attributes [see Fig. 4], especially until 250 attributes.

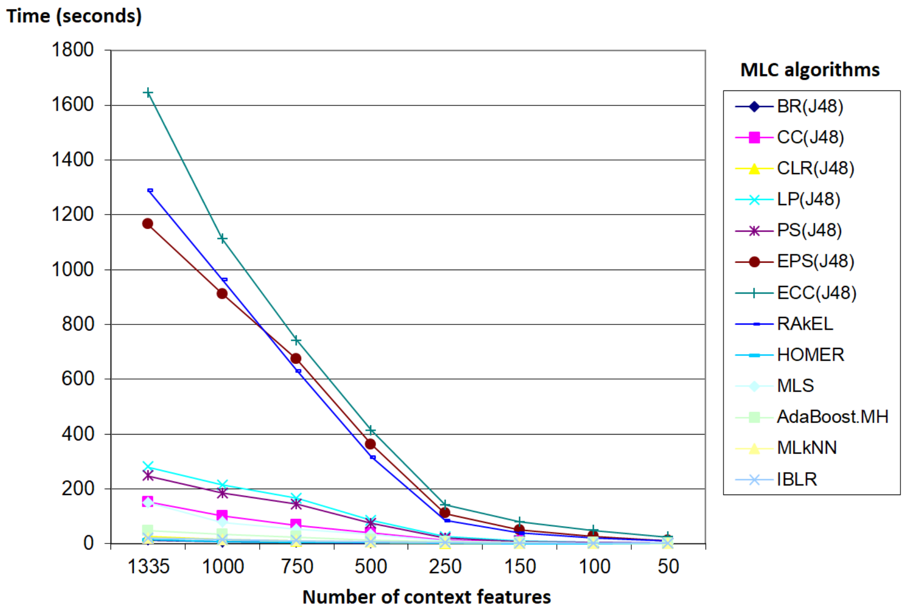


Figure 4: Algorithm training time when reducing the number of content attributes

Regarding algorithm performance, we have also found differences in all the algorithms for each evaluation measure when we reduce the number of used content attributes. As an example of classification evaluation performance, [Tab. 3] shows results obtained for one evaluation metrics (Hamming loss). We can see that not always the best results (lower Hamming loss value) are obtained when using all the attributes (1336).

To compare the classification performance of the algorithms by considering results for each feature reduction level (i.e. using all, 1000, 750, 500, etc. features), a Friedman test [Demšar 2006] has been carried out for each evaluation measures. The Friedman test is a non-parametric test that compares the average ranks of the reduction levels, where the reduction level with the best metric value for a certain algorithm is given a rank of 1 for that algorithm, the reduction level with the next best metric value is given a rank of 2 and so on. Finally, the average ranks for each reduction level are calculated. These ranks let us know which reduction level obtains the best results considering all algorithms. In this way, the reduction level with the value closest to 1 indicates the best reduction level in most algorithms.

Alg./Num. features	**1336**	**1000**	**750**	**500**	**250**	**150**	**100**	**50**
BR(J48)	**0,238**	0,242	0,239	0,241	0,243	0,246	0,248	0,248
CC(J48)	0,251	0,257	**0,250**	0,253	0,253	0,255	0,259	0,259
CLR(J48)	0,240	0,244	**0,239**	0,242	0,242	0,244	0,248	0,248
LP(J48)	0,254	0,259	0,260	**0,249**	0,265	0,263	0,263	0,265
PS(J48)	0,259	0,254	0,264	**0,248**	0,260	0,265	0,265	0,268
EPS(J48)	**0,233**	0,234	0,241	0,241	0,257	0,263	0,263	0,268
ECC(J48)	**0,234**	0,237	0,242	0,247	0,246	0,247	0,254	0,256
RAkEL	**0,226**	0,234	0,236	0,233	0,241	0,242	0,249	0,249
HOMER	0,249	0,243	**0,242**	0,244	0,243	0,245	0,249	0,250
Stacking	**0,238**	0,242	0,245	0,246	0,245	0,246	0,248	0,248
AdaB.MH	0,258	0,252	**0,248**	0,249	0,249	0,249	0,249	0,249
MLkNN	0,252	0,244	0,253	**0,241**	0,249	0,254	0,250	0,244
IBLR	0,246	0,249	0,240	**0,238**	0,241	0,247	0,248	0,248

Table 3: Algorithm performance on each reduced dataset (Hamming loss)

Friedman's p-value = 0.000002		Bonferroni-Dunn post test	
Reduction level	Ranking(order)	p-value	Null hypotheses
50	6.8077 (8)	0.000369	**Rejected**
100	6.5 (7)	0.001378	**Rejected**
150	5.7308 (6)	0.024319	**Rejected**
250	4.1923 (5)	1.305376	Accepted
1000	3.7692 (4)	2.649343	Accepted
750	3.1154 (3)	5.889487	Accepted
500	2.9615 (2)	6.776473	Accepted
1336 (control)	2.9231 (1)		

Table 4: Friedman test and Bonferroni-Dunn's post-test for Hamming loss

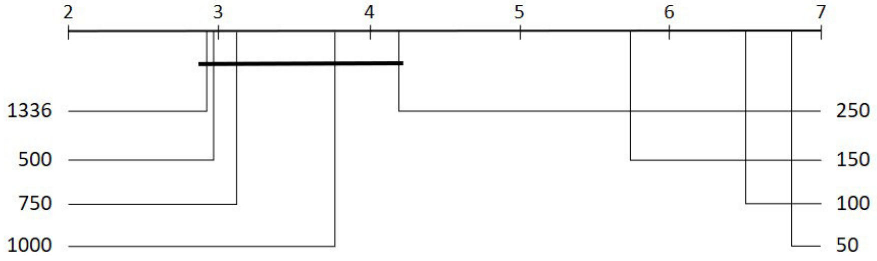


Figure 5: Critical diagram of Bonferroni-Dunn's post-test at 95% confidence for Hamming loss on each reduction level

