UNIVERSIDAD DE CÓRDOBA

Programa de doctorado:
Computación avanzada, energía y plasmas

TÍTULO:

MINERÍA DE DATOS SOBRE OBJETOS DE APRENDIZAJE

LEARNING OBJECTS DATA MINING

Tesis presentada por:

Pedro González Espejo

Directores:

Dr. D. Cristóbal Romero Morales

Dra. Dª. Eva Lucrecia Gibaja Galindo

Córdoba Septiembre 2020

TITULO: *MINERÍA DE DATOS SOBRE OBJETOS DE APRENDIZAJE*

AUTOR: *Pedro González Espejo*

© Edita: UCOPress. 2020
Campus de Rabanales
Ctra. Nacional IV, Km. 396 A
14071 Córdoba

https://www.uco.es/ucopress/index.php/es/
ucopress@uco.es

TÍTULO DE LA TESIS: MINERÍA DE DATOS SOBRE OBJETOS DE APRENDIZAJE

DOCTORANDO: Pedro González Espejo

INFORME RAZONADO DEL/DE LOS DIRECTOR/ES DE LA TESIS

(se hará mención a la evolución y desarrollo de la tesis, así como a trabajos y publicaciones derivados de la misma).

El doctorando (Pedro González Espejo) ha progresado enormemente como investigador desde que comenzara la tesis doctoral en el año 2016 en la Universidad de Córdoba. Durante estos 4 años el doctorando ha realizado todas las actividades obligatorias y opcionales, trabajado duro seguido siempre las pautas de trabajo que le hemos marcado los directores y el plan de investigación que se estableció. Como principales frutos del trabajo realizado se han derivado las dos siguientes publicaciones:

Artículo en Congreso Internacional CORE B:

González, P., Gibaja, E. Zapata, A., Menéndez, V. H., Romero, C. Towards Automatic Classification of Learning Objects: Reducing the Number of Used Features, EDM 2017. 394-395. Wuhan, China, 25-28/06/2017.

Artículo en Revista con Factor de Impacto (incluida en el JCR):

Espejo, P. G., Gibaja, E., Menéndez, V. H., Zapata, A., & Romero, C. Improving Multi-Label Classification for Learning Objects Categorization by Taking into Consideration Usage Information. Journal of Universal Computer Science. Volumen 25 (Issue 13): 1687-1716. 2019.

Por todo ello, se autoriza la presentación de la tesis doctoral.

Córdoba, 5 de julio de 2020

Firma del/de los director/es

Fdo.: Cristóbal Romero Morales

Fdo.: Eva Lucrecia Gibaja Galindo

La tesis titulada “MINERÍA DE DATOS SOBRE OBJETOS DE APRENDIZAJE”, que presenta Pedro González Espejo para optar al grado de doctor, ha sido realizada dentro del programa de doctorado computación avanzada, energía y plasmas, en la línea de investigación aprendizaje automático, modelado de sistemas y minería de datos, del Departamento de Informática y Análisis Numérico de la Universidad de Córdoba, bajo la dirección de los doctores Cristóbal Romero Morales y Eva Lucrecia Gibaja Galindo cumpliendo, en su opinión, los requisitos exigidos a este tipo de trabajos.

Córdoba, junio de 2020

El Doctorando

Fdo: Pedro González Espejo

El Director

Fdo: Dr. Cristóbal Romero Morales

La Directora

Fdo: Dra. Eva Lucrecia Gibaja Galindo

Esta tesis ha sido parcialmente subvencionada con el proyecto TIN2017-83445-P del Ministerio de Ciencia, Innovación y Universidades.

AGRADECIMIENTOS

Quiero dar las gracias a los doctores Romero y Gibaja por la enorme paciencia que han tenido conmigo. Muchas gracias a los dos.

TABLA DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

La reutilización es uno de los conceptos clave dentro de ingeniería del software, ya que permite racionalizar y agilizar el desarrollo de sistemas basándose en el aprovechamiento de componentes ya desarrollados y probados. Es natural extender este enfoque a otros campos de la informática, y es una de las bases sobre las que se asienta la idea del desarrollo de repositorios de objetos de aprendizaje (ROA, *learning object repository* - *LOR*), gracias a los cuales cualquier docente puede aprovechar objetos de aprendizaje (OA, *learning objetcs* - *LO*) desarrollados por algún otro colega. El potencial de los ROA se maximiza gracias a Internet, que permite que estos repositorios estén disponibles de manera global.

No obstante, aunque la idea básica resulta simple y evidente, tal como suele ser habitual, la dificultad radica en los detalles. Uno de los problemas a abordar es el de la búsqueda de OA. Los OA son de naturaleza variopinta y heterogénea, y no resulta obvio cuál es la mejor manera de describir un OA y hacer referencia al mismo, por lo que, cuando un usuario desea buscar un OA que pueda serle de utilidad para sus propósitos docentes, lo difícil suele ser cómo encontrar dicho OA entre todos los del repositorio. Para poder hacer este proceso de búsqueda lo más manejable posible es muy importante cuidar al máximo el trabajo previo de etiquetado de los OA, es decir, cómo caracterizamos cada OA en el momento en que lo añadimos a un repositorio.

En este trabajo proponemos un método novedoso que recomienda de manera automática las categorías a las que pertenece un OA cuando un usuario lo añade al repositorio. Se utiliza un

enfoque de aprendizaje multietiqueta, ya que cada OA puede estar asociado a diferentes categorías. Para poder satisfacer nuestro objetivo de la mejor manera posible se ha desarrollado una metodología con tres fases, de manera que primero se selecciona el conjunto de características de texto más adecuadas de entre los metadatos que describen a los OA; en segundo lugar se decide la cantidad de información histórica relativa a OA similares al nuevo OA a insertar en el repositorio que permita mejorar la calidad de la clasificación; finalmente se elige el algoritmo de aprendizaje multietiqueta que mejor resultado ofrece.

Se ha llevado a cabo un trabajo experimental sobre un conjunto de 519 OA que se han recogido en el repositorio AGORA a lo largo de 8 años. Se han comparado 13 algoritmos de clasificación multietiqueta utilizando 16 medidas de evaluación. Los resultados obtenidos muestran que la selección de características permite reducir el tiempo de ejecución sin perder precisión. También se ha podido comprobar que la utilización de información histórica acerca de OA similares al que se está añadiendo permite mejorar la calidad de la clasificación. Finalmente ha sido posible identificar un conjunto de algoritmos de aprendizaje multietiqueta que son los que mejor calidad de clasificación ofrecen sobre nuestros datos. Todo esto permite recomendar de manera automática las categorías a las que pertenece un nuevo OA añadido al repositorio.

ABSTRACT

Reuse is a cornestone concept for software engineering, because it allows to rationalize an speed up system development by leveraging on the utilization of already developed and tested components. It's logical to extend this approach to other computing fields beyond software engineering, and so, reuse is one of the rationales behind learning object repositories (LOR), which allow a teacher to use learning objects (LO) developed by some other colleague. The potential of LOR can be fully exploited thanks to the Internet, which allows repositories to be globally available.

Although the basic idea of LOR is simple an evident, as usual, the complexity hides in the details. One of the issues to be tackled is LO search. LOs are diverse and heterogeneous, and it is not obvious at all what is the better way for categorizing and referencing LOs. When a user wants to look for a LO that could be suitable for his teaching purposes, it can be really difficult to find such a LO between the contents of the repository. In order to make easier the search process it is pivotal to do a good previous work when labeling LOs, that is, the categorization of the LO when it is inserted into the repository.

In this work, we propose a novel approach for automatically recommending the categories that learning object belongs to when a user adds it to a repository. We use a multi-label learning approach since each learning object might be associated with multiple categories. In order to improve this goal, 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 usage information about the most similar LO 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 a reduction in the number of text attributes can improve time performance without losing precision. Another finding is that usage information about the most similar learning object can improve the classification. Finally, a set of algorithms which obtained the best performance in our data has been identified, so, they can be used for automatically recommending learning object categorization.

1

INTRODUCCIÓN

El hardware y las tecnologías de bases de datos disponibles nos permiten almacenar enormes cantidades de datos y acceder a ellos de manera eficiente, fiable y económica. Estas circunstancias han conducido a la proliferación de bases de datos masivas que dan lugar al problema de cómo obtener el máximo beneficio de los datos que contienen. El objetivo básico es obtener conocimiento, es decir, información valiosa, de las bases de datos. Pero el procesamiento manual de bases de datos tan enormes es totalmente descabellado, y las capacidades de consulta de los sistemas de gestión de bases de datos tradicionales no bastan.

Es en este escenario en el que hace entrada la minería de datos (MD). La MD, también conocida como descubrimiento de conocimiento en bases de datos (KDD, por las siglas en inglés de *knowledge discovery in databases*), es un proceso que consta de varios pasos en el que se aplican diferentes técnicas, métodos y herramientas para extraer patrones de conocimiento de alta calidad de bases de datos (Han y Kamber, 2011; Tan et al., 2018). No es fácil dar una definición de un proceso tan complejo. Tomamos la definición de (Fayyad et al., 1996), que establece que la MD es el proceso no trivial de identificar en los datos patrones válidos, novedosos, potencialmente útiles y esencialmente comprensibles.

En la actualidad, las instituciones educativas hacen un amplio uso de sistemas computerizados que se aplican a distintos aspectos de su funcionamiento cotidiano. Entre estos sistemas informáticos podemos destacar las plataformas electrónicas orientadas a facilitar el aprendizaje de los alumnos, como por ejemplo Moodle. Estos y otros muchos

sistemas computerizados de uso común en las instituciones educativas permiten recabar grandes cantidades de datos de diferente tipo como datos sociodemográficos de alumnos y otros miembros de la comunidad educativa, materiales de estudio, expedientes académicos, etc. Así pues, pueden obtenerse grandes cantidades de datos de este tipo, por lo que la aplicación de la MD es una idea que no tarda en surgir, dando lugar a lo que se conoce como minería de datos educativa (MDE o *EDM - educational data mining*, en inglés; ver Romero y Ventura, 2010).

El conocimiento obtenido mediante MDE permite evaluar los sistemas educativos, lo que posibilita mejorar su calidad y eficiencia. Este conocimiento es potencialmente útil para estudiantes y profesores. Por ejemplo, en base al conocimiento obtenido, se podrían hacer a los estudiantes recomendaciones, ya fueran de carácter general o individualizado, acerca de actividades y métodos de estudio que les pudieran resultar más provechosos. En el caso de los profesores, éstos podrían entender mejor los métodos didácticos que emplean y la manera en que estudian sus alumnos, lo que permitiría a los docentes adaptar sus estrategias de enseñanza a la idiosincrasia de sus estudiantes. Pero el conocimiento obtenido mediante MDE también puede ser de interés para otros actores relacionados con los procesos educativos, como los diseñadores y autores de materiales y procesos educativos, los desarrolladores de plataformas electrónicas de aprendizaje, los inspectores y gestores de la docencia, el personal de administración de las instituciones educativas, etc.

Hay muchas áreas en las que la MDE puede hacer valiosas aportaciones a los procesos educativos, tal como esbozamos a continuación:

- **Presentación de la información**. Se trata de aplicar técnicas de visualización de la información para destacar y presentar de manera fácil de asimilar la información que pueda servir como base para la toma de decisiones por parte de distintos tipos de actores de la comunidad educativa.

- **Retroalimentación como apoyo a los profesores**. Consiste en que el conocimiento que se va obteniendo de forma dinámica a partir de los datos recogidos durante la aplicación del proceso educativo permita a los docentes regular y mejorar el propio proceso.

- **Recomendación a los estudiantes**. Se trata de hacer recomendaciones a los estudiantes buscando optimizar su rendimiento académico. Dichas recomendaciones pueden enfocarse sobre muy diversos aspectos, como por ejemplo actividades a realizar, secuenciación de contenidos, pautas de trabajo personal, etc.

