Deanship of Graduate Studies
Al-Quds University

Consolidated Ranking and Recommendation Framework for Learning Objects Based on Usage Data

Baha' Hamzah Harasheh

M.Sc. Thesis

Jerusalem – Palestine

1434 / 2013

Deanship of Graduate Studies
Al-Quds University

Consolidated Ranking and Recommendation Framework for Learning Objects Based on Usage Data

Baha' Hamzah Harasheh

M.Sc. Thesis

Jerusalem – Palestine

1434 / 2013

Consolidated Ranking and Recommendation Framework for Learning Objects Based on Usage Data

Prepared By:

Baha' Hamzah Harasheh

B.Sc.: Birzeit University Palestine

Supervisor: Dr. Jad Najjar
Co-Supervisor: Dr. Rashid Jayousi

A thesis Submitted in Partial fulfillment of requirements for the Degree of Master of Computer Science - Al-Quds University

1434 / 2013

Al-Quds University
Deanship of Graduate Studies
Master in Computer Science
Computer Science & Information Technology

Thesis Approval

Consolidated Ranking and Recommendation Framework for Learning Objects Based on Usage Data

Prepared By: Baha' Hamzah Harasheh
Registration No: 20911886

Supervisor: Dr. Jad Najjar
Co-Supervisor: Dr. Rashid Jayousi

Master thesis submitted and accepted, Date: ……………………
The names and signatures of the examining committee members are as follows:

1- Head of Committee: Dr. Jad Najjar Signature …..………...
2- Internal Examiner: Dr. Raid Zaghal Signature …..………...
3- External Examiner: Dr. Derar Eleyan Signature …..………...
4- Committee Member: Dr. Rashid Jayousi Signature …..………...

Jerusalem – Palestine

1434 / 2013

بسم الله الرحمن الرحيم

"اللَّهُ لا إِلَهَ إِلا هُوَ الْحَيُّ الْقَيُّومُ لا تَأْخُذُهُ سِنَةٌ وَلا نَوْمٌ لَهُ مَا فِي السَّمَاوَاتِ وَمَا فِي الأَرْضِ مَنْ ذَا الَّذِي يَشْفَعُ عِنْدَهُ إِلا بِإِذْنِهِ يَعْلَمُ مَا بَيْنَ أَيْدِيهِمْ وَمَا خَلْفَهُمْ وَلا يُحِيطُونَ بِشَيْءٍ مِنْ عِلْمِهِ إِلا بِمَا شَاءَ وَسِعَ كُرْسِيُّهُ السَّمَاوَاتِ وَالأَرْضَ وَلا يَئُودُهُ حِفْظُهُمَا وَهُوَ الْعَلِيُّ الْعَظِيمُ"

سورة البقرة، آية 255

In the name of Allah, the Beneficent, the Merciful

"Allah - there is no god but He, the Ever-living, the Self-subsisting by Whom all subsist. Slumber overtakes Him not, nor sleep. To Him belongs whatever is in the heavens and whatever is in the earth. Who is he that can intercede with Him but by His permission? He knows what is before them and what is behind them. And they encompass nothing of His knowledge except what He pleases. His knowledge extends over the heavens and the earth, and the preservation of them both tires Him not. And He is the Most High, the Great"

The Holy Quran, Al-Baqarah, 2:255

Dedication

To my Mother
To my Mother
To my Mother
To my Father
To my Wife
To my Son Abdullah
To my Brothers
To my Sisters

Declaration

I certify that this thesis submitted for the degree of Master, is the result of my own research, except where otherwise acknowledged, and that this study (or any part of the same) has not been submitted for a higher degree to any other university or institution.

Signed: …………………

Baha' Hamzah Harasheh

Date: 28 July 2013

Acknowledgments

I would like to thank Al-Quds University for giving me the chance to be a student in master program, and work on this thesis.

Many thanks for my supervisors Dr. Jad Najjar and Dr. Rashid Jayousi. Thank you for guidance and support.

Thanks for other master program members, Dr. Badie Sartawi, Dr. Nedal Kafri, Dr. Raid Zaghal, and Dr. Wael Hassouneh. This thesis is a result of knowledge I grasped from all master courses.

Special appreciation to the external examiner Dr. Derar Eleyan, and the internal examiner Dr. Raid Zaghal. Thank you for your time.

Many thanks for Eummena project for their support and help.

Thank you for all my colleagues and friends who helped me in evaluating the framework.

Abstract

Share and reuse of learning materials (objects) is one of the main goals of educational repositories. Finding appropriate reusable learning materials is still one of the highest challenges facing users of educational repositories. Users still are not able to find enough high quality learning materials relevant to them. Several tools were developed to improve searching learning objects (resources), but most of these tools are content based oriented, and depend on descriptive information (metadata) of learning resources created during indexation of resources into the repositories. This thesis proposes a Consolidated Ranking and Recommendation Framework (CRRF) that uses usage data to improve finding of relevant learning objects. The proposed work includes analysis of usage data to create dynamic user profiles automatically to improve Ranking and Recommendation of learning materials. User profiles will have information about user's contexts, learning interests, search objectives, and information about peer users. Appropriate data mining techniques are used to analyse usage data, and create user profiles. These techniques have been selected based on their effectiveness and efficiency. The consolidated framework provides flexible criteria and formulas to allow extension of the services, functionalities, and capabilities of CRRF by adding new sources of data based on work of others. Ranking is the process of sorting learning resources to the user according to his/her contexts and search objectives. Ranking works when a user performs search query on learning materials. Recommendation is the process of suggesting learning materials to the user according to user interest based on frequently used materials or according to learning material that was of interest to his/her peer users.

ملخص الدراسة

مشاركة وإعادة استخدام مواد التعلم هي واحدة من الأهداف الرئيسية للموارد التعليمية. البحث عن مواد التعلم المناسبة القابلة لإعادة الاستخدام لا يزال واحداً من أكثر التحديات التي تواجه مستخدمي الموارد التعليمية. هؤلاء المستخدمين عادة غير قادرين على العثور على العديد من مواد التعلم ذات الجودة العالية أو المناسبة لهم. لقد تم تطوير العديد من الأدوات لتحسين البحث عن مواد التعلم، ولكن معظم هذه الأدوات تعتمد على محتوى مواد التعلم والبيانات الوصفية لتلك المواد فقط. في هذه الأطروحة، تم اقتراح النظام الموحد لترتيب وتوصية مواد التعلم الذي يعتمد على بيانات الاستخدام لتحسين البحث عن مواد التعلم ذات الصلة والمناسبة للمستخدم. العمل المقترح يشمل تحليل بيانات الاستخدام لإنشاء ملفات تعريف المستخدمين بشكل تلقائي لتحسين عملية الترتيب والتوصية. هذه الملفات تشمل معلومات حول سياقات المستخدم، واهتمامات التعلم، وأهداف البحث، ومعلومات حول المستخدمين الذين تربطهم علاقات مع المستخدم. تم استخدام تقنيات التنقيب عن البيانات لتحليل بيانات الاستخدام وإنشاء ملفات تعريف المستخدم. وقد تم اختيار التقنيات المناسبة لضمان دقة عالية خلال فترة زمنية مناسبة. يوفر النظام الموحد معايير وصيغ مرنة تسمح للباحثين بإضافة عملهم في هذا الإطار لتعزيز كفاءة ترتيب وتوصية مواد التعلم. الترتيب هو عملية فرز مواد التعلم للمستخدم وفقاً لسياقاته واهتماماته التعليمية وأهداف البحث. الترتيب يعمل عند قيام المستخدم بعملية استعلام عن مواد التعلم. التوصية هي عملية اقتراح مواد التعلم للمستخدم وفقاً لمواد التعلم المستخدمة بشكل متكرر، ولمواد التعلم الموصى بها من قبل المستخدمين الآخرين.