As an example, [Tab. 4] shows the results obtained by Friedman's test for Hamming loss. We have done the same with the 16 evaluation measures. The p-value (≤ 0,05) evidenced significant differences among reduction levels with high confidence level (95%). Next, in order to determine which reduction levels present significant differences, a Bonferroni-Dunn's post-test was performed. Results for Hamming Loss are also shown in [Tab. 4]. The control reduction level was the one with the best metric (i.e. using all features). The p-valued obtained show that there are significant differences among 50, 100 and 150 reduction levels and the rest at 95% confidence level. Therefore, the cut point given by Bonferroni-Dunn's test for Hamming loss is 250 attributes. We have also included a critical diagram corresponding to the Bonferroni-Dunn's post-test at 95% confidence for Hamming loss on each reduction level [see Fig. 5].

Measure/Num. features	**1336**	**1000**	**750**	**500**	**250**	**150**	**100**	**50**	**Cut point**
↑Average precision	4.19	**3.11**	3.42	3.77	3.88	5.46	6.23	5.92	150
↓Coverage	4.5	**2.19**	3.42	4.11	4.11	5.65	6.38	5.61	250
↑Example-based accuracy	**3.08**	3.23	3.58	3.35	3.96	5.58	6.50	6.73	150
↑Example-based f-measure	4.77	4.38	3.65	**3.35**	**3.35**	4.81	5.69	6.00	100
↑Example-based precision	3.69	4.38	**2.88**	3.27	4.19	5.88	5.85	5.85	250
↑Example-based recall	5.08	4.00	4.19	**3.58**	**3.58**	4.27	5.85	5.46	ND
↓Hamming loss	**2.92**	3.77	3.11	2.96	4.19	5.73	6.50	6.81	250
↑Macro-averaged f-measure	**2.00**	3.00	3.50	3.96	4.35	5.11	6.92	7.11	250
↑Macro-averaged precision	**2.31**	2.69	3.58	3.96	4.73	5.73	6.23	6.77	250
↑Macro-averaged recall	**2.46**	3.15	3.88	4.35	4.19	4.81	6.54	6.61	150
↑Micro-averaged f-measure	3.73	3.61	3.73	**3.58**	3.77	4.73	6.35	6.50	150
↑Micro-averaged precision	3.92	4.54	**3.11**	3.65	4.04	5.42	5.50	5.81	100
↑Micro-averaged recall	4.69	4.00	4.23	4.04	**3.61**	3.96	5.85	5.61	ND
↓One error	3.69	3.61	**3.58**	4.31	3.65	6.19	5.81	5.15	250
↓Ranking loss	4.50	**3.11**	3.88	4.42	3.96	4.88	5.96	5.27	ND
↑Subset accuracy	2.69	**2.23**	3.27	3.58	4.96	5.73	6.54	7.00	250
Average ranking	3.64	**3.44**	3.56	3.76	4.03	5.25	6.17	6.14	

Table 5: Average ranking and cut point for all metrics in each reduction level (↓ minimized metric, ↑ maximized metric)

Following the detailed procedure, the obtained ranking values for all metrics in each reduction levels are shown in [Tab. 5]. Each row in [Tab. 5] shows performance ranking and has been obtained from tables like that of [Tab. 3]. For each metric (each row), the best ranking value is shown in bold. It can be observed that there are many metrics in which the best ranking value is obtained by using less than the whole set of

1336 textual features. Thus, average precision, coverage, ranking loss and subset accuracy get the best ranking value for 1000 features. Example-based precision, micro-averaged precision and one error get the best ranking value for 750 features. Example-based f-measure, example-based recall and micro-averaged f-measure get the best ranking value for 500 features. Finally, example-based f-measure, example-based recall and micro-averaged recall get the best ranking value for 250 features. [Tab. 5] also represents in the last row the average rankings obtained. Again, it can be noted that, in average, the best ranking value is not obtained with the whole set of 1336 features but with 1000 features.

Friedman's p-values (≤ 0,05) show significant differences among reduction levels with high confidence level (95%) except for example-based recall, micro-averaged recall and ranking loss. This means that, for these three metrics there are not significant differences (ND) on using the complete set of features or a reduced set of features. For the rest of metrics significant differences among reduction levels were detected and we show the cut point (see last column of [Tab. 5]) for all the metrics obtained by all the Bonferroni Dunn post-tests performed. There are no significant differences in performance when using, at least, this number of features.

In addition, a meta-ranking (the rank of rank) of reduction levels has been computed by performing another Friedman test but, in this case with the ranking values gathered in [Tab. 6]. This way we can statistically evaluate which number of features has the best overall performance in most of the metrics in order to obtain a resulting meta-rank of reduction levels. It is remarkable that the best ranking value does not correspond to the complete feature set. As the test detected significant differences between reduction levels (p-value ≤ 0.05), a Bonferroni-Dunn post-test was performed. This test found that algorithms performed significantly worst with 150 attributes or less at 95% confidence level. So, we established 250 as our best overall reduction level for our dataset due to this is the lower number of attributes that we can select without losing significantly performance. A critical diagram corresponding to [Tab. 6] is provided in [Fig. 6].