- **Adaptación a los estudiantes de los modelos educativos**. Se plantea la posibilidad de adaptar de manera automatizada y personalizada los modelos educativos según las capacidades, habilidades y conocimientos de cada estudiante.

- **Detección de alumnos particulares**. Se trata de detectar a estudiantes con un comportamiento inusual, ya sea que destaquen por aspectos negativos (comportamiento antisocial, trampas, etc.) o positivos (alumnos con capacidades destacadas).

- **Agrupamiento de estudiantes**. Se pueden identificar grupos de estudiantes afines en base a distintos criterios, lo que ofrece numerosas posibilidades, como por ejemplo aplicar distintas variantes de los procesos y técnicas educativos por grupos.

- **Análisis de redes sociales**. Este tipo de análisis permite estudiar las relaciones entre individuos. Puesto que el proceso educativo es un proceso altamente social y colaborativo en el que la comunicación entre individuos es fundamental, el conocimiento obtenido es potencialmente relevante y útil para una gran diversidad de cuestiones relativas a la docencia.

- **Estructuración de contenidos**. Se trata de ayudar a los docentes a categorizar y estructurar los conocimientos que manejan, lo cual puede ayudar mucho a preparar contenidos y actividades a los profesores, mientras que puede ayudar a los alumnos a asimilar los contenidos.

- **Desarrollo de cursos basados en la web**. Es posible aplicar patrones obtenidos mediante MDE al diseño de cursos basados en la web y en el diseño de las rutas de navegación de los mismos.

- **Planificación y programación académica**. La idea consiste en apoyarnos en el conocimiento proporcionado por la MDE para diseñar planes de estudio y materias docentes.

- **Predicción del rendimiento académico de los alumnos**. En este caso suele hacerse énfasis en conseguir una predicción temprana (cuando el curso no está muy avanzado) del rendimiento de los alumnos, de modo que aún no sea demasiado tarde para aplicar medidas correctivas para tratar de mejorar los resultados en el caso de aquellos alumnos cuyas perspectivas iniciales resulten preocupantes.

1.1 Objetivos

El **objetivo general** de esta tesis es proponer una metodología que permita recomendar de manera automática las categorías a las que pertenece un OA que se añade a un ROA. Esto permitirá facilitar y mejorar la categorización de OA añadidos a un repositorio y en consecuencia facilitará la posterior búsqueda por parte de los usuarios de OA que se adecúen a sus necesidades.

Para alcanzar el objetivo general se plantean unos **objetivos específicos**:

- **O_1**: Analizar el nivel de reducción de características que permita agilizar el tiempo de ejecución del sistema de aprendizaje sin comprometer la calidad de clasificación.
- **O_2**: Estudiar la cantidad de información histórica sobre OA similares a un OA a añadir al repositorio, ya que esta información puede permitir mejorar el rendimiento del sistema.
- **O_3**: Comparar diferentes algoritmos de aprendizaje multietiqueta en base a distintas medidas de calidad para identificar los que sean más adecuados para la categorización de OA.

1.2 Hipótesis

Nuestras hipótesis de partida para los objetivos planteados han sido:

- **H_1**: Si seleccionamos de manera adecuada el número de atributos que describen cada OA podremos reducir el tiempo de aprendizaje sin comprometer la calidad del mismo.
- **H_2**: Si a la hora de etiquetar un nuevo OA al añadirlo a un repositorio nos fijamos en otros OA similares ya existentes en el repositorio, éstos pueden sugerirnos categorías a las que posiblemente pertenezca el nuevo OA.
- **H_3**: Si comparamos los diferentes algoritmos de aprendizaje multietiqueta utilizando varias medidas de calidad, podremos determinar qué algoritmo o conjunto de algoritmos proporcionan un mejor rendimiento para el tipo de datos que nosotros manejamos.

1.3 Propuesta

En esta tesis se propone una metodología en tres fases (selección de características, utilización de datos históricos de uso y clasificación multietiqueta) que permite automatizar la categorización de OA añadidos a un repositorio. El objetico último es mejorar la caracterización de los OA almacenados para que las posteriores búsquedas que realicen los usuarios sobre el repositorio resulten lo más fáciles y fructíferas.

Los OA utilizados proceden del repositorio AGORA, del que se ha tomado un conjunto de OA recogidos a lo largo de un periodo de ocho años. Se ha usado el software MLDA para el preprocesamiento y preparación de los datos para la clasificación multietiqueta. Para lo que es propiamente la clasificación multietiqueta se ha utilizado la implementación de los algoritmos de la librería MULAN.

1.4 Estructura

La Figura 1 muestra la estructura que sigue esta tesis, que se desglosa en los apartados de introducción, marco teórico, metodología, resultados y conclusiones.

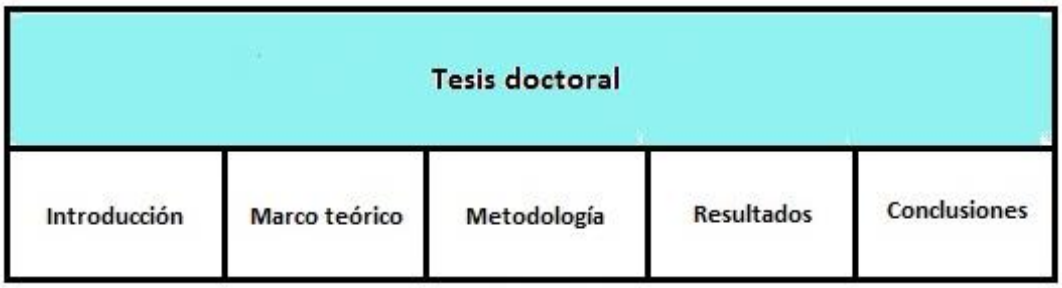

Figura 1. Esquema general de la tesis

2

MARCO TEÓRICO

En este capítulo se presentan los conceptos sobre los que se apoya nuestra investigación, las áreas de trabajo en las que se encuadra y las publicaciones afines existentes. Es necesario conocer esta base para poder entender el porqué de nuestros objetivos, así como la lógica de nuestro planteamiento y enfoque.

2.1 Contextualización

Un OA es un componente básico (una unidad dentro de un curso) o recurso digital modular que puede ser utilizado para dar apoyo al aprendizaje (Wiley, 2002). Toda entidad digital con un contenido didáctico específico y un objetivo educacional es considerada un OA. Dado que cada vez existen más OA, su búsqueda y recuperación de manera eficaz y eficiente es un asunto cada vez más importante (Zapata et al., 2015).

Se han propuesto diferentes estándares para caracterizar los OA y permitir así búsquedas complejas: SCORM, IMS-LD, LOM-ES, IEEE-LOM, etc. Sin embargo, estos estándares se basan en el uso de un elevado número de metadatos que describen a cada OA, por lo que introducir manualmente los valores correspondientes para cada metadato resulta ser muy laborioso y lento, lo que ha llevado a proponer técnicas de automatización que faciliten el poder ajustarse a estos estándares (Kannampallil y Farrell, 2005). En esta tesis pretendemos contribuir en la facilitación de esta labor permitiendo la clasificación automática de un nuevo OA ateniéndonos al estándar IEEE-LOM. Nuestro objetivo es recomendar de manera automática al usuario las posibles categorías a las que

pertenecería un nuevo OA que va a añadir a un repositorio, basando dicha recomendación en la información proporcionada acerca del OA como por ejemplo título, palabras clave y descripción. No obstante, como se explica a continuación, estamos interesados en explorar la posibilidad de mejorar la calidad de la clasificación teniendo en cuenta además información histórica relativa al uso de OA similares.

Para lograr este objetivo planteamos la aplicación de técnicas de clasificación multietiqueta para la categorización automática de los OA en áreas temáticas a partir de los términos de las características de texto puro que los caracterizan. La clasificación multietiqueta es un paradigma de clasificación en el que una instancia puede ser asignada simultáneamente a varias etiquetas, o dicho de otro modo, una instancia puede pertenecer a varias clases (como por ejemplo, una misma persona puede pertenecer a las categorías "padre", "profesor", "español"...) (Gibaja y Ventura, 2014). El enfoque multietiqueta es más adecuado que el tradicional de la clasificación monoetiqueta, ya que los OA pueden naturalmente pertenecer a varias categorías. En nuestro caso esta consideración resulta relevante particularmente cuando queremos saber en qué áreas temáticas puede ser relevante un OA, como por ejemplo un OA que sea interesante para las áreas de filosofía e informática. En esta tesis se estudia la aplicación de técnicas de clasificación multietiqueta para asociar OA a áreas temáticas en base a los metadatos textuales de los OA, pero se introduce además la novedad de considerar también información histórica acerca del uso de OA que se encuentren ya en el repositorio. La idea es que si un OA O_1 fue catalogado inicialmente como asociado a una determinada área temática A_1, pero a lo largo del tiempo ha habido una serie de docentes pertenecientes a otra área A_2 que han hecho uso de ese O_1, entonces, cuando se añada al repositorio un nuevo OA O_2 similar a O_1, puede tenerse en cuenta a la hora de asignarle automáticamente unas categorías no sólo los metadatos que caracterizan a O_2, sino también las categorías a las que pertenece O_1.

Así pues, el planteamiento básico de partida sería el de utilizar todos los metadatos proporcionados por los autores de un OA cuando lo añaden a un repositorio para clasificarlo como perteneciente a un conjunto de categorías (áreas temáticas). El ROA al que hemos recurrido en nuestro caso es AGORA (Zapata et al., 2013). A partir de este planteamiento base proponemos una serie de mejoras, la primera de las cuales consiste en hacer una selección de características. Normalmente, cuando se procesan los OA para su tratamiento automatizado, se obtiene un número de características de texto muy elevado (1336 en nuestro caso), lo cual influye de manera proporcional en el tiempo de procesamiento de los datos. Así pues, nosotros proponemos llevar a cabo un proceso de selección de los mejores atributos que permitan clasificar los OA sin que por ello se produzca una merma en la calidad del aprendizaje.

Hay que tener en cuenta que cuando un usuario añade un nuevo OA a un repositorio a veces no se molesta en indicar a qué área temática está asociado dicho OA, limitándose a incluir los datos típicos de título, descripción... Para solventar este problema se han hecho propuestas en las que se recurre a proponer la categoría a la que pertenece un OA en base a otros similares ya existentes en el repositorio y que sí estén categorizados. Siguiendo este enfoque, añadimos una mejora adicional a la de la selección de características: utilizar la información de los OA disponibles en el repositorio para tener en cuenta esta información, además de la de los metadatos, para categorizar nuevos OA. La información de uso disponible en AGORA para cada OA incluye el número de usuarios que han accedido, descargado y evaluado un OA. Así, nuestra propuesta consiste en tratar de mejorar la calidad de clasificación teniendo en cuenta no sólo la información de contenido del OA, sino también la de uso.

Finalmente, el otro aspecto que incluimos en nuestra propuesta para mejorar el planteamiento básico consiste en tratar de identificar el mejor algoritmo o conjunto de algoritmos de clasificación multietiqueta.

En los siguientes apartados de este capítulo presentamos las dos áreas relacionadas con esta tesis: los objetos de aprendizaje y la clasificación multietiqueta.