Table of Contents

List of Tables

List of Figures

Figure No. **Figure Title** **Page**

List of Appendices

List of Abbreviations

Abbreviation	Full Name
3A	Three Actors
3D	Three-Dimensions
ALOHA	Advanced Learning Object Hub Application
ANN	Artificial Neural Network
API	Application Programming Interface
APML	Attention Profiling Markup Language
B.Sc.	Bachelor in Science
BN	Bayes Net
CAM	Contextualized Attention Metadata
CAMf	Contextualized Attention Metadata framework
CAMs	Contextualized Attention Metadata schema
CAREO	Campus Alberta Repository of Educational Objects
CBT	Computer-Based Training
CML	Context Modeling Language
CNB	Complement Naïve Bayes
COM	Component Object Model
CPU	Central Processing Unit
CRRF	Consolidated Ranking and Recommendation Framework
CSKM	Cascade Simple K Means
DCMES	The Dublin Core Metadata Element Set
DCMI	Dublin Core Metadata Initiative
DMNB	Discriminative Multinomial Naïve Bayes
DMNBT	Discriminative Multinomial Naïve Bayes Text
DRI	Digital Repository Interoperability
ECL	eduSource Communication Layer
EdNA	Education Network Australia
ELKI	Environment for Developing KDD-Applications Supported by Index-Structures
EM	Expectation Maximisation
EPUB	Electronic Publication Format
ER	Entity Relationship
ETL	Extract, Transform and Load
FA	Filtered Associator
FF	Farthest First
FN	False Negative
FP	False Positive
GAS	Gathering and Analysis System
GB	Giga Byte
GNU	General Public License
GSP	Generalized Sequential Patterns
HR	Human Rank
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IBk	Instance-based K-nearest neighbours
ICU4J	International Components for Unicode for Java
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers

Abbreviation	**Full Name**
ILMS	Intelligent Learning Management System
IMS	Instructional Management Systems
IP	Internet Protocol
ISO	International Organization for Standardization
JDBC	Java Database Connectivity
KDD	Knowledge Discovery and Data Mining
LBR	Lazy Bayesian Rules
LBR	Lazy Bayesian Rules
LibSVM	Library for Support Vector Machines
LMS	Learning Management System
LO	Learning Object
LOM	Learning Object Metadata
LTSC	Learning Technology Standards Committee
LVQ	Learning Vector Quantization
LWL	Locally Weighted Learning
M.Sc.	Master in Science
MB	Mega Byte
MDBC	Make Density Based Clusterer
MERLOT	Multimedia Educational Resource for Learning and Online Teaching
MILOS	Multi-agent Infrastructure for Learning Object Support
MLP	Multilayer Perceptron
MLPClass	MLP Classifier (ANN) (Multi Layer Perceptron)
MLR	Metadata for Learning Resources
MLX	Maricopa Learning Exchange
MS	Microsoft
MSLF	Multi-layered Semantic LOM Framework
NB	Naïve Bayes
NBM	Naïve Bayes Multinomial
NBMT	Naïve Bayes Multinomial Text
NBMU	Naïve Bayes Multinomial Updateable
NBU	Naïve Bayes Updateable
OBAA	Agent-Based Learning Objects Metadata Standard
ODF	OpenDocument Format
ODM	Oracle Data Mining
OER	Open Educational Resources
OLE	Object Linking and Embedding
OOXML	Office Open XML
OPTICS	Ordering Points To Indentify the Clustering Structure
OS	Operating System
OWL	Web Ontology Language
PA	Predictive Apriori
PDF	Portable Document Format
PPLOM	Preferred Personalization Learning Object Model
RBF	Radial Basis Function
RDF	Resource Description Framework
RPC	Remote Procedure Call
RTF	Rich Text Format
SE	Standard Edition

Abbreviation	**Full Name**
SKM	Simple K Means
SMO	Sequential Minimal Optimization
SNA	Social Network Analysis
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SOM	Self Organizing Map
SPARQL	SPARQL Protocol and RDF Query Language
SQL	Structured Query Language
SR	System Rank
SSAS	SQL Server Analysis Services
SUS	System Usability Scale
SVC	Support Vector Classification
TN	True Negative
TP	True Positive
UIMA	Unstructured Information Management Applications
UNL	Universal Networking Language
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
XML	Extensible Markup Language

Chapter One

Introduction

A Learning management system (LMS) is the main application used to deliver online E-learning for experience students and teachers. Use of learning management systems became common in most universities, for example, most of the Palestinian universities use E-learning system in their education, and it is used in flexible (open) learning and face-to-face teaching as well. One of the most popular learning management systems in Palestine and worldwide is Moodle (Moodle, 2012) which is an open source system and has the ability to be customized according to needs of institutions.

Effective usage for LMS depends on learning materials available in that system. Creating learning materials with high quality requires high cost, and reuse learning materials will compensate the high cost and save time (Duval, 2004). The key to implement reusable learning materials is learning object.

Search and find relevant learning objects is very important to enhance learning objects reusability, and save expenses of design high quality learning materials. Search tools may face many challenges that prevent users from getting learning objects they need.

This thesis proposes Consolidated Ranking and Recommendation Framework (CRRF) based on usage data to improve learning objects usability.

1.1 Motivation

Design of high quality learning materials is expensive and time consuming. In order to reduce the time and cost of the design and production of such learning materials, it is recommended to use modular strategy where the learning objects are divided into smaller units that will be published in repositories so these units can be reused by other users. Reuse of learning objects is the main purpose of learning object repositories (Duval, 2004).

Repositories may have thousands of learning objects, and there is a real need for effective search tools that can locate learning objects as well as related learning objects in the most efficient way. This helps the user to build learning materials effectively and with high quality as the repository usually contains learning objects built by experts in the different fields (Najjar, 2008).

Search tools provide users with electronic form to enter their search query, and then use it to search within learning objects metadata.

Main challenges for such search tools are incomplete learning objects metadata, limited search parameters entered in search forms, and user context. First challenge is incomplete learning objects metadata which happens when authors of learning objects find it difficult to spend extra time and effort to add metadata to their learning objects (Duval & Hodgins, 2003). Second challenge is that users of search tools do not intend to fill in enough parameters in their search queries, and prefer to give only short keywords and this cause the limited number of search parameters (Najjar, Ternier, & Duval, 2004). The third challenge is the ability of search tools in finding relevant learning objects relevant to the user according to his context (Najjar, 2008).

Current search tools suffer from one of two problems: low recall and oversupply. 1) Search tools with high precision may return no result for user, and this will discourage him from using system again, and will cause low recall problem (Sokvitne, 2000) (Ochoa, 2008). 2) Oversupply problem happens when simple search tools provide users with many results, which require them to check many pages of results lists manually to find relevant learning objects (Ochoa, 2008) (Najjar, Klerkx, Vuorikari, & Duval, 2005).