Friedman's p-value = 7.801870260948363E-11		Bonferroni-Dunn post test	
Reduction level	Ranking (order)	p-value	Null hypotheses
50	7.4062 (8)	0	**Rejected**
100	7.2812 (7)	0	**Rejected**
150	6.00 (6)	0.000372	**Rejected**
250	3.7188 (5)	1.115387	Accepted
1336	3.2812 (4)	2.56899	Accepted
500	3.0938 (3)	3.450742	Accepted
750	2.7188 (2)	5.604091	Accepted
1000 (control)	2.50 (1)		

Table 6: Friedman test and Bonferroni-Dunn's post-test for meta ranking of reduction levels

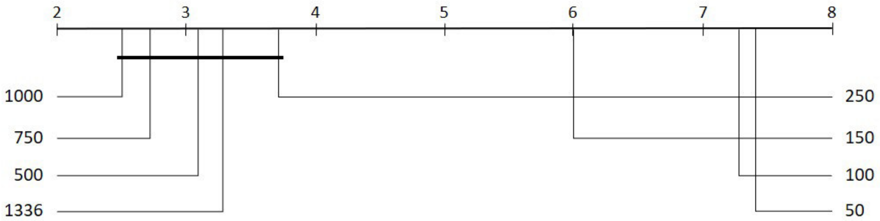


Figure 6: Critical diagram of Bonferroni-Dunn's post-test at 95% confidence for meta ranking of reduction levels

4.3.2 RQ2: Can we enhance classification performance of a LO by adding to the textual features historical usage information of the LO?

The LOs studied were compiled from year 2009 in AGORA. In the following years, these LOs have been used by a diverse community of teachers. AGORA repository record information about LO's usage (the number of downloads, the number of visualizations and the number of evaluations of each LO). These statistics are provided in total number or grouped by subject areas or discipline. It is also important to notice that we have detected that that in some cases, instructors of other areas have used LOs initially associated (by their labels) to other specific areas. So, in this work, we will use LOs usage information grouped in the five areas or disciplines. In fact, we are interested in studying if this information about actual usage of LOs could help improve classification performance.

We have carried out this second experiment using content features, usage features and labels for testing what is the effect of increasing the number of years of accumulative usage information in the MLC performance. In order to do it, we have obtained a series of data sets characterised by that amount of past usage information accumulated on a yearly basis. All these data sets use the same 250 attributes previously selected in the first experiment described in previous section plus accumulative information about usage. Hence, we have a first data set with no usage information at all, corresponding to the beginning of year 2009 (Content). Then we have a second data set with the same content data but with the new usage attributes recording to the usage data recorded through year 2009 (2009u). The third data set covers all usage data accumulated until year 2010 (2010u); and so on until year 2016 (2016u). Then, we have executed again all the 13 MLC algorithms to each one of these datasets and we have obtained the 16 evaluation measures. As an example, we present in [Tab. 7] the Hamming loss values obtained for each algorithm and data set. We can see that almost always the best results (lower Hamming loss value) are obtained when using accumulative usage information from 3 years old (2011u) and in one occasion from 2 years old (2010u).

Alg./Data	**Content**	**2009u**	**2010u**	**2011u**	**2012u**	**2013u**	**2014u**	**2015u**	**2016u**
BR(J48)	0,2470	0,2462	0,1571	**0,1410**	0,1556	0,1618	0,1738	0,1888	0,1772
CC(J48)	0,2584	0,2604	0,156	**0,1423**	0,1605	0,1624	0,1754	0,185	0,1791
CLR(J48)	0,2462	0,2489	0,1537	**0,1394**	0,1568	0,1610	0,1753	0,1868	0,1868
LP(J48)	0,2399	0,2352	0,1636	**0,1539**	0,1860	0,1947	0,2049	0,222	0,2032
PS(J48)	0,244	0,2296	**0,1592**	0,1597	0,1848	0,1901	0,2093	0,2246	0,2114
EPS(J48)	0,2301	0,2226	0,1492	**0,1413**	0,1629	0,1678	0,176	0,1856	0,1863
ECC(J48)	0,2366	0,2333	0,1445	**0,1357**	0,1531	0,1572	0,1647	0,1747	0,1728
RAkEL	0,2234	0,2219	0,1472	**0,1325**	0,1568	0,1599	0,1738	0,181	0,1787
HOMER	0,2477	0,2508	0,1768	**0,156**	0,1722	0,1799	0,1911	0,2011	0,1984
Stacking	0,2462	0,2454	0,1548	**0,1398**	0,1568	0,1684	0,1776	0,1842	0,1772
AdaBoos. MH	0,255	0,2697	0,2697	**0,2481**	0,2678	0,2643	0,2682	0,2693	0,2693
MLkNN	0,2408	0,2477	0,2003	**0,183**	0,1884	0,1961	0,2035	0,2061	0,2092
IBLR	0,2265	0,2415	0,1884	**0,1849**	0,1907	0,1892	0,1945	0,198	0,2049

Table 7: Hamming loss of algorithms on each accumulative year of usage

As we did in the previous section, a Friedman test has been conducted for each metric in order to determine if there are significant differences in performance obtained with accumulated usage datasets. For example, we present in [Tab. 8] results for Hamming loss evaluation metric. The p-value obtained shows that there are significant differences between datasets so that a Bonferroni-Dunn's post-test has been also performed corresponding the control case to usage information in year 2011. Hypotheses that have a p-value ≤ 0.05 are rejected at 95% confidence level which means that there are significant differences among these past usage information and the rest. In this case the cut point given by Bonferroni-Dunn's test show that there are not significant differences in the results obtained with the usage information of years 2013, 2012, 2011 and 2010. The corresponding critical diagram is shown in [Fig. 7].