2.2 Objetos de aprendizaje

Se han propuesto diferentes definiciones y taxonomías de OA (El Saddik et al., 2001; McDonald, 2006; IEEE, 2016a; Redeker, 2003; Ihsan et al., 2006; Innis-Allen y Mugisa, 2008). Nosotros adoptamos la definición de OA propuesta en IEEE, 2016a, para poder aclarar el concepto y poder disponer de una base para nuestro trabajo, pero no queremos ser dogmáticos acerca de la definición precisa de OA. Así pues, consideramos que un OA es una entidad digital o no digital que puede ser utilizada, reutilizada o referenciada durante un proceso de aprendizaje asistido por tecnología. Una de las principales ventajas de los OA es su potencial reutilización. El concepto de OA reutilizable ha evolucionado hasta convertirse en uno de los componentes principales dentro del aprendizaje electrónico (*e-learning*) (López et al., 2012). A menudo los OA son desarrollados y puestos a disposición de cualquiera que quiera usarlos. La reusabilidad gira en torno a dos conceptos fundamentales: metadatos y repositorios. Los metadatos permiten caracterizar un OA y, por lo tanto, permiten buscar los OA más adecuados para un determinado fin educativo. Un buen esquema de metadatos permite llevar a cabo búsquedas complejas basadas en criterios diversos. Se han propuesto diferentes estándares de metadatos, a saber:

- **SCORM** (Sharable Content Object Reference Model) es un estándar propuesto por ADL (Advanced Distributed Learning) enfocado a la compartición y reusabilidad de OA (ADL, 2016).
- **IMS-LD** (IMS-Learning Design), propuesto por IMS Global Consortium, es una especificación de un lenguaje destinado a describir procesos de aprendizaje, más que OA (IMS, 2016).
- **IEEE-LOM** (IEEE-Learning Object Metadata), propuesto por la IEEE (Institute of Electrical and Electronics Engineers), es un estándar basado en XML y que hace uso de un conjunto de etiquetas para la descripción de OA (IEEE, 2016b).
- **LOM-ES** es una adaptación del estándar IEEE-LOM a la lengua española que incluye no obstante sus propias modificaciones y añadidos (Blanco et al., 2008).

Para poder realizar búsquedas de OA es necesario disponer no sólo de un lenguaje que permita caracterizar los OA (metadatos) sino que también se precisa de un componente tecnológico que permita llevar a cabo el proceso de búsqueda. Éste último es el papel desempeñado por los ROA. Un ROA es un componente software que permite el almacenamiento racional de los OA (junto con sus correspondientes metadatos) y su búsqueda. Es evidente que, al apoyarse el aprendizaje electrónico en Internet, los ROA suelen ser sistemas web, lo que permite interactuar con el repositorio (tareas de definición y manipulación) a través de la web. Algunos conocidos ROA son:

- **MERLOT** (Multimedia Educational Resource for Learning and Online Teaching), desarrollado por la California State University Center for Distributed Learning (CSU-CDL), es un ROA que almacena únicamente metadatos y que referencia a los OA almacenados en ubicaciones externas (MERLOT, 2016).
- **ARIADNE** (Alliance of Remote Instructional Authoring & Distribution Networks for Europe), desarrollado por el Programa de Telemática para la Educación y la Formación de la Comisión Europea, consiste en una red jerárquica de nodos que almacenan tanto los OA como los metadatos (ARIADNE, 2016).
- **MACE** (Metadata for Architectural Contents in Europe) es una iniciativa europea para integrar ROAs distribuidos por diferentes países con el objetivo de difundir información digital sobre arquitectura (Stefaner et al., 2007).
- **AGORA** (Ayuda para la Gestión de Objetos Reutilizables de Aprendizaje), desarrollado por la Universidad de Castilla - La mancha (España) y la Universidad de Yucatán (México), es un ROA que incluye metadatos y sus recursos asociados (Zapata et al., 2013).

En nuestra tesis hemos utilizado el estándar IEEE-LOM para los metadatos y el ROA AGORA. IEEE-LOM define una estructura jerárquica que consta de nueve categorías (general, ciclo de vida, metadatos, técnico, educacional, derechos, relación, anotación y clasificación) que contiene más de 60 metadatos en total. Estos elementos pueden contener valores de diferentes elementos y tipos. Los metadatos de los OA se almacenan usando una estructura que constituye un esquema XML, lo que permite realizar una descripción formal de la estructura de los metadatos que facilita la gestión, búsqueda y recuperación de los recursos descritos (Berners-Lee et al., 2001).

2.3 Clasificación multietiqueta

La clasificación es una de las tareas más estudiadas en las áreas de aprendizaje automática y de minería de datos (Han y Kamber, 2011; Tan et al., 2018). La clasificación consiste en predecir el valor de un atributo categórico (la clase) en base a los valores de otros atributos (atributos predictores). En un problema de clasificación se dispone de un criterio de clasificación con un conjunto fijo de clases posibles para cada instancia. Normalmente, las clases son mutuamente excluyentes, es decir, que una instancia determinada pertenece a una y sólo una clase. Por ejemplo, si se trata de clasificar animales según (criterio de clasificación) su sexo, cada uno de ellos puede ser macho o hembra, pero no ambas cosas. No obstante, hay ocasiones en las que las clases pueden solaparse, es decir, que una instancia determinada puede pertenecer a varias clases. Por ejemplo, si se trata de clasificar fotografías según su temática, una fotografía puede clasificarse como de paisaje de montaña y paisaje marino simultáneamente si recoge una imagen de una playa con montañas al fondo. Este tipo concreto de aproximación a la clasificación es lo que se conoce como clasificación multietiqueta (Gibaja y Ventura, 2014). Nosotros nos interesamos por la clasificación multietiqueta en nuestra tesis, ya que es perfectamente posible que un OA pueda pertenecer a varias áreas de interés, por ejemplo, educación y ciencia, en el supuesto caso de que tuviéramos un OA relativo a técnicas didácticas para la enseñanza de contenidos científicos. Pasamos a describir las ideas fundamentales de la clasificación multietiqueta.

Se han identificado dos enfoques básicos para abordar problemas de clasificación multietiqueta, métodos de transformación del problema y métodos de adaptación del algoritmo (Gibaja y Ventura, 2014).

Los métodos de transformación del problema se fundamentan en la idea de transformar el conjunto de datos multietiqueta en uno o varios conjuntos de datos

monoetiqueta de tal modo que sea posible usar cualquier técnica de clasificación monoetiqueta tradicional. Dentro de este grupo hay distintos algoritmos:

- El algoritmo de **relevancia binaria** (Binary Relevance - BR) (Gibaja y Ventura, 2014) descompone un problema multietiqueta en un problema binario independiente por cada etiqueta y entonces se genera un clasificador estándar para cada uno de estos conjuntos de datos. Este es un método muy popular a causa de su eficiencia y simplicidad. No obstante, una de las principales críticas que se le hacen es su incapacidad para tener en cuenta las relaciones entre etiquetas. Esto ha llevado a proponer diferentes métodos que solventan este inconveniente.

- Uno de estos métodos es las **cadenas de clasificadores** (Classifier Chains - CC) (Read et al., 2011), que genera una cadena de clasificadores binarios, uno para cada etiqueta, de modo que el espacio de características de cada conjunto de datos es extendido con las etiquetas de los conjuntos de datos precedentes dentro de la cadena.

- El orden en que se forme la cadena puede afectar al rendimiento del clasificador, lo que ha llevado a su vez a la propuesta de una mejora del método anterior, aplicando un esquema de embolsamiento (bagging) y usando una cadena diferente para cada clasificador base, dando lugar al enfoque conocido como **conjunto de cadenas de clasificadores** (Ensemble of Classifier Chains - ECC) (Read et al., 2011).

- El **apilamiento multietiqueta** (Multi-Label Stacking - MLS) (Tsoumakas et al., 2009) es otro algoritmo basado en BR, consistente en aplicar BR dos veces. Primero hace un entrenamiento básico consistente en crear un clasificador binario por cada etiqueta. Luego se aprende un metanivel de clasificadores binarios siguiendo un enfoque de apilamiento (stacking) (Wolpert, 1992) que permite combinar las predicciones de nivel básico.

- Otro algoritmo basado en BR es la **jerarquía de clasificadores multietiqueta** (Hierarchy Of Multi-label classifiERs - HOMER) (Tsoumakas et al., 2008). En este caso se trata de un algoritmo diseñado especialmente para dominios en los que el número de etiquetas es alto. Transforma un problema de clasificación multietiqueta en una jerarquía en forma de árbol de problemas multietiqueta más simples que contienen cada uno de ellos un menor número de etiquetas.

- El algoritmo de los **conjuntos potenciales de etiquetas** (Label Powerset - LP) (Boutell et al., 2004) transforma un conjunto de datos multietiqueta en un conjunto de datos monoetiqueta de tal manera que cada combinación de etiquetas del conjunto original es considerada como una nueva clase, y entonces se aplica

cualquier algoritmo de clasificación monoetiqueta. La complejidad de este algoritmo es exponencial con respecto al número de etiquetas y no es capaz de predecir combinaciones de etiquetas que no aparezcan en el conjunto de datos original. Se han propuesto algunos enfoques para solucionar estos problemas.

- Uno de estos algoritmos es el de los **conjuntos podados** (Pruned Sets - PS) (Read et al., 2008), que lleva a cabo un proceso de poda para centrarse en los conjuntos de etiquetas más frecuentes y entonces aplica LP. No obstante, PS sigue teniendo el problema de no ser capaz de predecir conjuntos de etiquetas que no aparezcan en el conjunto de datos original.

- Para solucionar este último problema se ha propuesto el algoritmo del **grupo de conjuntos podados** (Ensemble of Pruned Sets - EPS) (Read et al., 2008).

- Otro método basado en LP es el de los **k conjuntos de etiquetas aleatorios** (Random-k-LabelSets - RAkEL) (Tsoumakas et al., 2011a), que descompone de manera aleatoria el conjunto de etiquetas original en varios conjuntos de pequeño tamaño para luego aplicar LP a cada uno de ellos. Las salidas son combinadas para formar una predicción multietiqueta mediante voto mayoritario.

- La **ordenación calibrada de etiquetas** (Calibrated Label Ranking - CLR) (Fürnkranz et al., 2008) genera un conjunto binario para cada par de etiquetas. Cada conjunto de datos contiene aquellas instancias pertenecientes a alguna de las dos etiquetas, pero no a ambas. Además, por cada etiqueta se crea un conjunto de datos binario adicional. Esto permite considerar una etiqueta virtual que actúa como punto de división para las etiquetas relevantes.

Los métodos de adaptación del algoritmo modifican un algoritmo de clasificación monoetiqueta de manera que sea capaz de tratar con problemas de clasificación multietiqueta sin tener que aplicar ninguna transformación previa a los datos.

- El método de los **k vecinos más cercanos multietiqueta** (Multi-Label k-Nearest Neighbours - MLkNN) (Zhang y Zhou, 2007) adapta el célebre algoritmo de los k vecinos más cercanos al terreno multietiqueta. Cuenta los vecinos cercanos que pertenecen a cada etiqueta y, para una nueva instancia a clasificar, se usa el principio del máximo a posteriori (maximum a posteriori principle - MAP) para determinar el conjunto de etiquetas asociadas.

- La **regresión logística basada en instancias** (Instance-based Logistic Regression - IBLR) (Cheng y Hüllermeier, 2009) combina el aprendizaje basado en instancias y la regresión logística usando las etiquetas de los vecinos como atributos complementarios en un esquema de regresión logística.

- Otro método es **AdaBoost.MH** (Schapire y Singer, 2000), que adapta el método AdaBoost (Freund y Schapire, 1995), un algoritmo que genera de manera iterativa un conjunto de clasificadores de modo que cada clasificador se centra en las instancias que han sido más difíciles de clasificar para los clasificadores anteriores, lo cual se indica asignando pesos a las instancias. Adaboost.MH funciona de manera similar pero asignando pesos tanto a las instancias como a las etiquetas.

Un asunto importante a tener en cuenta es la evaluación del rendimiento de los algoritmos de clasificación multietiqueta, para lo que es necesario disponer de métricas de evaluación que permitan llevar a cabo comparaciones. Las medidas de evaluación multietiqueta pueden dividirse en dos grupos: basadas en ejemplos y basadas en etiquetas.

- Las métricas basadas en ejemplos se calculan para cada instancia y luego esos valores son promediados. Dentro de este grupo se incluyen las métricas para evaluar biparticiones (Hamming loss, subset accuracy, precision, recall, f–measure y accuracy) y las métricas para evaluar ordenaciones (rankings) (average precision, coverage, one-error y ranking-loss)[1].