1.2 Objectives

The main objective of this thesis is to find a solution for searching learning objects problems (low recall and oversupply) by ranking search results based on their relevance. This approach will provide searchers with all returned result from simple search, and also rank them to allow the searcher find most relevant objects easily. Also provides recommendation of learning objects during navigating and using the system will improve learning objects reusability.

All methodologies, mentioned in related work section, tried to solve particular problems in searching learning objects, but none of them tried to develop a consolidated ranking and recommendation framework that allows all researchers to participate towards global improvement. For example, In the research (Fouad Ibrahim, 2012), the author proposed ranking and recommendation functions based on learner profile, but his framework is not flexible, and cannot use usage data or data mining techniques to build the profile. In another research (Chen & Huang, 2012), authors proposed recommendation framework that based on learner profile domain and learner profile settings, but again data mining techniques nor usage data are used, also the framework doesn't allow researcher to add any additional data to the framework. Another example is the research (Govaerts, El Helou, Erik, & Gillet, 2011), authors proposed a federated search engine that search for learning objects within many learning object repositories, then use built-in recommendation service, but this framework has no user profile, doesn't use data mining techniques, and doesn't allow flexible ranking and recommendation criteria. This thesis will try to find a general framework that can be used by all researchers in this domain and allow them to contribute in improve ranking and recommendation of learning objects. Also we designed and developed algorithm which is compliant with this framework based on usage data.

1.3 Methodology

Find a solution for searching learning objects is the main objective, so this thesis tried to collect information about related work in searching learning objects and applications of data mining in learning technology. Available frameworks were analysed, and new abstract framework has been proposed. The proposed framework tried to provide a system with high flexibility and ability to improve searching learning objects. Data mining techniques are used to analyse usage data and generate dynamic user profiles. User profile depends on the available data in learning objects metadata, related work, and ability of data mining techniques. User profiles are the base knowledge for ranking and recommendation modules that have formulas to calculate learning objects rank, and recommend learning objects for users. Evaluation of data mining techniques was applied to choose best algorithms with ability to get high correctness within reasonable time. At the end of the development of the proposed framework, evaluation for the entire framework has been performed to make sure that the objective can be achieved and determine the added value of the proposed approach. Two methods are used to evaluate the framework: Precision and Recall metrics, and System Usability Scale (SUS) survey. The framework has been published on internet, and fifteen users used the system for two weeks. At the end of the evaluation period, the log data from the system were collected and analysed; in addition to the SUS survey that were filled by the users.

1.4 Thesis Structure

This thesis is organized in seven chapters. Chapter one is introduction for the thesis. Background on learning objects and data mining is given in chapter two. Chapter three explains related work for searching learning objects and data mining applications in learning technology. The research methodology is explained in chapter four. Chapter five describes the proposed Consolidated Ranking and Recommendation Framework (CRRF) in detail. Chapter six presents evaluation results of the data mining algorithms and CRRF with results. Conclusions of the research and recommendations for further research are given in chapter seven.

Chapter Two

Background

This thesis is based on two main topics: learning objects and data mining. The sections below give background information about these two topics. Learning objects topics include learning objects metadata, learning object repositories, learning object life cycle, issues and challenges, and attention metadata. Data mining topics include data mining techniques, data mining tools, and algorithms available in Weka library (Weka 3: Data Mining Software in Java, 2013).

2.1 Learning Objects

The idea of learning objects in learning technologies is adopted from object oriented programming languages (Sosteric & Hesemeier, 2004). Learning objects have similar meaning of classes and objects in programming languages, but are special for learning technology.

The basic idea of learning objects is to create small instructional components that can be used in many courses and in different contexts (Wiley, 2002).

One definition for learning objects is set by the IEEE Learning Technology Standards Committee (IEEE Draft Standard, 2005, p. 6). They defined it as “any entity, digital or non-digital, that may be used for learning, education, or training”. This definition is general and includes non-digital objects into reusable materials.

The scope of this thesis is the computer science, so we prefer to use different definition that provided by (Wiley, 2002, p. 7). He defined learning object as “any digital resource that can be reused to support learning”. This definition includes only digital resources which look applicable in computer applications and learning management systems.

When an instructor creates teaching material for a course, he can divide this material into small pieces (small in comparison to the whole course), taken into consideration that this pieces can be used by other instructors. According to (Hodgins, 2002), smaller components are more reusable than larger ones, and can be integrated easily with many contexts. Learning object attributes may vary from one context to another, and depends on scope of usage.

Learning object characteristics according to (Rehak & Mason, 2003) are: reusable, accessible, interoperable/portable, and durable. Reusable means that learning object can be modified and versioned for different courses. While accessible means that it can be indexed for easy retrieval using metadata standards. On the other hand, interoperable/portable learning objects can operate across different hardware and software. Last attribute is durable that allows learning object to remain intact through upgrades to hardware and software.

Learning object attributes according to (Downes, 2004) are: shareable, digital, modular, interoperable, and discoverable. Sharable means that learning objects may be available for many institutions and be used in many different courses. While digital means that learning object can be distributed using the internet and network. Combining learning object with other objects into larger entity is known as modular attribute. Interoperability allows different institutions to use learning objects in different applications. Last attribute is discoverable which means that learning object can be found easily.

Learning objects are stored with descriptive metadata (properties) to explain them; these properties are known as metadata. This metadata is very important for learning objects reusability (Ternier, 2008).

2.1.1. Learning Objects Metadata

IEEE Learning Technologies Standard Committee defined the purpose of the metadata as to allow "search, evaluation, acquisition, and use of learning objects" (IEEE Draft Standard, 2005). Therefore, learning object metadata can be defined as any instructional component that can be used to search, evaluate, acquire, and use learning objects.

There are many standards to describe learning object metadata. Metadata standard defines what attributes and data fields needed to describe learning object, their type, and mandatory or optional fields (Ochoa, 2008). The availability of various standards make problem in exchange data between them (interoperability), unless they agree on common metadata standards.

There are several standards for learning object metadata, IEEE Learning Object Metadata (LOM) (IEEE Draft Standard, 2005), The Dublin Core Metadata Element Set (DCMES) (Dublin Core, 2003), and Metadata for Learning Resources (MLR) (ISO/IEC Framework, 2011).

Table 1 compares supported attributes in three learning objects metadata standards. It shows that IEEE LOM supports more attributes, but ISO/IEC is flexible and allows its users to define additional data element specification attributes and the rules governing them (ISO/IEC Framework, 2011).

Table 1: Comparison for Attributes of Learning Objects Metadata.