Friedman's p-value = 5.068545583242212E-11		Bonferroni-Dunn post test	
Usage info	Ranking(order)	p-value	Null hypotheses
2009u	8.4231 (9)	0	**Rejected**
Content	8.0769 (8)	0	**Rejected**
2015u	6.6923 (7)	0.000001	**Rejected**
2016u	6.1538 (6)	0.000018	**Rejected**
2014u	5.1538 (5)	0.001179	**Rejected**
2013u	3.7692 (4)	0.097573	Accepted
2012u	2.9231 (3)	0.685388	Accepted
2010u	2.7308 (2)	0.989171	Accepted
2011u (control)	1.0769 (1)		

Table 8: Hamming loss Friedman test and Bonferroni-Dunn's post-test for each accumulative year of usage

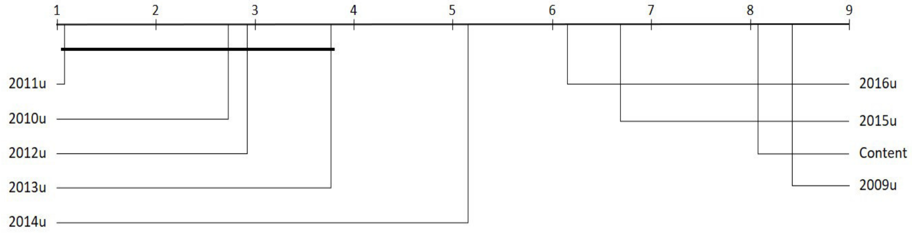


Figure 7: Critical diagram of Bonferroni-Dunn's post-test at 95% confidence for Hamming loss on each accumulative year of usage

Meas./Year	Content	2009u	2010u	2011u	2012u	2013u	2014u	2015u	2016u	Cut point
↑Avg prec	8.46	7.92	2.85	**1.38**	3.31	3.92	5.23	6.15	5.77	2010
↓Coverage	8.38	8.00	3.92	**1.54**	3.08	3.46	4.46	6.15	6.00	2010
↑E-based acc	8.08	8.38	3.23	**1.15**	3.15	3.46	5.15	6.31	6.08	2010
↑E-based f-m	7.92	8.54	3.77	**1.15**	3.08	3.31	5.31	6.31	5.62	2010
↑E-based prec	8.38	7.77	3.00	**1.15**	3.19	3.88	5.15	6.62	5.85	2010
↑E-based rec	6.08	7.15	6.08	**1.69**	4.08	3.85	3.92	6.54	5.62	2012
↓Ham loss	8.08	8.42	2.73	**1.08**	2.92	3.77	5.15	6.69	6.15	2010
↑Macro-avg f-m	8.38	8.15	2.85	**1.38**	3.31	3.69	4.85	6.23	6.15	2010
↑Macro-avg prec	8.54	7.62	1.92	**1.77**	3.92	4.15	4.69	6.31	6.08	2010
↑Macro-avg rec	8.08	8.23	5.08	**2.23**	3.31	2.77	3.62	5.92	5.77	2010
↑Micro-avg f-m	7.92	8.54	2.85	**1.08**	3.08	3.77	5.15	6.46	6.15	2010
↑Micro-avg prec	8.46	7.69	2.38	**1.15**	3.08	3.92	5.38	6.77	6.15	2010
↑Micro-avg recall	7.08	8.00	5.85	**1.77**	3.69	3.08	3.92	6.08	5.54	2012
↓One error	8.38	8.00	2.54	**1.38**	3.46	3.85	5.81	6.19	5.38	2010
↓Ranking loss	8.38	8.00	3.46	**1.38**	3.15	3.77	4.46	6.23	6.15	2010
↑Subset accuracy	8.54	8.38	3.31	**1.27**	2.50	3.65	4.58	6.42	6.35	2010
Average ranking	8.07	8.05	3.49	**1.41**	3.27	3.64	4.80	6.34	5.93	

Table 9: Average rankings for all metrics and usage data (↓ minimized metric, ↑ maximized metric)

[Tab. 9] summarises the ranking obtained for all the evaluation metrics in each accumulative usage year. These results show that optimum performance is obtained when taking into consideration usage data accumulated through three years old (until

2011), but performance is even not significantly degraded if we used only data compiled through two years old (until 2010).

As in the previous experiment, we have obtained a meta-ranking, this time corresponding to usage information obtained through the years, in order to evaluate which year has the best overall performance in most of the metrics. The results are shown in [Tab. 10]. The corresponding critical diagram is provided in [Fig. 8].

Again, we can see that there is not statistically significant difference for accumulative usage information of years 2010, 2011, 2012 and 2013. And although the best results are obtained with 2012, we have selected 2010 as the best cut point year for our dataset, since it is the lower one and in this way we only have to wait during two years of accumulative usage information in order to obtain good performance with MLC algorithms.