- Las métricas basadas en etiquetas tienen en cuenta el número de positivos verdaderos (true positives - tp), negativos verdaderos (true negatives - tn), positivos falsos (false positives - fp) y negativos falsos (false negatives - fn). Se calcula una tabla de contingencia para cada etiqueta y a partir de esto pueden aplicarse dos estrategias diferentes para promediar los valores de la métrica. El enfoque macro-promedio (macro-average) calcula primero la métrica para cada etiqueta y luego promedia estos valores. El enfoque micro-promedio (micro-average) agrega primero los valores de las tablas de contingencia en una única tabla a partir de la que se calcula entonces la métrica. De esta forma pueden calcularse versiones tanto macro-promedio como micro-promedio de precision, recall y f-measure.

Las definiciones de todas estas métricas pueden encontrarse en Gibaja y Ventura, 2015.

[1] En el caso de los nombres de las métricas se ha optado por no traducirlas al español, dada la confusión que crearía el traducir los términos *precision* y *accuracy*, que se pueden traducir ambos de igual manera en español (exactitud, precisión...).

2.4 Aplicación de clasificación multietiqueta a objetos de aprendizaje

La aplicación de técnicas de clasificación multietiqueta a la categorización de OA ha sido escasamente explorada en la bibliografía especializada. En López et al., 2012 los autores aplican únicamente dos algoritmos de clasificación multietiqueta (RAkEL y MLkNN) a dos conjuntos de datos, uno de los cuales consta de 253 OA y otro con 1000 OA. Se aplica un enfoque de clasificación multietiqueta con el objetivo de hallar las etiquetas correspondientes con los temas tratados por los OA. Los dos algoritmos son comparados en base a seis medidas de rendimiento, llegando a la conclusión de que RAkEL tiende a ofrecer mejores resultados que MLkNN.

En Aldrees y Chikh, 2016 los autores comparan cuatro algoritmos de clasificación multietiqueta (ECC, RAKEL, EPS y MLkNN) en base a 16 métricas de rendimiento. Utilizan un conjunto de 658 OA procedentes del repositorio ARIADNE. Su objetivo es encontrar el mejor algoritmo de clasificación multietiqueta para la categorización de OA. Sus resultados indican que ECC es superior al resto de algoritmos.

Estos dos trabajos aplican algoritmos de clasificación multietiqueta a la categorización de OA, pero en ambos casos se considera un número reducido de algoritmos de aprendizaje. En nuestra tesis tenemos en cuenta un mayor número de algoritmos de clasificación multietiqueta (13) y además aplicamos pruebas estadísticas para comprobar si las diferencias de rendimiento son estadísticamente significativas. Además, aportamos una contribución novedosa: analizamos la posibilidad de mejorar el rendimiento de la clasificación teniendo en cuenta información de uso de los OA. La Tabla 1 recoge los datos fundamentales que permiten comparar nuestra tesis con los trabajos previos citados.

Tabla 1. Comparación de esta tesis con trabajos previos

Referencia	Repositorio	#OA	#Atributos	#Etiquetas	#Algoritmos	#Métricas	Información usada
López et al., 2012	Repositorio privado	253	1442	38	2	6	Atributos IEEE-LOM
	LORNet y MERLOT	1000	1442	-			
Aldrees y Chikh, 2016	ARIADNE	658	6166	30	4	16	Atributos IEEE-LOM
Esta tesis	AGORA	519	1336	5	13	16	Atributos IEEE-LOM y datos de uso

3

METODOLOGÍA

La metodología seguida en esta tesis doctoral consta de dos fases principales. La primera de estas fases está concebida para ser ejecutada fuera de línea. El primer paso consiste en crear un fichero de datos con el formato adecuado a partir de los metadatos de los OA. Se extraen de la base de datos del ROA las características de texto que caracterizan a los OA, almacenándolos con el formato requerido por las herramientas software que vamos a utilizar a continuación. Después se lleva a cabo un preprocesamiento de datos, que a su vez contempla dos aspectos. En primer lugar se hace una selección de atributos, ya que el número de atributos que describe a los OA en base a su contenido tiende a ser muy elevado (1336 en nuestro caso), por lo que valdrá la pena llevar a cabo este preprocesamiento, para poder reducir el tiempo de aprendizaje sin que disminuya la calidad del mismo seleccionando el subconjunto más idóneo de atributos. Para intentar mejorar los resultados del sistema se lleva a cabo una segunda tarea de preprocesamiento consistente en añadir al conjunto de datos atributos relativos al pasado uso de OA existentes en el ROA que sean similares a los nuevos OA a añadir. La fase fuera de línea concluye con un paso de en el que se comparan varios algoritmos de aprendizaje multietiqueta en base a un conjunto de métricas de calidad para determinar cuál es el algoritmo o conjunto de algoritmos que mejores resultados ofrecen. El objetivo de esta primera fase fuera de línea es pues poder estudiar la influencia de tres factores en la calidad de la predicción del sistema, a saber, el número de atributos de contenido en los que el sistema se basa para llevar a cabo la clasificación, la cantidad de información histórica de uso de OA similares y el algoritmo de clasificación multietiqueta a aplicar.

Una vez estudiada la influencia de cada uno de estos tres factores, es posible elegir de entre los valores más favorables para cada uno de ellos para configurar el sistema de recomendación, que se ejecutaría en una segunda fase en línea cuando se añade un nuevo OA al repositorio. Así, cuando un usuario añade un nuevo OA al repositorio, nuestro sistema puede recomendar de manera automática algunas disciplinas a las que el OA pertenecería.

Esta metodología se ilustra en la Figura 2. En las siguientes secciones se describe cada uno de los pasos que la componen.

Figura 2. Propuesta para sistema de recomendación

3.1 Recogida y preprocesamiento de datos

Los datos usados en esta tesis han sido obtenidos del repositorio AGORA, correspondiendo a los OA recogidos en el mismo entre 2009 y 2016 (Zapata et al., 2013). Cuando un usuario añade un nuevo OA al repositorio AGORA debe incluir una serie de metadatos como por ejemplo título, palabras reservadas, descripción... Con este propósito se utiliza un formulario (ver Figura 3). Aquí, el usuario puede indicar las áreas temáticas (una o varias) a las que pertenece el OA, de entre cinco posibles valores preestablecidos:

- Ciencias de la Salud
- Ciencias Naturales y Exactas
- Ciencias Sociales y Administrativas
- Ingeniería y Tecnología
- Educación, Humanidades y Arte

Figura 3. Formulario AGORA para introducir metadatos de un OA

Se ha utilizado el software MLDA (Moyano et al. 2017) para preparar los datos con vistas a los sucesivos pasos a ejecutar sobre los mismos, dándoles un formato adecuado para el enfoque multietiqueta. En nuestro trabajo hemos empleado 519 OA extraídos del repositorio AGORA. A partir del título, palabras reservadas y descripción de todos estos OA se han extraído 1336 términos (atributos de contenido), después de eliminar las palabras vacías (*stop words*) y de llevar a cabo un proceso para reducir los términos a sus raíces (*stemming*). A continuación, se calcula la frecuencia de dichas raíces para cada OA, obteniendo así su representación de la frecuencia de término (*term frequency - TF*) (Ochoa

Y Duval, 2008). De aquí se obtiene una matriz instancia/término en la que cada elemento indica cuántas veces aparece un término en una instancia determinada. Posteriormente se normaliza la frecuencia de término, que indicará la importancia de cada término, obteniendo así en nuestro caso un conjunto de 1336 atributos referentes a la información de contenido de los OA.

A continuación hemos añadido información histórica acerca del uso que se ha hecho en el pasado de los OA. Esta información está separada por años, desde 2009 hasta 2016. La información añadida recoge el número de descargas, número de visualizaciones y número de evaluaciones para cada una de las cinco disciplinas académicas (áreas temáticas). De este modo, tenemos 15 atributos de uso para cada OA. Finalmente, hemos añadido a cada OA cinco etiquetas de clase en formato binario como posibles clases a predecir (ver Tabla 2). En la Figura 4 se presenta un ejemplo de fichero de datos.

Tabla 2. Descripción de atributos

Número de atributo	**Descripción**
1 a 1336	Información de contenido acerca de los términos que aparecen en el OA (número real entre 0 y 1)
1337 a 1351	Información de uso (número entero indicando número de descargas, visualizaciones y evaluaciones dentro de cada una de las cinco disciplinas)
1352 a 1356	Etiqueta acerca de la clase (valor binario 0/1 indicando si el OA pertenece o no a una determinada área)

Contenido	**Uso**	**Etiquetas**
0,0,0.21,0,0,0,0,0,0.16,0,0,0,0.05, …	… ,0,0.44,	2,1,0,0,0,1,0,0,0,0,2,0,7,3,3,	1,0,0,1,1
0,0,0.03,0,0,0.25,0,0,0,0,0,0.33, …	…, 0.14,0,	0,7,0,0,5,0,0,0,0,0,0,0,1,6,0,	1,0,0,0,1
0,0,0,0,0,0,0,0,0,0,0,0,0.10,0.06, …	…,0,0.66,	1,1,1,0,4,0,0,0,0,0,3,0,3,4,1,	1,0,0,1,1
…	…	…	…
0,0,0.07,0,0,0,0,0,0,0,0,0.55,0, …	…,0.03,0,	1,6,0,0,0,0,0,0,0,0,0,0,2,0,0,	1,1,1,0,0

Figura 4. Ejemplo de fichero de datos

3.2 Selección de características

El primer paso en el preprocesamiento fuera de línea de los datos tiene como objetivo entender el efecto que tiene la reducción del número de características predictivas en el rendimiento de los algoritmos de clasificación multietiqueta. En este primer paso hemos decidido tener en cuenta únicamente un factor: el número de atributos de contenido. Por este motivo, en los experimentos correspondientes se ha empleado un conjunto de datos en el que se incluyen los atributos de contenido (originalmente, 1336) junto con las etiquetas de clase, pero no los atributos de uso. Por regla general, el tiempo empleado por un algoritmo de clasificación multietiqueta para generar un modelo es directamente proporcional al número de instancias de entrenamiento y al número de atributos que describen cada instancia. Nuestra hipótesis es que el tiempo de ejecución se reducirá si reducimos el número de atributos empleados. No obstante, al reducir el número de atributos podríamos estar descartando información relevante, lo que llevaría a la obtención de un modelo poco fiable. Es por esto por lo que hemos aplicado un método de selección de características a distintos niveles de reducción, para después poder comparar el rendimiento de los clasificadores usando distinto número de características y así seleccionar el máximo nivel de reducción de características que no llegue a incurrir en una disminución del rendimiento del clasificador. Hay que tener en cuenta que al reducir el número de atributos, no sólo disminuirá el tiempo de ejecución del algoritmo de clasificación, sino que en algunos casos es posible que el rendimiento del clasificador sea incluso mejor que en el caso de usar todos los atributos, ya que entre éstos puede haber atributos irrelevantes y ruidosos que afecten negativamente a la capacidad predictiva de los modelos obtenidos.

La selección de características se ha llevado a cabo tal como se propone en Tsoumakas et al., 2011a. Primero se aplica para cada etiqueta el método de ordenación de atributos $\chi 2$. Después se estima para cada etiqueta el valor que aporta cada atributo calculando el valor estadístico $\chi 2$ con respecto a la etiqueta, lo que nos da una indicación de su independencia. La razón para hacer esto es que si un atributo es independiente de una clase, ese atributo puede eliminarse. Así se obtiene para cada etiqueta y cada atributo una puntuación basada en el máximo valor del estadístico $\chi 2$ sobre el conjunto de todas las etiquetas. A partir de estas puntuaciones, se elegirán aquellos *n* atributos con mejor puntuación, siendo *n* un valor indicado por el usuario.