No.	**Attribute**	**IEEE LOM**	**DCMES**	**ISO/IEC**
1	Title	Yes	Yes	Flexible
2	Language	Yes	Yes	Flexible
3	Description	Yes	Yes	Flexible
4	Keyword	Yes	No	Flexible
5	Version	Yes	No	Flexible
6	Status	Yes	Yes	Flexible
7	Contribute	Yes	Yes	Flexible
8	Meta-Metadata	Yes	Yes	Flexible
9	Format	Yes	Yes	Flexible
10	Size	Yes	No	Flexible
11	Location	Yes	No	Flexible
12	Requirement	Yes	No	Flexible
13	Installation	Yes	No	Flexible
14	Remarks	Yes	No	Flexible
15	Interactivity Type	Yes	Yes	Flexible
16	Learning Resource Type	Yes	Yes	Flexible
17	Interactivity Level	Yes	Yes	Flexible
18	Semantic Density	Yes	Yes	Flexible
19	Context	Yes	Yes	Flexible
20	Difficulty	Yes	No	Flexible
21	Cost	Yes	No	Flexible

22	Copyright and Restrictions	Yes	Yes	Flexible
23	Relation	Yes	Yes	Flexible
24	Annotation	Yes	No	Flexible
25	Classification	Yes	No	Flexible
26	Identifier	No	Yes	Yes

2.1.1.1. IEEE Learning Object Metadata (LOM):

The most used learning object metadata standard is LOM (Learning Object Metadata). This standard uses W3C XML schema (World Wide Web Consortium (W3C) Extensible Markup Language) to define structure, and constraints to represent learning object metadata. LOM XML schema allows this standard to exchange LOM instances between various systems to support interoperability (IEEE Draft Standard, 2005).

IEEE LOM proposes around 45 metadata fields grouped into nine categories (IEEE Draft Standard, 2005). These categories are described in Appendix 1.

2.1.1.2. The Dublin Core Metadata Element Set (DCMES):

The DCMES is a standard for cross-domain information resource description, and maintained by Dublin Core Metadata Initiative (DCMI). According to DCMES, this metadata standard can be assigned to any type of information resources without fundamental restrictions (Dublin Core, 2003).

Elements for this standard according to (Dublin Core, 2003) are described in Appendix 1.

Dublin Core standard is simple and its elements overlap with IEEE LOM.

2.1.1.3. Metadata for Learning Resources (MLR):

The MLR is developed by ISO (the International Organization for Standardization) and IEC (the International Electro-technical Commission). Mainly this standard has two purposes: 1) describe learning resource using standard-based approach to identify metadata elements needed to describe a learning resource, and 2) search, discovery, acquisition, evaluation, and use learning resources by learners, instructors, and automated applications. This standard takes into consideration diversity of culture and linguistics contexts (ISO/IEC Framework, 2011).

This standard uses a common set of attributes to specify essential characteristics of data elements. This allows the user of this standard to define data element specifications. Each data element specification has the attributes mentioned in Table 2 that give the flexibility to the user of this standard to define metadata elements.

Table 2: Data element specifications in ISO/IEC (ISO/IEC Framework, 2011).

No.	Attribute Name	Attribute Definition
1	Identifier	Data element specification identifier
2	Property name	Data element name
3	Definition	Data element definition
4	Linguistic indicator	Data element linguistic Indicator
5	Domain	Data element domain
6	Range	Data element range
7	Content value rules	
8	Refines	
9	Examples	
10	Notes	

In (ISO/IEC - Basic application profile, 2011), this standard defines metadata elements and their attributes to describe learning resource. These elements have been prepared using best practices based on an international survey, and reflect actual practices of IEEE LOM and DCMES.

Data element specifications from other parts of the MLR standard are: title, creator, subject, publisher, contributor, relation, coverage, and rights. Also data element specifications (locally defined) are: date, description, format, identifier, language, source, and type (ISO/IEC - Basic application profile, 2011).

2.1.2. Learning Object Repositories

Learning object repositories are databases used to store learning objects. These repositories may contain learning objects and learning objects metadata, or contain only learning objects metadata. The most common form for learning object repositories is a centralized model where all data stored in a single server. Distributed servers model for data also used in some repositories (Verbert, 2008).

Examples of learning object repositories are:

2.1.2.1. ARIADNE:

ARIADNE is a non-profit association working in learning technologies. ARIADNE worked for European stakeholders in the beginning, and now expanding into worldwide members (ARIADNE-A, 2012).

ARIADNE aims to carry out research to improve creation, sharing, and reuse of knowledge through technology. In addition to developing and deploying methodologies and software that provide flexible, effective, and efficient access to knowledge. Also

ARIADNE project supports educational and research communities (ARIADNE-A, 2012).

ARIADNE infrastructure consists of three layers: storage layer that stores learning objects and metadata in several databases, middle layer that manages learning objects and metadata through a set of services, and toolset layer that allows end users to access learning objects by hide low layers complexity (ARIADNE-B, 2012).

Figure 1 shows layers of ARIADNE infrastructure.

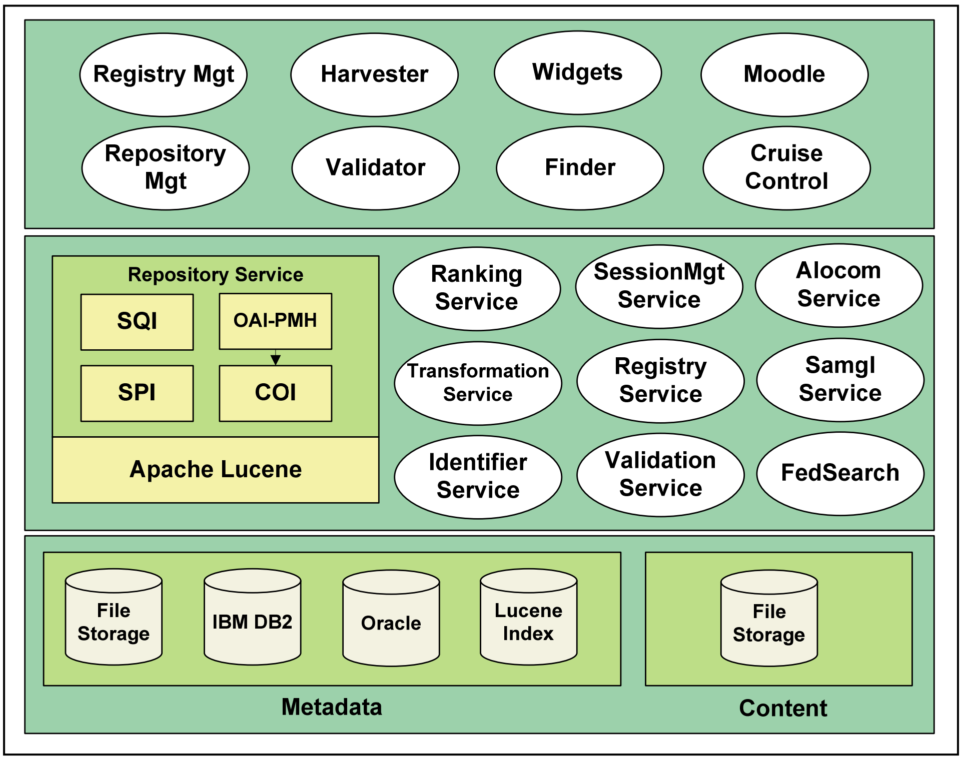


Figure 1: ARIADNE infrastructure (ARIADNE-B, 2012).

ARIADNE uses the IEEE LTSC LOM to describe their metadata, also they support other standards like Dublin Core and ISO/IEC MLR using automatic transformation between metadata formats (ARIADNE-C, 2012).

ARIADNE provides a tool (ARIADNE Finder) to search on learning materials online. The repository has 823,337 learning resources in time of writing this chapter (ARIADNE-D, 2012).

2.1.2.2. Connexions:

This repository calls learning objects as modules. Modules are defined as educational material made of small knowledge chunks. This repository is free and open for use by authors, instructors, and learners (Connexions-A, 2012).