Friedmans' p-value = 8.938327855645412E-11		Bonferroni-Dunn post test	
Past usage	Ranking (order)	p-value	Null hypotheses
Content	8.4688 (9)	0	**Rejected**
2009u	8.4375 (8)	0	**Rejected**
2015u	7.0625 (7)	0	**Rejected**
2016u	5.8125 (6)	0.000005	**Rejected**
2014u	4.8125 (5)	0.000659	**Rejected**
2013u	3.5 (4)	0.078586	Accepted
2010u	3.1562 (3)	0.207597	Accepted
2012u	2.75 (2)	0.565609	Accepted
2011u (control)	1 (1)		

Table 10: Friedman test and Bonferroni-Dunn's post-test for meta ranking of yearly usage information

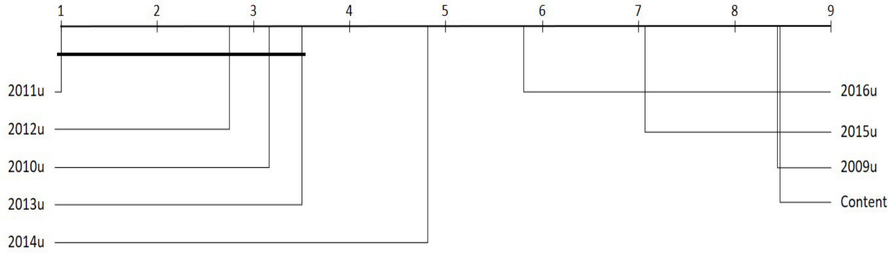


Figure 8: Critical diagram of Bonferroni-Dunn's post-test at 95% confidence for meta ranking of usage data

4.3.3 RQ3: Can We Select an Algorithm or Group of Algorithms as the Best Performers for Categorization Recommendation?

In this third experiment, another similar statistical analysis to the two previous ones has been carried out in order to identify what is the best performing MLC algorithm. For this final study we have used the previously selected configuration corresponding to the data set with the best 250 attributes and with usage data until year 2010, in accordance with the conclusions obtained in the preceding experiments. Starting from the 16 performance evaluation metrics obtained by the 13 MLC algorithms we have derived a final ranking where we record the algorithms performing ranking for each metric [see Tab. 11]. Looking at the average ranking in [Tab. 11] we can see that ECC is the best average ranked MLC algorithm because it has the highest ranking more times (5 out of 13 times) and when it hasn't the highest it always presents a quite good ranking (it never has a ranks under 4). For example, if we compare to CLR, this algorithm is the best one for 4 metrics, but for the other metrics it has a quite variable ranking interval and RAkEL, an algorithm that is the best one for only one metric, has an average ranking better than CLR.

Finally, we performed a Friedman test [see Tab. 12] post-test over the ranking results that show that ECC is the best performing MLC algorithm (selected as control algorithm), but the difference in performance is not statistically significant with some other algorithms such as CC, EPS, CLR and RAkEL. The results of the Bonferroni-Dunn test are shown in [Tab. 12]. Hypotheses that have a p-value ≤ 0.05 are rejected at 95% confidence level. The corresponding critical diagram is shown in [Fig. 9].

The results obtained [see Tab. 12] show that there are some MLC algorithms that can be used for classifying our LOs with a similar performance with our dataset. Then, if we are interested in having the best possible performance, we should use ECC, but we must know that there are also other algorithms with similar performance that we could choose like CC, EPS, CLR and RAkEL.

Measure/Algorith	**BR**	**CC**	**CLR**	**LP**	**PS**	**EPS**	**ECC**	**RAkEL**	**HOMER**	**MLS**	**AdaB.MH**	**MLkNN**	**IBLR**
↑Avg prec	5	8	**1**	13	4	3	2	7	11	6	12	10	9
↓Coverage	4	6	**1**	13	5	8	2	10	11	3	12	9	7
↑E-based acc	9	5	7	6	4	2	**1**	3	8	10	13	12	11
↑E-based f-m	9	6	8	7	4	2	**1**	3	5	10	13	12	11
↑E-based prec	8	6	7	5	2	**1**	4	3	10	9	13	12	11
↑E-based rec	9	7	4	8	6	5	2	3	1	10	13	12	11
↓Ham loss	7	6	4	9	8	3	**1**	2	10	5	13	12	11
↑Macro-avg f-m	6	5	4	10	9	7	**1**	2	8	3	13	12	11
↑Macro-avg prec	5	4	6	9	8	7	2	3	10	**1**	13	12	11
↑Macro-avg rec	6	5	4	10	9	8	2	3	**1**	7	13	12	11
↑Micro-avg f-m	7	5	4	10	8	3	**1**	2	9	6	13	12	11
↑Micro-avg prec	5	7	8	9	6	**1**	3	4	12	2	13	11	10
↑Micro-avg recall	7	5	4	8	9	6	2	3	**1**	10	13	12	11
↓One error	6	7	**1**	13	5	4	3	2	11	8	12	10	9
↓Ranking loss	4	5	**1**	13	6	7	2	10	11	3	12	9	8
↑Subset accuracy	8	5	4	7	6	2	3	**1**	10	9	13	12	11
Average ranking	6,56	5,75	4,25	9,38	6,19	4,31	**2**	3,81	8,06	6,38	12,75	11,31	10,25

Table 11: Algorithm performance ranking for each evaluation metric (↓ minimized metric, ↑ maximized metric)