3.3 Información histórica de uso

Los 519 OA empleados en esta tesis fueron recogidos durante el año 2008 en el repositorio AGORA. A lo largo de los sucesivos años dichos OA han sido utilizados por una comunidad diversa de docentes. El repositorio AGORA recoge información relativa al uso que se hace de los OA (número de descargas, número de visualizaciones y número de evaluaciones). Puede disponerse de estos datos como totales o bien agrupados por área de conocimiento. Se ha detectado que en ocasiones hay docentes que han usado OA asignados inicialmente (según sus etiquetas) a determinadas áreas perteneciendo ellos a áreas diferentes. En nuestro trabajo recurrimos a la información de uso de los OA agrupada por las diferentes áreas de conocimiento. Nuestro objetivo es averiguar si esta información acerca del uso que se da en el mundo real a los OA puede ayudar a mejorar el rendimiento de los clasificadores. Para ello, cuando se añade un nuevo OA al repositorio, tenemos en cuenta la información de uso de aquellos OA ya presentes en el repositorio que más se parecen al nuevo OA. Para poder identificar aquellos OA que más se parecen a uno dado hemos utilizado una métrica de similitud que se propuso originalmente para mejorar los sistemas de gestión de OA (Menéndez et al., 2011). Esta métrica mide la similitud entre OA en base a la información representada en sus metadatos. Hemos elegido esta métrica porque se ajusta al estándar IEEE-LOM y ha sido probada de forma satisfactoria con los OA del repositorio AGORA (Menéndez et al., 2011).

Para entender si la inclusión de datos históricos de uso puede llegar a mejorar el rendimiento de clasificación se han utilizado conjuntos de datos en los que se incluyen atributos de contenido, atributos de uso y etiquetas de clase, creando distintos conjuntos de datos que se diferencian únicamente en la información acumulada acerca del uso histórico de los OA, incorporando información acumulada a lo largo de periodos de tiempo de diferente duración por años. El resto de información (atributos de contenido y etiquetas de clase) son exactamente los mismos en todos los conjuntos de datos. Los atributos de contenido utilizados son siempre el subconjunto que ha sido identificado como el ideal mediante el preprocesamiento de selección de características. Así, hay un conjunto de datos sin información de uso alguna; un segundo conjunto de datos añade al anterior los valores de los atributos de uso correspondientes al año 2009; en un tercer conjunto de datos, los valores de los atributos de uso serán los que recogen la información del uso que se ha hecho de los OA a lo largo de los años 2009 y 2010… De esta manera componemos una serie de conjuntos de datos en los que en cada uno de ellos se va incorporando más información de uso por años, hasta el último conjunto de datos en el que los valores de los atributos de uso acumulan toda la información del uso que se ha hecho de los OA entre 2009 y 2016.

3.4 Algoritmo de clasificación multietiqueta

En el último paso de la fase fuera de línea se trata de encontrar el algoritmo o conjunto de algoritmos que mejor rendimiento ofrecen para el conjunto de datos que estamos utilizando. Se compondrá entonces un conjunto de datos confeccionado según los resultados arrojados por los dos anteriores pasos de preprocesamiento, es decir, que se usará el subconjunto de atributos identificado como óptimo y el intervalo cronológico por años que se haya revelado como más enriquecedor en el estudio de la mejora del rendimiento de clasificación mediante la inclusión de datos históricos de uso. Con este objetivo se han aplicado 13 algoritmos de clasificación multietiqueta al mismo conjunto de datos y los resultados han sido evaluados según 16 medidas de evaluación. Respecto a los algoritmos, se han usado diez métodos de transformación del problema y tres métodos de adaptación de algoritmo. Los métodos de transformación del problema son: relevancia binaria, cadena de clasificadores, conjunto de cadenas de clasificadores, apilamiento multietiqueta, jerarquía de clasificadores multietiqueta, conjuntos potenciales de etiquetas, conjuntos podados, grupo de conjuntos podados, k conjuntos de etiquetas aleatorios y ordenación calibrada de etiquetas. Los métodos de adaptación de algoritmo son: k vecinos más cercanos multietiqueta, regresión logística basada en instancias y AdaBoost.MH. Las medidas de evaluación utilizadas son 16, diez basadas en ejemplos (average precision, coverage, example-based accuracy, example-based f-measure, example-based precision, example-based recall, Hamming loss, one error, ranking loss y subset accuracy) y seis basadas en etiquetas (macro-averaged f-measure, macro-averaged precision, macro-averaged recall, micro-averaged f-measure, micro-averaged precision, micro-averaged recall).

3.5 Fase de recomendación en línea

Disponiendo ya de un sistema ajustado según los resultados obtenidos en la fase fuera de línea, cuando el usuario añade un nuevo OA al repositorio es posible recomendar automáticamente un conjunto de categorías a las que dicho OA pertenecería. Nuestro sistema es capaz de categorizar un OA no etiquetado. Imaginemos por ejemplo que un usuario añade al repositorio un nuevo OA, indicando que pertenece al ámbito de ciencias de la salud. Si en el repositorio hay OA similares que en el pasado han sido utilizados por usuarios pertenecientes al área de ciencias sociales, sería útil recomendar añadir al nuevo OA en la categoría de ciencias sociales, además de en la que propuso originalmente el

usuario que lo introdujo, es decir, ciencias de la salud. Esto podría mejorar los resultados de las búsquedas que se lleven a cabo sobre el repositorio con posterioridad.

4

RESULTADOS

Hemos llevado a cabo una serie de experimentos para comprobar las hipótesis H_1, H_2 y H_3 planteadas en la 1.2 Hipótesis, que se corresponden con los tres pasos de la fase de ejecución fuera de línea descritas en el capítulo 3. Tal como se explica en el mencionado capítulo, todos los experimentos implican la ejecución de 13 algoritmos de clasificación multietiqueta y la evaluación de los resultados ofrecidos en base a 16 medidas de evaluación, ya sea para averiguar cuál es el subconjunto de características óptimo a usar sin comprometer la calidad de la clasificación, ya sea identificar la cantidad ideal de información histórica de uso para aumentar la capacidad predictiva del sistema o ya sea para hallar los mejores algoritmos de clasificación multietiqueta. Se ha utilizado la librería MULAN (Tsoumakas et al. 2011) para la ejecución de los algoritmos de clasificación multietiqueta. En la Tabla 3 se indican los 13 algoritmos indicando su nombre en español, su nombre en inglés y las siglas que corresponden a sus nombres en inglés, que son por las que suelen ser más conocidos y las que hemos empleado en las tablas en las que más adelante recogemos los resultados de los experimentos.

Los algoritmos han sido configurados tal como se indica a continuación. ABMH utiliza el tope de decisión (*decision stump*) como algoritmo de aprendizaje base. BR, LP, CC, PS y CLR usan la implementación J48 del algoritmo C4.5 de árboles de decisión como algoritmo base. HOMER se ha ejecutado con BR y J48 como algoritmo base utilizando tres grupos. RAkEL se ha ejecutado usando LP con J48 como algoritmo base y un tamaño de subconjunto de 3, un número de modelos que es el doble del número de etiquetas y 0.5 como umbral. MLkNN se ha ejecutado con 10 vecinos y un factor de amortiguación de 1.0. MLS se ha ejecutado con J48 como algoritmo base. ECC se ha ejecutado con J48 como algoritmo base, usando 10 modelos, confianza y muestreo con reemplazo. EPS se ha

ejecutado con 10 modelos por conjunto y la llamada estrategia A (manteniendo los 2 subconjuntos con mejor puntuación), el 66% de los datos como muestra, J48 como clasificador base, un umbral de 0.5 y podando los conjuntos de etiquetas que se dan menos de 3 veces.

Tabla 3. Algoritmos de clasificación multietiqueta

Nombre español	**Nombre inglés**	**Siglas**
relevancia binaria	binary relevance	BR
cadenas de clasificadores	classifier chains	CC
conjunto de cadenas de clasificadores	ensemble of classifier chains	ECC
apilamiento multietiqueta	multi-label stacking	MLS
jerarquía de clasificadores multietiqueta	hierarchy of multi-label classifiers	HOMER
conjuntos potenciales de etiquetas	label powerset	LP
conjuntos podados	pruned sets	PS
grupo de conjuntos podados	ensemble of pruned sets	EPS
k conjuntos de etiquetas aleatorios	random-k-labelsets	RAkEL
ordenación calibrada de etiquetas	calibrated label ranking	CLR
k vecinos más cercanos multietiqueta	multi-label k-nearest neighbours	MLkNN
regresión logística basada en instancias	instance-based logistic regression	IBLR
	adaboost.mh	ABMH

En las siguientes secciones describimos los experimentos realizados y explicamos los resultados obtenidos.

4.1 Reducción del número de atributos de contenido

Para llevar a cabo el primer experimento hemos partido del conjunto de datos original obtenido a partir del repositorio AGORA que contiene atributos de contenido de los OA obtenidos a partir de los metadatos que los describen, junto con las correspondientes etiquetas de clase. Si reducimos el número de atributos predictivos, disminuirá el tiempo que tardan los algoritmos de clasificación multietiqueta en generar sus modelos. Nos planteamos hacer un estudio que nos permita analizar hasta qué punto podemos reducir la cantidad de atributos utilizados sin llegar a incurrir en una pérdida de calidad de los resultados de clasificación. Hay que tener en cuenta además que en algunos casos la selección de características puede incluso mejorar la calidad de clasificación, ya que pueden elminarse atributos ruidosos o irrelevantes que estarían perjudicando la capacidad predictiva de los modelos obtenidos.

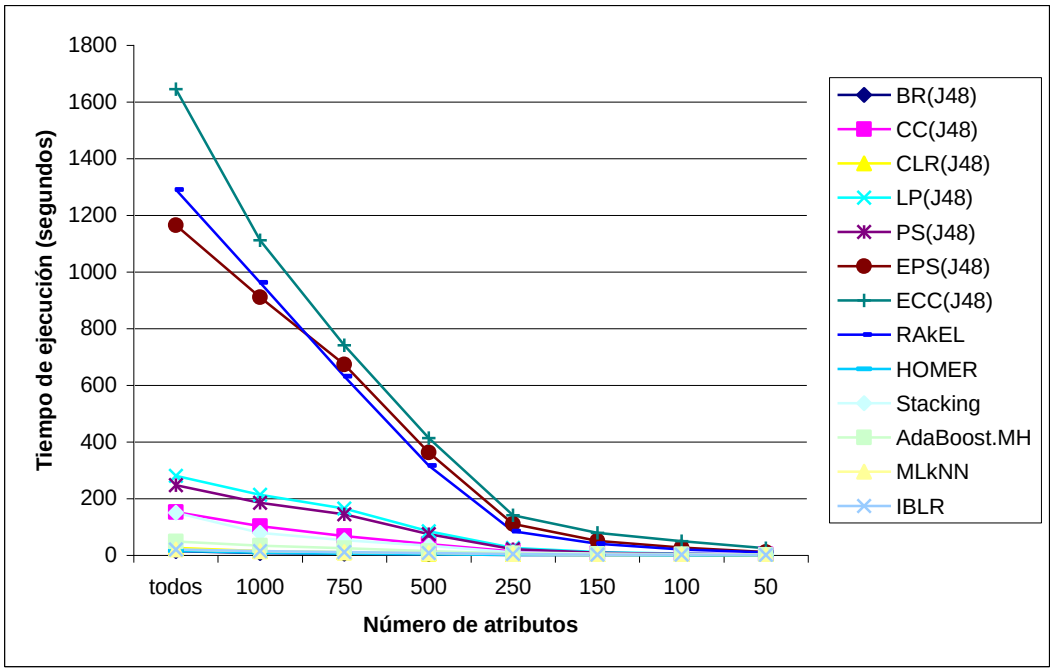

Figura 5. Tiempo de ejecución respecto a número de atributos

El conjunto de datos original contiene 519 instancias, descritas cada una de ellas por 1336 atributos de contenido. A partir de este conjunto de datos original hemos seleccionado conjuntos con 1000, 750, 500, 250, 100 y 50 atributos, quedándonos siempre con aquel subconjunto de atributos que presentan una mayor puntuación según el criterio

descrito en la sección 3.2 Selección de características. Una vez obtenidos estos subconjuntos se han aplicado a cada uno de ellos 13 algoritmos de clasificación multietiqueta y los resultados han sido evaluados empleando 16 métricas de calidad (ver sección 4.3 Selección del mejor algoritmo de clasificación multietiqueta). Se ha utilizado validación cruzada con 10 particiones y 10 semillas.