Architecture for this system is divided into four major components: content, repository, editing environment, and lensing systems (Connexions-B, 2012). Two types of content can be published: learning objects and full materials like textbooks. Both of them are saved and managed in XML languages. Repository used to store and access the content. Content includes open educational resources for all ages and from the globe. Editing environment has a web portal for creation and management of materials. Users can highlight content using lenses system to allow experts checking content in order to guarantee user-driven quality control.

Connexions provides tool to search on learning materials online. The repository has 20,913 modules in time of writing this chapter (Connexions-C, 2012).

2.1.2.3. MERLOT:

MERLOT (Multimedia Educational Resource for Learning and Online Teaching) is free and open online community, and USA initiative for learning resources. MERLOT mainly designed for higher education faculties (MERLOT, 2012).

MERLOT is a centralized learning objects repository containing metadata only with links to external locations. MERLOT uses its own format for metadata to describe learning objects (Cafolla, 2002).

MERLOT provides a tool to search on learning materials online. The repository has 38,446 materials in time of writing this chapter (MERLOT, 2012).

2.1.2.4. OER Commons:

OER Commons launched in February 2007 to support building and reusing open educational resources (OER). This project offers many functions such as curriculum alignment, quality evaluation, social bookmarking, tagging, rating, and reviewing. The network has more than 500 international partners, and this allowed it to have 42,000 open educational resources with high quality and from around the world (OER Commons, 2007).

Resources in OER Commons project are organized using different categories to find them easy. Some categories can be defined by resource creator, while administrator is responsible about define others. Resource categories are: subject areas, grade levels, material types, media formats, courses, and libraries. Users can perform simple search, advanced search, and browse list of materials (OER Commons, 2007).

2.1.2.5. CAREO:

CAREO (Campus Alberta Repository of Educational Objects) is Canadian repository contains more than 4,000 learning objects. This repository has learning objects and also links to external objects. Membership is free and open to anyone (CAREO, 2012).

CAREO repository is integrated with ALOHA (Advanced Learning Object Hub Application) metadata server, and provides additional functions to users. XML-RPC (SOAP) technology used as communication channel between components, and JDBC used to communicate with internal databases. First security level managed by ALOHA server, and second level by CAREO using grant/revoke functions to/from users. Educational objects available and can be located from internet (Mattson, Norman, & Purdy, 2002).

CAREO uses IEEE LOM standard for metadata to describe learning objects (Verbert, 2008).

2.1.2.6. Edutella:

Edutella is peer-to-peer network to search semantic web metadata (Edutella, 2012).

Edutella has two steps: find information and how it can represent the semantic web, and provide translation between system presentation and semantic web presentation (Edutella, 2012).

2.1.2.7. Maricopa Learning Exchange:

Maricopa Learning Exchange (MLX) is an electronic warehouse of ideas, examples, and resources to support student learning at the Maricopa Community Colleges (Maricopa, 2012).

Maricopa provides tool to search on learning materials online. The repository has 1,825 learning items in time of writing this chapter (Maricopa, 2012).

2.1.3. Learning Object Life Cycle

Understanding the learning object life cycle is important for any research in learning technology domain. According to (Collis & Strijker, 2004), learning object life cycle consists of six stages: obtaining or creating, labelling, offering, selecting, using, and retaining.

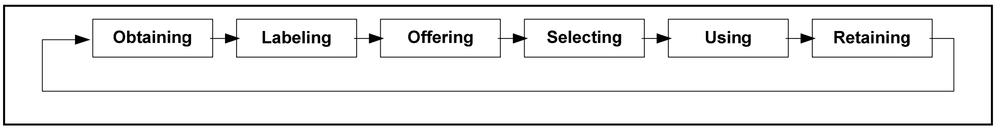


Figure 2: Learning Object Life Cycle Stages. (Collis & Strijker, 2004)

Obtaining is the first stage in life cycle that obtains or creates learning object. Learning material will be added using digital form. In some organizations, they have templates for all material types to guarantee consistency and quality.

Labelling is similar to process of classifying book and label it by librarian. Gathering metadata about learning object is important in this stage, even to be entered manually by person who creating the object, or to be generated automatically. Profile to predefine set of data needed to be filled is good practice. This stage is very important to support next stages.

Learning object provider can offer material for use by others. Offering stage can be done for users inside same organization, for fee and charge, or to be free for all users in internet.

First three stages (obtaining, labelling, and offering) are related to learning object provider, but other stages (selecting, using, and retaining) are related to the user.

User needs to select learning object from repository to use it. In selecting stage user will decide if object is usable and has desired material, get it in short time and by effective way. Develop search tool to achieve this stage is challenge, and main research area for this thesis.

In using stage, learning object can be used as it is without any change, or can be modified and adapted for its new environment. Learning object can be combined with other objects to design new learning material.

After using the learning object for some period, it becomes out of date, and needs to be deleted or revised. Retaining stage is responsible about archive old learning objects.

2.1.4. Issues and Challenges

Learning technology is a new domain in comparison to other domains in information technology, so there are many issues and challenges in this domain. Many areas still open for research. Main challenges are: publishing learning objects, searching learning objects, and learning objects interoperability. This thesis focuses on searching of learning objects challenge, but other challenges are out of scope of this thesis.

2.1.4.1. Publishing Learning Object:

Publishing learning objects correctly is necessary to support search and reuse of learning objects. According to (Ternier, Duval, & Neven, 2003), users of learning materials are not motivated to publish learning objects that they have created, because there is no immediate benefit for it.

Learning objects authors found it tedious to upload learning objects, and not willing to spend extra time and effort to add metadata to their learning objects (Duval & Hodgins, 2003). Critical mass learning objects will be lost and not available for reuse (Ternier, 2008).

2.1.4.2. Searching Learning Object:

Searching learning object is very important for high reuse of learning materials, which is the main aim of learning object repositories. If search tool is not effective, then probably learning objects will not be found or reused, and the repository will not give useful information for its users.

Also context of usage of learning objects is important because same element may be perceived in different ways by different communities (Najjar, 2008).

2.1.4.3. Learning Object Interoperability:

Interoperability defined as "the ability for objects from multiple and unknown or unplanned sources, to work or operate technically when put together with other objects" (Duval & Hodgins, 2003).

There are four kinds of interoperability in learning technology: between learning objects, between learning management systems, between learning object repositories, and between metadata schemas (Duval & Hodgins, 2003).

Interoperability classified by (Decker, et al., 2000) into two classes: syntactic interoperability (software applications must be able to read and exchange data with different systems), and semantic interoperability (understand the meaning of data by different systems).

Interoperability can be obtained using platform independent communication layer protocol to allow all learning repositories to communicate and share resources. An example for this implementation is eduSource Communication Layer (ECL) protocol which is compatible with IMS Digital Repository Interoperability (DRI) specifications. ECL protocol supports four functions: search/expose, submit/store, gather/expose, and request/deliver. (Eap, Hatala, & Richards, 2004)

2.1.5. Attention Metadata

Attention metadata describes what the user like, dislike, read, write, discuss, listen, and what things he pays attention to. Many kinds of content can be used like web site, movie, music, text, chat, email, etc. All this information will be saved in user preferences (Najjar, Wolpers, & Duval, 2006-A, 2006).

Actual interest that users gave to learning objects can be saved into user profiles, and used later to improve searching learning objects (Najjar, 2008).