Friedmans' p-value = 9.243494858424128E-11		Bonferroni-Dunn post test	
MLC algorithm	Ranking (order)	p-value	Null hypotheses
AdaBoost.MH	12.75 (13)	0	**Rejected**
MLkNN	11.3125 (12)	0	**Rejected**
IBLR	10.25 (11)	0	**Rejected**
LP	9.375 (10)	0.000001	**Rejected**
HOMER	8.0625 (9)	0.000128	**Rejected**
BR	6.5625 (8)	0.011052	**Rejected**
MLS	6.375 (7)	0.017829	**Rejected**
PS	6.1875 (6)	0.02827	**Rejected**
CC	5.75 (5)	0.077507	Accepted
EPS	4.3125 (4)	1.116641	Accepted
CLR	4.25 (3)	1.226818	Accepted
RAkEL	3.8125 (2)	2.256599	Accepted
ECC (control)	2 (1)		

Table 12: Friedman test and Bonferroni-Dunn's post-test for meta ranking of MLC algorithms

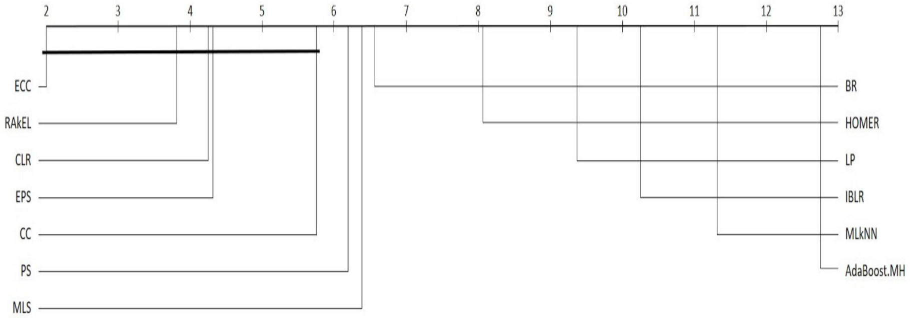


Figure 9: Critical diagram of Bonferroni-Dunn's post-test at 95% confidence for meta ranking of MLC algorithms

5 Conclusions and Future Works

In this work, we have proposed a novel approach for automatically categorizing LOs by using MLC algorithms. The novelty value of our proposal is the enhancement of LO classification performance by taking into consideration LO usage information. We have compared 13 MLC classification algorithms over 519 LOs gathered from AGORA repository and using a set of 16 performances metrics. We have used MLDA and MULAN frameworks for our experimentation.

Regarding the three research questions that we posed at the beginning of this research, the answers found after experimenting with the AGORA repository data are as follows:

1. Can we find the minimal number of LO text features that assure sufficiently good classification performance? Yes, we have found a quite smaller subset of attributes that allow a sufficiently good performance. In our case, we have been able to reduce from 1336 (all the features) to only 250 attributes without losing significant performance.
2. Can we enhance classification performance of a LO by adding to the textual features historical usage information of the LO? Yes, we can enhance classification performance by using historical information about usage of a LO. In our case, a good performance is achieved when adding usage information corresponding to only two accumulative years old with no statistical significant difference versus the best result obtained of three accumulative years.
3. Can we select an algorithm or group of algorithms as the best performers for categorization recommendation? Yes, we can select an algorithm as the best performer for categorization recommendation. In our case, ECC is the best although there are some other algorithms reaching similar performance levels like CC, EPS, CLR and RAkEL.

Currently, we are working in the integration of the best obtained classification model into AGORA for doing recommendation in real time. In our case we will integrate the model obtained by ECC algorithm (with 250 content features and 2 years of usage information) in order to recommend categories that a new LO could belong to. In this way, we will test the on-line stage of our proposal by evaluating the obtained recommendations. We also want to consider its impact in the perceived usability by users in the labelling of new objects. In a previous related experiment we have tested the interface of AGORA system in terms of its usability in general [Menéndez 2010, Menéndez 2011b, Menéndez-Domínguez 2011] and we want to extend this experiment to the specific usability of the LO recommendations.

In the future, we want to replicate the experiment using the same repository but considering a higher number of categories, for example according to a new taxonomic structure of knowledge that is based on that specified by the National Council of Science and Technology of Mexico (CONACYT). We also want to test the performance of MLC algorithms when using not only the optimal minimum number of text features and the number of usage years without decreasing the performance, but the best number of features and usage years in order to obtain the best performance. We also want to test the result obtained if we use only the usage information about the number of visualization due to the number of downloads and evaluations are much lower than the number of visualizations.

Finally, we want to do more experimentation using data from other repositories, in order to generalize our conclusions about the effect in MLC algorithms' performance and the selection of the number of years of accumulative usage information. We want to deal the problem of overfitting considering that the features are unigrams extracted from training instances and in our dataset the number of features is nearly 3 times higher than number of instances (LOs). We also want to test our proposal when using a number of categories or areas and not only five classes. We consider the study of the applicability of our system to other LORs different from

AGORA as one of the main lines for future research. The most important advantage of our proposal is the capability of recommending the categories that a newly registered LO belongs to, but this applicability is restrained by several limitations:

- The repository must store LOs in some accessible format like for example plain text, html, xml... Our system is not capable of working with LOs in binary format.
- The repository must allow classifying LOs into categories, that is, the labels that our system works with.
- In order to take full advantage of our system's potential, the repository must be able to store LO usage information.

Acknowledgements

The authors gratefully acknowledge the financial subsidy provided by Spanish Ministry of Science and Technology TIN2017-83445-P.