Los resultados obtenidos indican que el tiempo de ejecución se reduce significativamente al ir reduciendo el número de atributos (ver Figura 5).

Tabla 4. Hamming loss según nivel de reducción

Alg./Num. atr.	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
AdaBoost.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

En lo que se refiere a la calidad de clasificación, se aprecian diferencias según se usen más o menos atributos. A modo de ejemplo del efecto del número de atributos sobre el rendimiento de clasificación, la Tabla 4 muestra los resultados obtenidos para una de las métricas, Hamming loss, que indica una mejor calidad de clasificación cuanto más bajo es su valor. Puede apreciarse que no siempre los mejores resultados corresponden a la utilización de todos los atributos.

Para comparar el rendimiento de los algoritmos de clasificación según el nivel de reducción de atributos se ha realizado un test de Friedman[2]. Se trata de una prueba no paramétrica que compara las puntuaciones medias de los niveles de reducción, donde el nivel de reducción con mejor valor para una determinada métrica para un determinado algoritmo recibe una puntuación de 1, el nivel de reducción con el siguiente mejor valor para la misma métrica y el mismo algoritmo recibe una puntuación de 2 y así sucesivamente. A partir de estas puntuaciones se calculan unas puntuaciones medias para cada nivel de reducción, lo que nos permite saber qué niveles de reducción son los que ofrecen mejores resultados teniendo en cuenta todos los algoritmos. Así, el nivel de reducción con valor más cercano a 1 será el mejor nivel de reducción en general para todos los algoritmos.

Tabla 5. Test de Friedman y Bonferroni-Dunn para Hamming loss según nivel de reducción

p-valor Friedman = 2,000E-6		Bonferroni-Dunn post test	
Nivel reducción	Puntuación (orden)	p-valor	Hipótesis nula
50	6,808 (8)	3,690E-04	**Rechazada**
100	6,500 (7)	1,378E-03	**Rechazada**
150	5,731 (6)	2,432E-02	**Rechazada**
250	4,192 (5)	1,305	Aceptada
1000	3,769 (4)	2,649	Aceptada
750	3,115 (3)	5,889	Aceptada
500	2,962 (2)	6,776	Aceptada
1336 (control)	2,923 (1)		

A modo de ejemplo, la Tabla 5 muestra los resultados del test de Friedman para la Hamming loss. El p-valor (≤ 0,05) indica diferencias significativas entre los niveles de reducción con un elevado nivel de confianza (95%). Para saber qué niveles de reducción presentan diferencias significativas se ha ejecutado un test a posteriori de Bonferroni-

[2] Todas las pruebas estadísticas que se mencionan a partir de este punto han sido realizadas con el software KEEL (Alcalá et al., 2009).

Dunn, cuyos resultados se muestran también en la Tabla 5. El nivel de reducción de control es el que tiene un mejor valor (en este caso, el que corresponde al uso de todos los atributos). El p-valor obtenido indica que hay diferencias significativas entre los niveles de reducción de 50, 100 y 150 atributos y el resto de niveles de reducción con un nivel de confianza del 95%. Por lo tanto, el punto de corte indicado por el test de Bonferroni-Dunn para Hamming loss es el de 250 atributos, ya que las reducciones a 50, 100 y 150 atributos sí son significativamente diferentes (peores) que el nivel de control (que es el mejor). El resto de niveles de reducción no tienen diferencia significativa respecto al valor de control y de entre todos ellos nos interesaría quedarnos obviamente con el de 250, ya que es el valor más bajo de este subconjunto y el que implicaría un menor tiempo de ejecución sin perjuicio de la calidad de clasificación. La Figura 6 ofrece una representación gráfica de las diferencias relativas correspondientes al test de Bonferroni-Dunn de la Tabla 5.

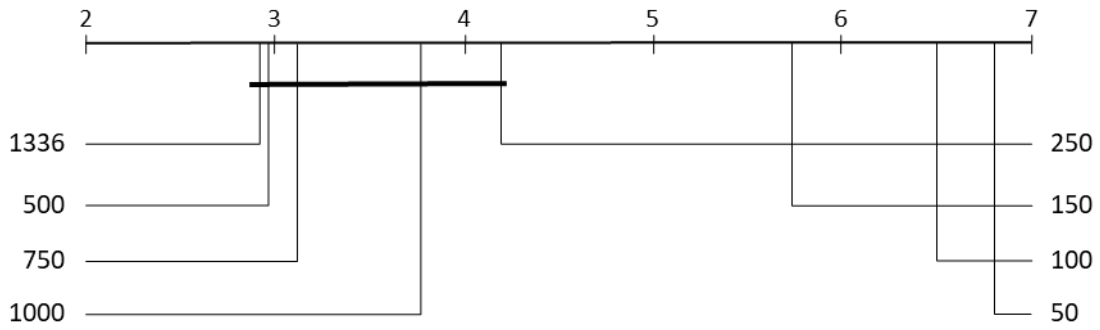

Figura 6. Diagrama de distancia crítica correspondiente al test de Bonferroni-Dunn para Hamming loss según nivel de reducción

El procedimiento descrito a modo de ejemplo para la métrica Hamming loss se ha aplicado de la misma manera a todas las métricas, y los resultados se muestran en la Tabla 6. Cada fila de la tabla contiene valores de puntuación. En la tabla se indica con el símbolo '↑' aquellas métricas que son a maximizar y con el símbolo '↓' aquellas que son a minimizar. Puede observarse que hay métricas en las que la mejor puntuación se obtiene usando menos del máximo (1336) de atributos. Average precision, coverage, ranking loss y subset accuracy obtiene la mejor puntuación usando 1000 atributos. Example-based precision, micro-averaged precision y one error obtienen la mejor puntuación usando 750 atributos. Example-based f-measure, example-based recall y micro-averaged obtienen la mejor puntuación usando 500 atributos. Example-based f-measure, example-based recall y micro-averaged recall obtienen la mejor puntuación usando 250 atributos. En la última fila

de la Tabla 6 se indican la puntuación media para cada nivel de reducción. Aquí también se aprecia que la mejor puntuación no corresponde a 1336 atributos sino a 1000.

Tabla 6. Puntuaciones medias y puntos de corte

Métrica/Num. atr.	1336	1000	750	500	250	150	100	50	Corte
↑Average precision	4,192	3,115	3,423	3,769	3,885	5,462	6,231	5,923	150
↓Coverage	4,500	2,192	3,423	4,115	4,115	5,654	6,385	5,615	250
↑E-based acc	3,077	3,231	3,577	3,346	3,962	5,577	6,500	6,731	150
↑E-based f-meas	4,769	4,385	3,654	3,346	3,346	4,808	5,692	6,000	100
↑E-based prec	3,692	4,385	2,885	3,269	4,192	5,885	5,846	5,846	250
↑E-based recall	5,077	4,000	4,192	3,577	3,577	4,269	5,846	5,462	ND
↓Hamming loss	2,923	3,769	3,115	2,962	4,192	5,731	6,500	6,808	250
↑Macro-avg f-meas	2,000	3,000	3,500	3,962	4,346	5,115	6,962	7,115	250
↑Macro-avg prec	2,308	2,692	3,577	3,962	4,731	5,731	6,231	6,769	250
↑Macro-avg recall	2,462	3,154	3,885	4,346	4,192	4,808	6,539	6,615	150
↑Micro-avg f-meas	3,731	3,615	3,731	3,577	3,769	4,731	6,346	6,500	150
↑Micro-avg prec	3,923	4,539	3,115	3,654	4,039	5,423	5,500	5,808	100
↑Micro-avg recall	4,692	4,000	4,231	4,039	3,615	3,962	5,846	5,615	ND
↓One error	3,692	3,615	3,577	4,308	3,654	6,192	5,808	5,154	250
↓Ranking loss	4,500	3,115	3,885	4,423	3,962	4,885	5,962	5,269	ND
↑Subset accuracy	2,692	2,231	3,269	3,577	4,962	5,731	6,539	7,000	250
Puntuación media	3,639	3,440	3,565	3,765	4,034	5,248	6,171	6,139	-

Los p-valores (≤ 0,05) del test de Friedman indican que hay diferencias significativas entre los niveles de reducción con un alto nivel de confianza (95%) excepto para example-based recall, micro-averaged recall y ranking loss. Esto significa que para estas tres métricas no hay diferencia significativa (ND) entre usar todos los atributos o algún subconjunto de los mismos. Para el resto de métricas sí que se detectan diferencias estadísticamente significativas, casos en los que indicamos el punto de corte según los test Bonferroni-Dunn realizados (ver última columna de la Tabla 6). El punto de corte indica que, para la métrica en cuestión, no hay diferencia significativa de rendimiento si usamos, al menos, el número indicado de atributos.

Se ha calculado también una metaordenación (ordenación de ordenaciones) de los niveles de reducción realizando un nuevo test de Friedman, pero en este caso sobre las puntuaciones (ver Tabla 7). Esto nos permite evaluar qué número de atributos ofrece el mejor rendimiento global para la mayoría de las métricas, lo que nos permitiría obtener una metaordenación de los niveles de reducción. Es interesante observar que la mejor puntuación no corresponde al conjunto completo de atributos. Puesto que el test de Friedman indica diferencias significativas entre niveles de reducción (p-valor ≤ 0.05), se ha realizado un test a posteriori Bonferroni-Dunn, que indica con un nivel de confianza del 95% que los algoritmos tienen un rendimiento significativamente peor con 150 atributos o menos. Esto nos lleva finalmente a considerar que para nuestro conjunto de datos el mejor nivel de reducción global es de 250 atributos, ya que es el menor número de atributos que podemos seleccionar sin que haya una pérdida estadísticamente significativa de rendimiento. La Figura 7 proporciona una representación gráfica donde pueden apreciarse las diferencias relativas en los valores de metaordenación analizados según el test de Bonferroni-Dunn de la Tabla 7.

Tabla 7. Test Friedman y Bonferroni-Dunn para la metaordenación correspondiente a nivel de ordenación

p-valor Friedman = 7,802E-11		Bonferroni-Dunn post test	
Nivel reducción	Puntuación (orden)	p-valor	Hipótesis nula
50	7,406 (8)	0	**Rechazada**
100	7,281 (7)	0	**Rechazada**
150	6,000 (6)	3,720E-04	**Rechazada**
250	3,719 (5)	1,115	Aceptada
1336	3,281 (4)	2,569	Aceptada
500	3,094 (3)	3,451	Aceptada
750	2,719 (2)	5,604	Aceptada
1000 (control)	2,500 (1)	-	-

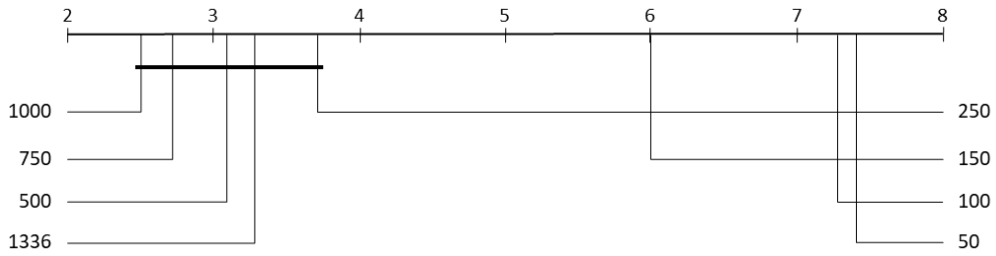

Figura 7. Diagrama de distancia crítica correspondiente al test de Bonferroni-Dunn para la metaordenación correspondiente al nivel de reducción

4.2 Adición de información de uso

Nuestro segundo experimento está orientado a descubrir si añadiendo información relativa al uso que se ha hecho en el pasado de OA similares a un nuevo OA, es posible mejorar la clasificación del nuevo OA. Para llevar a cabo este experimento tomamos como conjunto de datos base el que nos indican los resultados del experimento anterior, es decir, un conjunto en el que los atributos de contenido se reducen a los 250 mejores. A este conjunto de datos base añadiremos información de uso recogida en 15 atributos (ver sección 3.1 Recogida y preprocesamiento de datos). De manera similar a como hemos hecho en el primer experimento, construimos distintos conjuntos de datos que se diferencian por la cantidad de información de uso acumulada en los 15 atributos indicados, según se tenga en cuenta información de uso de un periodo de tiempo mayor o menor. Así, tenemos un conjunto de datos que no incluye información de uso alguna (Contenido). A continuación tenemos otro conjunto de datos en el que a la información de contenido se le añade la información de uso recogida a lo largo del año 2009. Luego tenemos otro conjunto de datos en el que se incluye la información de contenido y la información de uso acumulada a lo largo de los años 2009 y 2010. Y así sucesivamente hasta terminar con el conjunto de datos que recoge información de contenido e información de uso desde 2009 hasta 2016. Sobre cada uno de estos conjuntos de datos hemos aplicado los consabidos 13 algoritmos de clasificación multietiqueta, cuyos resultados hemos evaluado con las 16 métricas de rendimiento.