Suitable and relevant learning object for a user depends on the context, and maybe the search result for two users using same query will be different. For example, the search result may depend on instructor profile and his/her interests, course type and level, and profiles of students and their needs (Najjar, 2008).

Attention metadata can be described using an appropriate schema which is different from the schema for learning object metadata. Examples of attention metadata schemas are (Najjar, 2008): AttentionXML (Najjar, Wolpers, & Duval, 2006-A, 2006), Attention Profiling Markup Language (APML) (APML-B, 2012), and the Contextualized Attention Metadata schema (CAMs) (Najjar, 2008).

Real use and implementation of attention metadata must be done using a framework and models that will be responsible of all functions. Related work in this domain has been presented in (Butoianu, Verbert, Duval, & Broisin, 2010): TaskTracer, Swish, Contextualized Attention Metadata framework (CAMf), Dyonipos, Context Modeling Language, WildCAT, and jNotify.

2.1.5.1. Attention Metadata Schemas:

Schema for attention metadata will allow multiple applications and services to share data easily (Butoianu, Verbert, Duval, & Broisin, 2010).

AttentionXML:

AttentionXML schema specifications were introduced to track RSS feeds and Blogs. This schema has basic elements such as (title, identifier, mime type, etag, last updated, last read, duration, followed links, rel/vote link, tags, and read time). This standard is missing important elements about user activities. Example of missing elements are types of action (view or edit), and information about application (browser, mail client, etc) (Najjar, Wolpers, & Duval, 2006-A, 2006).

AttentionXML is not designed to be used for interaction with learning objects, and doesn't allow capture activities like searching and downloading documents from internet, read and write documents, listen to music, chat, send email, etc. (Najjar, 2008)

In order to be able to collect rich and detailed attention metadata, this schema has been extended to Contextualized Attention Metadata (Najjar, Wolpers, & Duval, 2006-A, 2006).

Attention Profiling Markup Language (APML):

Attention Profiling Markup Language (APML) allows the user to share his personal attention profile with other users and applications. Main idea for this standard is to provide portable file format to include all forms of attention metadata with ranked interests (APML-A, 2012).

APML proposes their feature as turn normal user actions to Attention Metadata, then export attention metadata to central location, and finally create Attention Profile for users.

Figure 3 shows APML feature.

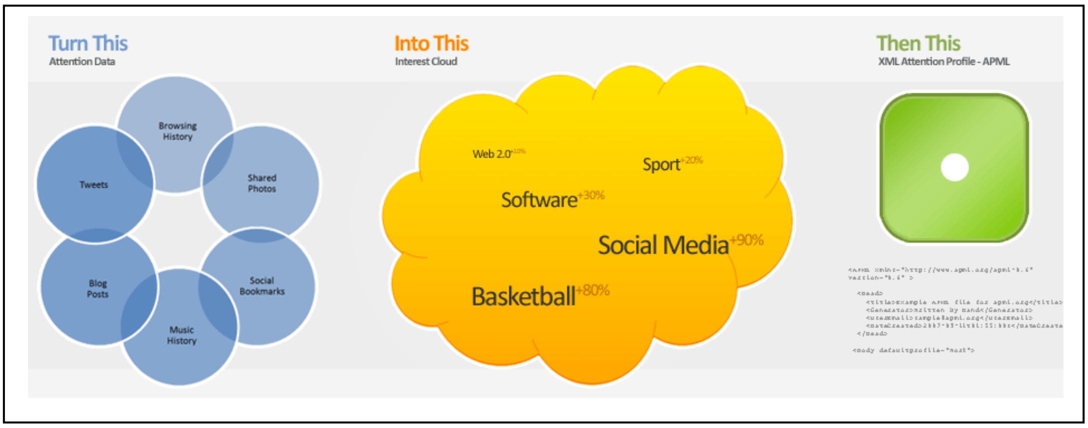


Figure 3: APML feature (APML-A, 2012).

APML allows implicit interests concepts (calculated automatically using algorithms), and explicit interests that can be provided manually by the user (Najjar, 2008). APML collects user interests only, while other schemas like CAM can relate user interests with digital content, and events on objects (Najjar, 2008).

Contextualized Attention Metadata schema (CAMs):

CAM schema is an extension for AttentionXML to allow capture of more observations, and describes the context in more details. CAM captures the type of event for each item with detailed properties. This allows capturing observations from any kind of tool (Najjar, 2008).

CAM collects multiple attentions for the same item. If the user accessed one item in many systems, then this attention metadata will be collected. Manual ranking, annotations, and tags entered by the user will be collected as well (Najjar, 2008).

2.1.5.2. Attention Metadata Frameworks:

Software architecture for context management system has been presented by (Henricksen & Indulska, 2005). The architecture consists of six loosely coupled layers. These layers are: context gathering, context reception, context management, query, adaptation, and application. Context gathering layer has sensors to collect context information and process it through interpreters and aggregators. Transfer gathered information to management layer and direct queries from management layer to gathering layer will be done by context reception layer. Context management layer is responsible about context models and repository. Query layer provides applications with interface to query management layer. Adaptation layer manages common repositories. Application layer has tools and APIs to support programming.

There are many frameworks to capture context information presented in (Butoianu, Verbert, Duval, & Broisin, 2010):

TaskTracer:

TaskTracer allows locating, discovering, and reusing past processes that workers completed successfully. It uses Publisher-Subscriber architecture to collect and process events. TaskTracer uses COM plug-in in MS Office applications, windows CBT hook, and .NET to collect data from desktop resources. This framework uses relational database for management layer (repository), and SQL for query layer (Butoianu, Verbert, Duval, & Broisin, 2010).

Swish:

Swish detects automatically users' tasks by monitoring desktop windows and relation between them. Swish is not restricted to predefined applications from where it can collect information, because it can monitor any Windows application. Swish uses a hook into Windows OS to listen for events produced by each window. This framework uses relational database for management layer (repository), and SQL for query layer (Butoianu, Verbert, Duval, & Broisin, 2010).

Dyonipos:

Dyonipos identifies user's task, and provides him/her with information from both personal and organization data. Context in this framework is composed of five dimensions: action, resource, user, application, and information need. Dyonipos uses two sources to collect data (sensors): mainstream applications (MS Office, internet browsers, email clients), and sensors for operating system (file system, clipboard, network stream). This framework uses RDF (Resource Description Framework) (RDF, 2012) for management layer (repository), and RDF query for query layer (Butoianu, Verbert, Duval, & Broisin, 2010).

Context Modeling Language (CML):

CML is a graphical context modelling approach. It explains information type, information classification, metadata quality, and dependencies between types of information. This framework uses PostgreSQL for management layer (repository), and SQL for query layer (Butoianu, Verbert, Duval, & Broisin, 2010).

WildCAT:

WildCAT is Java framework to allow creating context-aware applications by providing common interface to ease integration between heterogeneous information. Model schema for this framework contains four types: context, context domain, resource, and attribute. This general schema allows framework to support different levels of extensions, and customize implementation.

Contextualized Attention Metadata framework (CAMf)

Figure 4 shows CAM framework.