References

[ADL 2016] ADL: SCORM. (2016) https://www.adlnet.gov/adl-research/scorm/

[AENOR 2014] AENOR (Asociación Española de Normalización y Certificación): Perfil de Aplicación LOM-ES V 1.0 (2014). http://educalab.es/intef/tecnologia/recursos-digitales/lom-es

[Aldrees 2016] Aldrees, A., Chikh, A.: Comparative evaluation of four multi-label classification algorithms in classifying learning objects. Computer Applications in Engineering Education, 24, 4 (2016), 651-660

[ARIADNE 2016] ARIADNE Foundation: ARIADNE. (2016) http://www.ariadne-eu.org/

[Berners-Lee 2001] Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific American, 284, 5 (2001), 34-43

[Boutell 2004] Boutell, M. R., Luo, J., Shen, X., Brown, C. M.: Learning multi-label scene classification. Pattern recognition, 37, 9 (2004), 1757-1771

[Bruce 2004] Bruce, T. R., Hillmann, D. I.: The continuum of metadata quality: defining, expressing, exploiting. In: Metadata in practice

[Cechinel 2009] Cechinel, C., Sánchez-Alonso, S., Sicilia, M. Á.: Empirical analysis of errors on human-generated learning objects metadata. In: Metadata and semantic research, 46 (2009), 60-70

[Cheng 2009] Cheng, W., Hüllermeier, E.: Combining instance-based learning and logistic regression for multilabel classification. Machine Learning, 76, 2 (2009), 211-225

[Currier 2003] Currier, S., Barton, J., O'Beirne, R., Ryan, B. Quality assurance for digital learning object repositories: how should metadata be created? In: Communities of practice: research proceedings of the 19th Association for Learning Technology Conference, (2003), 130–142

[Demšar 2006] Demšar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine learning research, 7 (2006), 1-30

[Freund 1995] Freund, Y., Schapire, R. E.: A desicion-theoretic generalization of on-line learning and an application to boosting. In: European conference on computational learning theory (1995), 23-37

[Fu 2018] Fu, H., Cheng, J., Xu, Y., Wong, D. W. K., Liu, J., Cao, X.: Joint optic disc and cup segmentation based on multi-label deep network and polar transformation. IEEE transactions on medical imaging, 37,7 (2018), 1597-1605

[Fürnkranz 2008] Fürnkranz, J., Hüllermeier, E., Loza Mencía, E., Brinker, K.: Multilabel classification via calibrated label ranking. Machine learning, 73, 2 (2008), 133-153

[Gibaja 2014] Gibaja, E., Ventura, S.: Multi-label learning: a review of the state of the art and ongoing research. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 4, 6 (2014), 411-444

[Gibaja 2015] Gibaja, E., Ventura, S.: A tutorial on multilabel learning. ACM Computing Surveys, 47, 3 (2015), 52

[Han 2011] Han, J., Kamber, M., Pei, J.: Data Mining - Concepts and Techniques, 3rd ed., Morgan Kaufmann (2011)

[IEEE 2016a] IEEE: IEEE Learning Technology Standards Committee.: WG12: Learning Object Metadata. (2016) http://grouper.ieee.org/groups/ltsc/wg12/

[IEEE 2016a] IEEE: IEEE Standard for Learning Object Metadata. (2016) https://standards.ieee.org/findstds/standard/1484.12.1-2002.html

[IMS 2016] IMS Global Learning Consortium: Learning Design Specification. (2016) https://www.imsglobal.org/learningdesign/index.html

[Innis-Allen 2008] Innis-Allen, C., Mugisa, E.: A flexible taxonomy of learning objects based on content and media centric approaches to granularity. In: Proceedings of the 7th IASTED International Conference on Web-Based Education (WBE 2008), Innsbruck, Austria (2008), 275-280

[Leal Fonseca 2010] Leal Fonseca, D.: Iniciativa colombiana de objetos de aprendizaje, Apertura, 0, 8, (2010)

http://www.udgvirtual.udg.mx/apertura/index.php/apertura4/article/view/102

[López 2012] López, V. F., de la Prieta, F., Ogihara, M., Wong, D. D.: A model for multi-label classification and ranking of learning objects. Expert Systems with Applications, 39, 10 (2012), 8878-8884

[Lytras 2007] Lytras, M. D., Sicilia, M. A.: Where is the value in metadata? International Journal of Metadata, Semantics and Ontologies, 2,4 (2007), 235-241

[Magee 2001] Magee, M., Friesen, N.: Report: CAREO overview and goals (2001) http://www.careo.org/documents/overview.html

[McDonald 2006] McDonald, J.: Learning object: A new definition, a case study and an argument for change. In: Proceedings of the 23rd Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education - Who's Learning? Whose Technology? (ASCILITE 2006), Sidney, Australia (2006), 535-544

[Menéndez 2010] Menéndez, V., Prieto, M., Zapata, A.: Sistemas de Gestión Integral de Objetos de Aprendizaje. Revista Iberoamericana de Tecnologías del Aprendizaje, 5, 2, (2010), 56-62

[Menéndez 2011a] Menéndez, V. H., Zapata, A., Prieto-Mendez, M. E., Romero, C., Serrano-Guerrero, J.: A similarity-based approach to enhance learning objects management systems. In: IEEE Intelligent Systems Design and Applications (2011), 996-1001

[Menéndez 2011b] Menéndez, V. H., Castellanos, M. E., Zapata, A., Prieto, M. E.: Generación de objetos de aprendizaje empleando un enfoque asistido, Píxel-Bit - Revista de Medios y Educación, 38 (2011), 141-153

[Menéndez 2012] Menéndez Domínguez, V.H., Castellanos Bolaños, M.E., Pech Campos, S.J.: Fomento de la innovación y flexibilidad en desarrollo de objetos de aprendizaje. La plataforma AGORA, Apertura, 3, 1 (2012), 100-109