A modo de ejemplo, la Tabla 8 muestra los resultados obtenidos para una de las métricas, Hamming loss, para cada algoritmo y conjunto de datos. Se aprecia que casi siempre los mejores resultados (Hamming loss menor) se obtienen usando información

acumulada a lo largo de 3 años (2011). Hay un caso en que el mejor resultado se ha obtenido a partir de la información obtenida a lo largo de dos años (2010).

Tabla 8. Hamming loss según cantidad de información de uso acumulada por año

Alg./Datos	Cont.	2009	2010	2011	2012	2013	2014	2015	2016
BR(J48)	0,247	0,246	0,157	0,141	0,156	0,162	0,174	0,189	0,177
CC(J48)	0,258	0,260	0,156	0,142	0,161	0,162	0,175	0,185	0,179
CLR(J48)	0,246	0,249	0,154	0,139	0,157	0,161	0,175	0,187	0,187
LP(J48)	0,240	0,235	0,164	0,154	0,186	0,195	0,205	0,222	0,203
PS(J48)	0,244	0,230	0,159	0,160	0,185	0,190	0,209	0,225	0,211
EPS(J48)	0,230	0,223	0,149	0,141	0,163	0,168	0,176	0,186	0,186
ECC(J48)	0,237	0,233	0,145	0,136	0,153	0,157	0,165	0,175	0,173
RAkEL	0,223	0,222	0,147	0,133	0,157	0,160	0,174	0,181	0,179
HOMER	0,248	0,251	0,177	0,156	0,172	0,180	0,191	0,201	0,198
Stacking	0,246	0,245	0,155	0,140	0,157	0,168	0,178	0,184	0,177
AdaBoost.MH	0,255	0,270	0,270	0,248	0,268	0,264	0,268	0,269	0,269
MLkNN	0,241	0,248	0,200	0,183	0,188	0,196	0,204	0,206	0,209
IBLR	0,227	0,242	0,188	0,185	0,191	0,189	0,195	0,198	0,205

Procediendo de manera análoga a la seguida en el primer experimento, se ha realizado un test de Friedman para cada métrica, con el objetivo de determinar si hay diferencias significativas en la calidad de la clasificación según la cantidad de información de uso acumulada. Se presentan en la Tabla 9 los resultados para Hamming loss. El p-valor indica que hay diferencias significativas, por lo que procedemos a aplicar el test a posteriori Bonferroni-Dunn, en el que el caso de control corresponde a 2011. Las hipótesis con un p-valor ≤ 0.05 son rechazadas con un nivel de confianza del 95%, indicando que hay diferencia significativa entre ese año y el resto. El punto de corte obtenido en este caso por test de Bonferroni-Dunn indica que no hay diferencias significativas en los resultados obtenidos con la información de uso de 2013, 2012, 2011 y 2010. La Figura 8 muestra las diferencias relativas de los valores del test de Bonferroni-Dunn de la Tabla 9.

Tabla 9. Test de Friedman y Bonferroni-Dunn para Hamming loss según información de uso

p-valor Friedman = 5,069E-11		Bonferroni-Dunn post test	
Información uso	Puntuación (orden)	p-valor	Hipótesis nula
2009	8,423 (9)	0	**Rechazada**
Contenido	8,077 (8)	0	**Rechazada**
2015	6,692 (7)	1,000E-06	**Rechazada**
2016	6,154 (6)	1,800E-05	**Rechazada**
2014	5,154 (5)	1,179E-03	**Rechazada**
2013	3,769 (4)	0,098	Aceptada
2012	2,923 (3)	0,685	Aceptada
2010	2,731 (2)	0,989	Aceptada
2011 (control)	1,077 (1)	-	

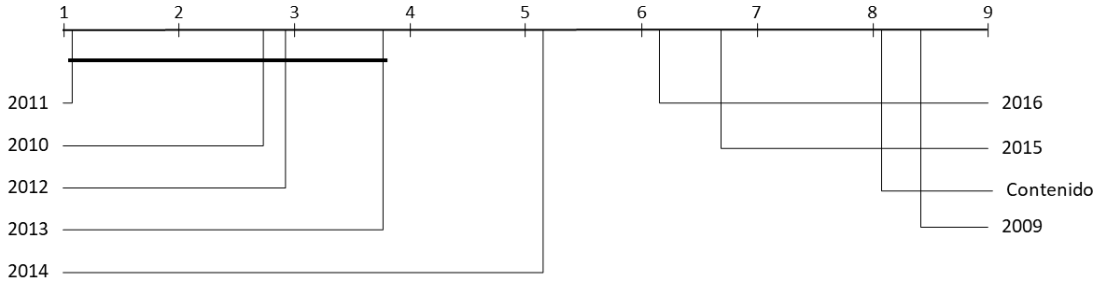

Figura 8. Diagrama de distancia crítica correspondiente al test de Bonferroni-Dunn para Hamming loss según información de uso

La Tabla 10 recoge las puntuaciones obtenidas por cada cantidad de información de uso para cada una de las métricas de rendimiento. Estos resultados muestran que el rendimiento óptimo se obtiene cuando se tiene en cuenta la información de uso recogida a lo largo de tres años (2011), aunque el rendimiento no es significativamente peor empleando los datos recogidos en solo dos años (2010).

Tabla 10. Puntuaciones medias y puntos de corte

Medida/Año	Contenido	2009	2010	2011	2012	2013	2014	2015	2016	Corte
↑Avg prec	8,462	7,923	2,846	1,385	3,308	3,923	5,231	6,154	5,769	2010
↓Coverage	8,385	8,000	3,923	1,539	3,077	3,462	4,462	6,154	6,000	2010
↑E-based acc	8,077	8,385	3,231	1,154	3,154	3,462	5,154	6,308	6,077	2010
↑E-based f-m	7,923	8,539	3,769	1,154	3,077	3,308	5,308	6,308	5,615	2010
↑E-based prec	8,385	7,769	3,000	1,154	3,192	3,885	5,154	6,615	5,846	2010
↑E-based rec	6,077	7,154	6,077	1,692	4,077	3,846	3,923	6,539	5,615	2012
↓Ham loss	8,077	8,423	2,731	1,077	2,923	3,769	5,154	6,692	6,154	2010
↑Macro-avg f-m	8,385	8,154	2,846	1,385	3,308	3,692	4,846	6,231	6,154	2010
↑Macro-avg prec	8,539	7,615	1,923	1,769	3,923	4,154	4,692	6,308	6,077	2010
↑Macro-avg rec	8,077	8,231	5,077	2,231	3,308	2,769	3,615	5,923	5,769	2010
↑Micro-avg f-m	7,923	8,539	2,846	1,077	3,077	3,769	5,154	6,462	6,154	2010
↑Micro-avg prec	8,462	7,692	2,385	1,154	3,077	3,923	5,385	6,769	6,154	2010
↑Micro-avg recall	7,077	8,000	5,846	1,769	3,692	3,077	3,923	6,077	5,539	2012
↓One error	8,385	8,000	2,539	1,385	3,462	3,846	5,808	6,192	5,385	2010
↓Ranking loss	8,385	8,000	3,462	1,385	3,154	3,769	4,462	6,231	6,154	2010
↑Subset accuracy	8,539	8,385	3,308	1,269	2,500	3,654	4,577	6,423	6,346	2010
Puntuación media	8,072	8,050	3,488	1,411	3,269	3,644	4,803	6,337	5,925	

También en este caso hemos calculado una metaordenación, referente en esta ocasión a la información de uso recopilada a lo largo de los años, para evaluar qué periodo proporciona el mejor rendimiento global para el conjunto de las métricas. Estos resultados se muestran en la Tabla 11. En este caso se aprecia que no hay diferencia estadísticamente significativa para los periodos hasta 2010, 2011 y 2012 y 2013. En nuestro caso nos interesa quedarnos con el periodo más corto, ya que implica esperar un menor número de años para obtener una información de uso que sea capaz de mejorar la calidad de la clasificación, por lo que nos quedaríamos con el periodo hasta 2010. La Figura 9 proporciona una representación gráfica donde pueden apreciarse las diferencias relativas en los valores de metaordenación analizados según el test de Bonferroni-Dunn de la Tabla 11.

Tabla 11. Test Friedman y Bonferroni-Dunn para metaordenación correspondiente a información de uso

p-valor Friedman = 8,938E-11		Bonferroni-Dunn post test	
Información uso	Puntuación (orden)	p-valor	Hipótesis nula
Contenido	8,469 (9)	0	**Rechazada**
2009	8,438 (8)	0	**Rechazada**
2015	7,063 (7)	0	**Rechazada**
2016	5,813 (6)	5,000E-06	**Rechazada**
2014	4,813 (5)	6,590E-04	**Rechazada**
2013	3,500 (4)	0,079	Aceptada
2010	3,156 (3)	0,208	Aceptada
2012	2,750 (2)	0,566	Aceptada
2011 (control)	1,000 (1)	-	-

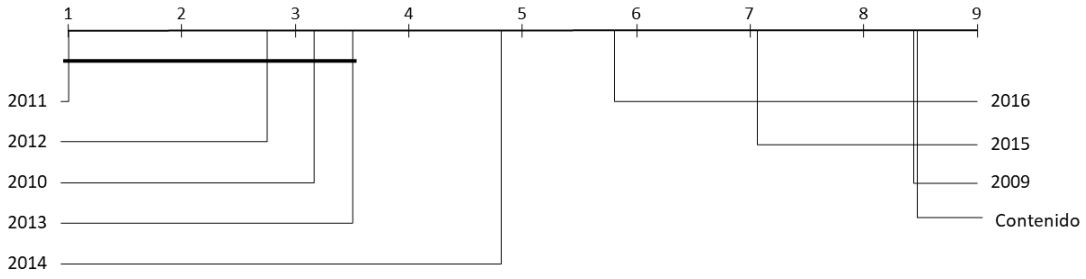

Figura 9. Diagrama de distancia crítica correspondiente al test de Bonferroni-Dunn para la metaordenación correspondiente a la información de uso

4.3 Selección del mejor algoritmo de clasificación multietiqueta

En nuestro tercer y último experimento hemos llevado a cabo un análisis estadístico similar al de los dos experimentos anteriores. Nuestro objetivo ahora es encontrar el mejor algoritmo de clasificación multietiqueta para nuestros datos. El conjunto de datos que hemos utilizado para este tercer experimento es el configurado según los resultados de los dos primeros experimentos, es decir, el obtenido a partir del conjunto de datos original seleccionando los 250 mejores atributos de contenido y añadiendo los atributos de uso correspondientes a los dos primeros años (hasta 2010). A partir de los valores de las 16 métricas de rendimiento obtenidas por los 13 algoritmos de clasificación multietiqueta, hemos obtenido la puntuación de cada algoritmo para cada métrica, que se presenta en la Tabla 12. Esta tabla nos indica que el mejor algoritmo es ECC, ya que es el que obtiene la puntuación máxima más veces (5 de 13 casos), y en los casos en los que no tiene la mayor puntuación sigue teniendo puntuaciones bastante buenas (valores nunca superiores a 4). Si nos fijamos por ejemplo en el método CLR, vemos que este algoritmo es el mejor para 4 métricas, pero para el resto de métricas presenta unas puntuaciones muy variantes y RAkEL, un algoritmo que presenta la mejor puntuación sólo para una métrica, tiene una puntuación media mejor que CLR.