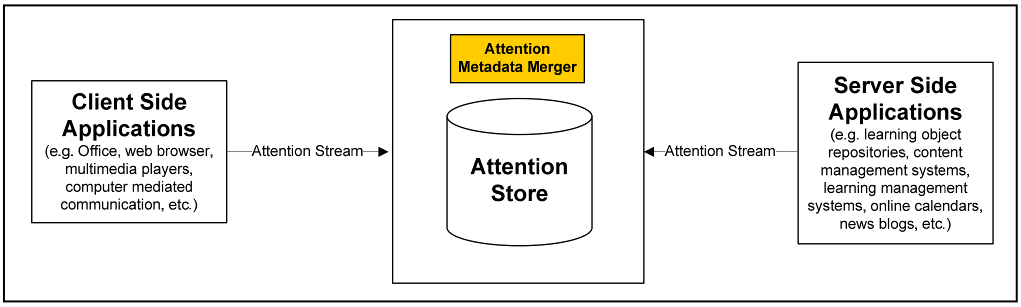


Figure 4: CAM framework (Wolpers, Najjar, Verbert, & Duval, 2007).

CAM is designed to collect observations from all applications and store it in CAM format. CAM identifies source and location of observation and its social type to relate observation with one user only or with other users. CAM uses desktop tools to load data from applications used on daily basis by most users such as: MS Office, internet browsers, multimedia players, messaging and chat tools. Also CAM depends on logs available in servers such as web search engines, learning object repositories, and online games (Wolpers, Najjar, Verbert, & Duval, 2007).

2.2 Data Mining

Data mining became important topic in information technology due to the huge amount of data, and the need for useful information and knowledge. Data mining can be used in many applications to extract needed knowledge such as market analysis, fraud detection, customer retention, production control, science exploration, and usage data analysis. (Han & Kamber, 2006)

Data mining is defined as "extracting or mining knowledge from large amounts of data" (Han & Kamber, 2006). Data mining consists of seven steps (Han & Kamber, 2006): data cleaning, data integration, data selection, data transformation, data mining, pattern evaluation / model testing, knowledge presentation, decisions and use of discovered knowledge. Removing noise and inconsistent data will be done in data cleaning step, while combine data from multiple sources will be done in data integration step. After that, retrieve data relevant to the analysis task from the database will be done in data selection. Data transformation is to transform data or consolidate it into appropriate forms for mining. After all previous preparation steps, data mining step which is an essential process to extract data patterns and build models using intelligent methods. After build the models using data mining, pattern evaluation / model testing will be used to identify the correctness of patterns and models. Finally, knowledge presentation is used to visualize the mined knowledge to the user. Figure 5 shows architecture of a typical data mining system including the steps mentioned above.

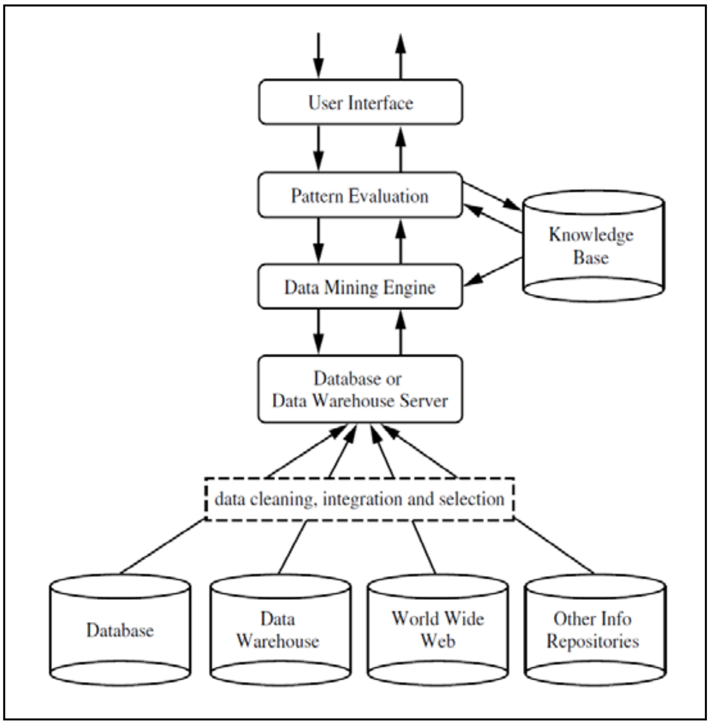


Figure 5: Architecture of a typical data mining system (Han & Kamber, 2006, p. 8).

2.2.1. Data Mining Techniques

Data mining techniques are used to find patterns and build models in data mining tasks. Data mining techniques can be classified into two categories: descriptive and predictive. Descriptive techniques characterize properties of the data, but predictive techniques perform tasks on current data to make predictions. (Han & Kamber, 2006)

Data mining techniques are described in the coming subsections (Han & Kamber, 2006):

2.2.1.1. Concept/Class Description:

Associate data with classes or concepts, and this can be done using data characterization under target class, or using data discrimination by comparing target class with other contrasting classes.

2.2.1.2. Mining Frequent Patterns, Associations, and Correlations:

Find patterns that occur frequently in data. There are three types of frequent patterns: itemsets, subsequences, and substructures. Frequent itemsets are items that appear together in a transactional data, such as bought books together from a bookstore. Frequent subsequence patterns are actions occur in same sequence, such as buying computer, then digital camera, then memory card. Different structural forms can be used such as graphs and trees, which may be combined with itemsets and subsequences patterns. (Han & Kamber, 2006)

2.2.1.3. Classification and Prediction:

Classification uses models to find class for unknown data. The technique uses training data with known class to build model. Classification model can be built using if-then rules, decision tree, or neural network. (Han & Kamber, 2006)

2.2.1.4. Cluster Analysis:

Unlike classification and prediction, cluster analysis works without known classes. This technique groups objects and find similarity to form clusters. Each cluster will have set of objects that will be used to form rules. (Hall, Frank, Holmes, Pfahringer, Reutemann, & H, 2009)

2.2.1.5. Outlier Analysis:

Outliers are data objects that do not comply with general behaviour of the data. Outlier detection is very important in some applications such as fraud detection. Outliers can be detected using statistical tests by measure distance, or using deviation-based methods. (Han & Kamber, 2006)

2.2.2. Data Mining Tools

There are many data mining tools available with different functionalities. These tools can be categorized into two types: free open-source and commercial tools.

Examples of free open-source data mining tools are: Weka, RapidMiner, ELKI, jHelpWork, KNIME, Orange, and UIMA.

Weka (Waikato Environment for Knowledge Analysis) (Hall, Frank, Holmes, Pfahringer, Reutemann, & H, 2009) provides collection of machine learning algorithms that can be applied directly to dataset or called from Java code. The tool has pre-processing, classification, regression, clustering, association rules, and visualization.

RapidMiner (RapidMiner Overview, 2013) is available as stand-alone application or be integrated with other applications as data mining engine. It provides data integration, analytical ETL (Extract, Transform and Load), data analysis, and reporting. The tool can be managed using graphical user interface to design the process. The tool supports data loading, data transformation, data modelling, and data visualization methods.

ELKI (Environment for Developing KDD-Applications Supported by Index-Structures) (ELKI Background, 2013) is resulted from data mining research. It provides index-structures that improved data mining tasks. Data mining algorithms and data management are separated in this tool, and this improved the performance and evaluation.

jHelpWork (jHepWork Home, 2013) is a tool designed for scientists, engineers, and students to perform scientific computation, data analysis, and data visualization. This tool supports coherent interface using scripting concept. jHelpWork can be used by many scripting language such as Jython, BeanShell, and native Java. In addition to that, Matlab/Octave can be used for symbolic calculations.