[Menendez-Dominguez 2011] Menendez-Dominguez, V. H., Zapata, A., Prieto-Mendez, M. E., Romero, C., Serrano-Guerrero, J.: A Similarity-based approach to enhance learning objects management systems. In: 11th International Conference on Intelligent Systems Design and Applications, (2011)

[MERLOT 2016] MERLOT: MERLOT - Multimedia Educational Resource for Learning and Online Teaching. (2016) https://www.merlot.org/merlot/index.htm

[Moyano 2018a] Moyano, J. M., Gibaja, E. L., Ventura, S.: MLDA: A tool for analyzing multi-label datasets. Knowledge-Based Systems, 121 (2018), 1-3

[Moyano 2018b] Moyano, J. M., Gibaja, E. L., Cios, K. J., Ventura, S.: Review of ensembles of multi-label classifiers: models, experimental study and prospects. Information Fusion, 44 (2018), 33-45

[Ochoa 2008] Ochoa, X., Duval, E.: Relevance ranking metrics for learning objects. IEEE Transactions on Learning Technologies, 1, 1 (2008), 34-48

[Ochoa 2011] Ochoa, X.: Modeling the macro-behavior of learning object repositories, Interdisciplinary Journal of E-skills and Lifelong Learning, 7 (2011), 25-35

[Paulsson 2006] Paulsson, F., Naeve, A.: Establishing technical quality criteria for learning objects. In: Exploiting the knowledge economy: issues, applications, case studies, (2006), 1431-1439

[Prieto Méndez 2014] Prieto Méndez, M.E., Menéndez Domínguez, V.H., Vidal Castro, C.L.: Metadata and ontologies in e-learning. In: Handbook of metadata, semantics and ontologies, (2014), 140-196

[Read 2008] Read, J., Pfahringer, B., Holmes, G.: Multi-label classification using ensembles of pruned sets. In: Data Mining, 2008 - ICDM'08 - Eighth IEEE International Conference on Data Mining (2008), 995-1000

[Read 2011] Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. Machine learning, 85, 3 (2011), 333-359

[Rodés-Paragarino, 2016] Rodés-Paragarino, V., Gewerc-Barujel, A., Llamas-Nistal, M.: Use of Repositories of Digital Educational Resources: State-of-the-Art Review. IEEE Revista Iberoamericana de Tecnologias del Aprendizaje, 11, 2 (2016), 73-78

[Schapire 2000] Schapire, R. E., Singer, Y.: BoosTexter: A boosting-based system for text categorization. Machine learning, 39, 2-3 (2000), 135-168

[Sicilia 2005] Sicilia, M. A., García, E., Pages, C., Martínez, J. J., Gutiérrez, J. M.: Complete metadata records in learning object repositories: some evidence and requirements. International Journal of Learning Technology, 1, 4 (2005), 411-424

[Stefaner 2007] Stefaner, M., Vecchia, E. D., Condotta, M., Wolpers, M., Specht, M., Apelt, S., et al: MACE-Enriching Architectural Learning Objects for Experience Multiplication. In Duval, E., Klamma, R., Wolpers. M., editors. EC-TEL 2007 (2007), 322-336

[Tan 2018] Tan, P.-N., Steinbach, M., Karpatne, A., Kumar, V.: Introduction to Data Mining, 2nd ed., Addison-Wesley (2018)

[Tsoumakas 2008] Tsoumakas, G., Katakis, I., Vlahavas, I.: Effective and efficient multilabel classification in domains with large number of labels. In: Proceedings of the ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08) (2008), 30-44

[Tsoumakas 2009] Tsoumakas, G., Dimou, A., Spyromitros, E., Mezaris, V., Kompatsiaris, I., Vlahavas, I.: Correlation-based pruning of stacked binary relevance models for multi-label learning. In: Proceedings of the 1st International Workshop on Learning from Multi-label Data. (2009), 101-116

[Tsoumakas 2011a] Tsoumakas, G., Katakis, I., Vlahavas, I.: Random k-labelsets for multilabel classification. IEEE Transactions on Knowledge and Data Engineering, 23, 7 (2011), 1079-1089

[Tsoumakas 2011b] Tsoumakas, G., Spyromitros-Xioufis, E., Vilcek, J., Vlahavas, I.: Mulan: a java library for multi-label learning. Journal of Machine Learning Research, 12 (2011), 2411-2414

[Wolpert 1992] Wolpert, D. H.: Stacked generalization. Neural networks, 5, 2 (1992), 241-259

[Yen 2009] Yen, N. Y., Hou, F. F., Chao, L. R., Shih, T. K.: Weighting and ranking the e-learning resources. In: Ninth IEEE International Conference on Advanced Learning Technologies, (2009), 701-703

[Zapata 2013] Zapata, A., Menéndez, V. H., Prieto, M.E., Romero, C.: A Framework for Recommendation in Learning Object Repositories: An Example of Application in Civil Engineering. Advances in Engineering Software, 56 (2013), 1-14

[Zapata 2015] Zapata, A., Menéndez, V. H., Prieto, M.E., Romero, C.: Evaluation and Selection of Group Recommendation Strategies for Collaborative Searching of Learning Objects. International Journal of Human-Computer Studies, 76 (2015), 22-39

[Zhang 2007] Zhang, M. L., Zhou, Z. H.: ML-KNN: A lazy learning approach to multi-label learning. Pattern recognition, 40, 7 (2007), 2038-2048