Finalmente, hemos realizado un test de Friedman y un test a posteriori Bonferroni-Dunn a las puntuaciones de rendimiento de los algoritmos teniendo como valor de control el de ECC, que es el mejor algoritmo. Los resultados se muestran en la Tabla 13. Se aprecia que la diferencia respecto a la puntuación de rendimiento de algunos algoritmos no es estadísticamente significativa con la de ECC, a saber, CC, EPS, CLR y RAkEL. Las hipótesis con un p-valor ≤ 0.05 se rechazan con un nivel de confianza del 95%. Los resultados obtenidos indican que hay un grupo de algoritmos de clasificación multietiqueta que ofrecen un rendimiento similar, sin diferencias estadísticamente significativas, para nuestros datos. Así, en caso de estar interesados en obtener unos resultados óptimos podríamos utilizar ECC, que es el algoritmo con mejor puntuación, pero siendo conscientes de que otros algoritmos pueden ofrecer un rendimiento análogo, como son CC, EPS, CLR y RAkEL.

Tabla 12. Puntuación del rendimiento de los algoritmos para cada métrica

Medida/Algoritmo	BR	CC	CLR	LP	PS	EPS	ECC	RAkEL	HOMER	MLS	AdaBoost.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

Tabla 13. Test Friedman y Bonferroni-Dunn para puntuación del rendimiento de los algoritmos

p-valor Friedman = 9,243E-11		Bonferroni-Dunn post test	
Algoritmo MLC	Puntuación (orden)	p-valor	Hipótesis nula
AdaBoost.MH	12,750 (13)	0,000E+00	**Rechazada**
MLkNN	11,313 (12)	0,000E+00	**Rechazada**
IBLR	10,250 (11)	0,000E+00	**Rechazada**
LP	9,375 (10)	1,000E-06	**Rechazada**
HOMER	8,063 (9)	1,280E-04	**Rechazada**
BR	6,563 (8)	1,105E-02	**Rechazada**
MLS	6,375 (7)	1,783E-02	**Rechazada**
PS	6,188 (6)	2,827E-02	**Rechazada**
CC	5,750 (5)	7,751E-02	Aceptada
EPS	4,313 (4)	1,117	Aceptada
CLR	4,250 (3)	1,227	Aceptada
RAkEL	3,813 (2)	2,257	Aceptada
ECC (control)	2,000 (1)	-	-

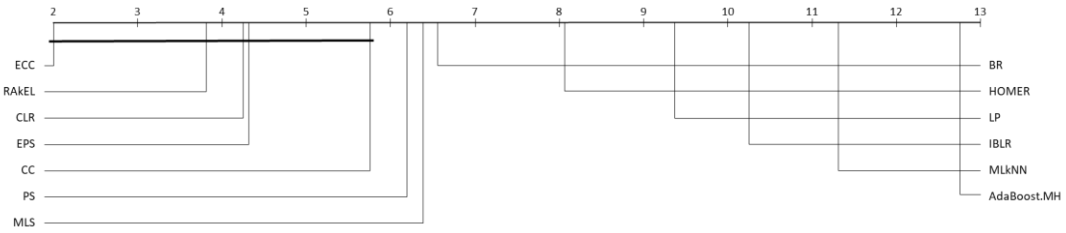

Figura 10. Diagrama de distancia crítica correspondiente al test de Bonferroni-Dunn para puntuación del rendimiento de los algoritmos

5

CONCLUSIONES

En esta tesis hemos propuesto un enfoque novedoso para categorizar automáticamente OA mediante el uso de algoritmos de clasificación multietiqueta. Hemos mejorado el rendimiento de la clasificación multietiqueta mediante la reducción del número de atributos de contenido y añadiendo información histórica de uso de OA similares. Hemos comparado 13 algoritmos de clasificación multietiqueta en base a 16 métricas de rendimiento. Hemos llevado a cabo nuestra investigación sobre un conjunto de datos con 519 instancias provenientes del ROA AGORA. Hemos usado el software MLDA, MULAN y KEEL para nuestros experimentos. Nuestra investigación ha permitido satisfacer los objetivos O_1, O_2 y O_3, lo cual a su vez ha permitido validar las hipótesis H_1, H_2 y H_3.

- Hemos comprobado que, efectivamente, es posible reducir el número de atributos de contenido que describen los OA sin que esto perjudique la calidad de clasificación. En nuestro caso, hemos logrado un nivel de reducción muy considerable, desde los 1336 atributos originales hasta 250.

- Hemos comprobado que, efectivamente, podemos mejorar la calidad de clasificación añadiendo al conjunto de datos información relativa al uso que se ha hecho en el pasado de OA similares a uno nuevo que se vaya a añadir al repositorio. En nuestro caso, hemos podido mejorar la calidad de clasificación añadiendo información de uso acumulada a lo largo de un periodo de dos años.

- Hemos comprobado que, efectivamente, podemos distinguir un grupo de algoritmos de clasificación multietiqueta que son los que mejores resultados ofrecen a la hora de categorizar los OA. Para nuestros datos, el algoritmo que

mejor funciona es ECC, aunque hay otros que tienen un nivel de rendimiento que no es significativamente peor: CC, EPS, CLR y RAkEL.

5.1 Futuras mejoras

En el futuro queremos extender nuestra experimentación utilizando también datos de otros repositorios, con el objetivo de poder alcanzar unas conclusiones más genéricas respecto al efecto de la reducción del número de atributos de contenido, adición de información histórica de uso y selección del mejor algoritmo de clasificación multietiqueta.

Otro objetivo futuro es el de integrar nuestro modelo dentro del repositorio AGORA para que pueda llevar a cabo la recomendación en tiempo real en el momento en que un usuario añade un nuevo OA al repositorio.

5.2 Contribuciones científicas

Se indican a continuación la comunicación a congreso internacional y el artículo en revista con índice de impacto cuya publicación avala el trabajo descrito en esta tesis:

- **Artículo en congreso internacional CORE B:**

González, P., Gibaja, E., Zapata, A., Menéndez, V. H., Romero, C. Towards Automatic Classification of Learning Objects: Reducing the Number of Used Features, EDM 2017. 394-395. Wuhan, China, 25-28/06/2017.

- **Artículo en Revista con índice de impacto (incluida en el JCR-2020):**

Espejo, P. G., Gibaja, E., Menéndez, V. H., Zapata, A., Romero, C. Improving Multi-Label Classification for Learning Objects Categorization by Taking into Consideration Usage Information. Journal of Universal Computer Science. Volume 25 (Issue 13): 1687-1716. 2019.

REFERENCIAS

ADL. 2016. *SCORM*. https://www.adlnet.gov/adl-research/scorm/ [visitado 21/07/2016].

Alcalá, J., Sánchez, L., García, S., del Jesus, M. J., Ventura, S., Garrell, J. M., Otero, J., Romero, C., Bacardit, J., Rivas, V. M., Fernández, J. C., Herrera, F. 2009. KEEL: a software tool to assess evolutionary algorithms to data mining problems. *Soft Computing*, 13 (3), 307-318.

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

ARIADNE Foundation. 2016. *ARIADNE*, http://www.ariadne-eu.org/ [visitado 21/07/2016].

Berners-Lee, T., Hendler, J., Lassila, O. 2001. The semantic web. *Scientific american*, 284(5), 34-43.

Blanco, J. J., Galisteo del Valle, A., García, A. 2008. Perfil de aplicación LOM-ES V.1.0. Asociación Española de Normalización y Certificación (AENOR).

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

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

El Saddik, A., Fischer, S., Steinmetz, R. 2001. Reusability and adaptability of interactive resources in Web-based educational systems. *ACM Journal Educational Resources in Computing*, 1(1).

Fayyad, U. M.; Piatetsky-Shapiro, G.; Smyth, P. 1996. From data mining to knowledge discovery in databases. *AI Magazine*, 17 (3), 37-54.

Freund, Y., Schapire, R. E. 1995. A desicion-theoretic generalization of on-line learning and an application to boosting. *European Conference on Computational Learning Theory*, 23-37.

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

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

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

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

IEEE. 2016a. *Learning Object Metadata*, http://grouper.ieee.org/groups/ltsc/wg12/ [visitado 21/07/2016].

IEEE. 2016b. *IEEE Standard for Learning Object Metadata*. https://standards.ieee.org/findstds/standard/1484.12.1-2002.html [visitado 21/07/2016].

Ihsan, I., Mohib-Ur-Rehman, Ahmed, M. U., Qadir, M. A., 2006. UREKA learning-object taxonomy & repository architecture - ULTRA. *10th IEEE International Multitopic Conference (INMIC 2006)*, 231-236.

IMS Global Learning Consortium. 2016. *Learning Design Specification*. https://www.imsglobal.org/learningdesign/index.html [visitado 21/07/2016].

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

Kannampallil, T. G., Farrell, R. G. 2005. Automatic learning object categorization for instruction using an enhanced linear text classifier. *Knowledge Management: Nurturing Culture, Innovation, and Technology*, 299-304.

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

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

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

MERLOT. 2016. *MERLOT - Multimedia Educational Resource for Learning and Online Teaching*, https://www.merlot.org/merlot/index.htm [visitado 21/07/2016].

Moyano, J. M., Gibaja, E. L., Ventura, S. 2017. MLDA: a tool for analyzing multi-label datasets. *Knowledge-Based Systems*, 121, 1-3.

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

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

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

Redeker, G. H. J. 2003. An educational taxonomy for learning objects. *3rd IEEE International Conference on Advanced Learning Technologies (ICALT 2003)*, 250-251.

Romero, C., Ventura, S. 2010. Educational data mining: a review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics*, 40 (6), 601-618.

Schapire, R. E., Singer, Y. 2000. BoosTexter: a boosting-based system for text categorization. *Machine Learning*, 39 (2-3), 135-168.

Stefaner, M., Vecchia, E. D., Condotta, M., Wolpers, M., Specht, M., Apelt, S., Duval, E. 2007. MACE - enriching architectural learning objects for experience multiplication. *Creating New Learning Experiences on a Global Scale: Second European Conference on Technology Enhanced Learning (EC-TEL 2007)*, 322-336.

Tan, P.-N., Steinbach, M., Karpatne, A., Kumar, V. Introduction to Data Mining (2nd ed.) Pearson, 2018.

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

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

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

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

Wiley, D. A. 2002. Connecting learning objects to instructional design theory: a definition, a metaphor, and a taxonomy. *The Instructional Use of Learning Objects: online version*. 1-35.

Wolpert, D. H. 1992. Stacked generalization. *Neural Networks*, 5 (2), 241-259.

Zapata, A., Menéndez, V. H., Prieto, M.E., Romero, C. 2013. A framework for recommendation in learning object repositories: an example of application in civil engineering. *Advances in Engineering Software*, 56, 1-14.

Zapata, A., Menéndez, V. H., Prieto, M. E., Romero, C. 2015. Evaluation and selection of group recommendation strategies for collaborative searching of learning objects. *International Journal of Human-Computer Studies*, 76, 22-39.

Zhang, M. L., Zhou, Z. H. 2007. ML-KNN: a lazy learning approach to multi-label learning. *Pattern Recognition*, 40 (7), 2038-2048.