KNIME (KNIME, 2013) is a graphical workbench for data access, data transformation, initial investigation, predictive analytics, visualization, and reporting. Also this tool includes additional functions such as shared repositories, authentication, remote execution, scheduling, SOA integration, and user interface.

Orange (Orange Data Mining, 2013) performs data mining through visual programming or Python scripting. It has tools for bioinformatics and text mining.

UIMA (Apache UIMA, 2013) or Unstructured Information Management applications are to analyse large volumes of unstructured information to discover knowledge.

Examples of commercial data mining tools are: IBM SPSS Modeler, Microsoft SQL Server Analysis Services SSAS, Oracle Data Mining (ODM), and MineSet.

IBM SPSS Modeler (SPSS Modeler, 2013) is a data mining workbench to build predictive models without programming. This tool provides automated modelling, text analytics, and entity analytics.

Microsoft SQL Server Analysis Services SSAS (SQL Server - Analysis Services, 2013) allows user to build analytic solutions for predictive analysis and interactive exploration of aggregated data. The system architecture allows integration with other Microsoft products such as Office and SharePoint.

Oracle Data Mining (ODM) (Oracle Data Mining, 2013) is part of Oracle Database, and available as native SQL. ODM provides predictive models, data analysis, build and evaluate models, save and share analytical methodologies. Techniques supported by this tool are: classification, regression, attribute importance, anomaly detections, clustering, association, and feature extraction.

MineSet (MineSet, 2013) is a tool for data mining and real-time 3D visualization. It supports searching, sorting, and filtering.

2.2.3. Weka Library Algorithms

Weka (Weka 3: Data Mining Software in Java, 2013) is open software under GNU General Public License. It has been developed by University of Waikato in New Zealand. The tool provides user interface to load datasets and apply data mining techniques, also it can be called from Java code. This library has been chosen to be used in this thesis for three reasons. First reason is that library supports Java and can be called using programming code and this allows us to develop application that will create user profiles automatically from the application. Second reason is that library is an open source code and can be used without any commercial limitations. Third reason is that the library is part of research done by a university in New Zealand, so many people from all around the world participated in this library and added many data mining algorithms to it (classifications, clustering, and association rules).

The library supports pre-processing, classification, regression, clustering, association rules, and visualization.

Pre-processing has many filters applied on attribute of instance level. Multiple filters can be applied on data before use data mining algorithms.

The library supports the following classification algorithms:

DMNB Text (Discriminative Multinomial Naïve Bayes) algorithm tried to change generative parameter of learning method into discriminative element. This algorithm computes frequencies from data discriminatively, and then estimates parameters based on computed frequencies (Su, Zhang, Ling, & Matwin, 2008).

Naïve Bayes (John & Langley, 1995) algorithm uses estimator classes. Based on analysis of the training data, numeric estimator precision values will be chosen. The classifier is not an Updateable Classifier, and needs training data to be initialized.

Naïve Bayes Updateable (John & Langley, 1995) algorithm is the updateable version of Naïve Bayes. Classifier uses default precision of 0.1 for numeric attributes when build classifier is called with zero training instances.

Naïve Bayes Multinomial (Mccallum & Nigam, 1998) is simple naive Bayes algorithm and doesn't capture number of times a word occurs in a document, but multinomial algorithms consider number of occurrences of each word, and calculate probability accordingly.

Naïve Bayes Multinomial Updateable (Mccallum & Nigam, 1998) algorithm is the updateable version of Naïve Bayes Multinomial. Classifier uses default precision of 0.1 for numeric attributes when build classifier is called with zero training instances.

Naïve Bayes Multinomial Text (Mccallum & Nigam, 1998) algorithm operates directly on String attributes, and no need to convert text data to String vector like other algorithms. It works with String attributes only, and other types are ignored during build classifier.

Bayes Net (Bayesian Network, 2013) algorithm uses various local search algorithms such as (K2 and B), and global search algorithms such as (simulated annealing and tabu search). There are local score metrics implemented such as (Bayes, BDe, MDL, entropy, and AIC), and global score metrics such as (leave one out cv, k-fold cv, and cumulative cv). Direct estimates and Bayesian model averaging can be used for parameter estimation.

Complement Naïve Bayes (Rennie, Shih, Teevan, & Karger, 2003) algorithm asks for complement class least fit specific document. Complement class is agglomeration for all other classes. Algorithm can work using one-label-per-document by use complement part only, or multi-label by compare versus all-document class.

SMO (Sequential Minimal Optimization) (Platt, 1998) (Keerthi, Shevade, Bhattacharyya, & Murthy, 2001) (Hastie & Tibshirani, 1998) algorithm replaces all missing values and transforms nominal attributes into binary, and normalizes all attributes. Option that fits logistic regression models to the outputs of the support vector machine can be used to obtain proper probability estimates. Predicted probabilities are coupled using Hastie and Tibshirani's pairwise coupling method in the multi-class case.

LibSVM (Library for Support Vector Machines) (Chang & Lin, 2001) algorithm supports vector machine. This algorithm supports classification (C-SVC, nu-SVC), regression (epsilon-SVR, nu-SVR), and distribution estimation (one-class SVM). Library implemented in C++ and Java. Java library supported by what (EL-Manzalawy, 2013).

OneR (Holte, 1993) is a simple algorithm uses 1R classifier which is the minimum error attribute for prediction.

Multilayer Perceptron (ANN) (Multilayer perceptron, 2013) is an artificial neural network algorithm that maps input data into required output. This algorithm uses multiple layers of nodes in a directed graph. Classifier uses back propagation supervised learning technique to train algorithm.

MLP Classifier (ANN) (Multi Layer Perceptron) (Weka 3: Data Mining Software in Java, 2013) algorithm uses multilayer perceptron with one hidden layer using optimization class by minimizing the squared error. An approximate version of the logistic function is used as the activation function to improve speed. Parallel calculation can be used when multiple CPU cores are present. Nominal attributes are transferred to binary using filter.

RBF Classifier (Radial Basis Function Networks Classifier) (Weka 3: Data Mining Software in Java, 2013) algorithm uses radial basis function networks for classification. A fully supervised manner using optimization class is used in algorithm to train classifier. All attributes are normalized into the (0 and 1) scale. It is possible to learn attribute the weights for the distance function, also it is possible to use conjugate gradient descent. Parallel calculation of squared error and gradient is available when multiple CPU cores are present. Nominal attributes are transferred to binary using filter.

IBk (Instance-based K-nearest neighbours classifier) (Aha & Kibler, 1991) is a K-nearest neighbours classifier and can select appropriate value of K based on cross-validation. In addition to that, it can do distance weighting. This algorithm is lazy which means that it will not build model until need to use it.

KStar (Cleary & Trigg, 1995) or K* is an instance-based classifier. It differs from other instance-based learners in its usage for entropy-based distance function. This algorithm is laze as well.

LBR (Lazy Bayesian Rules) (Zheng & Webb, 2000) algorithm provides a simple and effective approach to classifier learning. This algorithm is lazy as well.

LWL (Locally Weighted Learning) (Frank, Hall, & Pfahringer, 2003) (Atkeson, Moore, & Schaal, 1996) algorithm is an instance-based algorithm to assign instance weights, and then used by a handler. It can provide classification using naive bayes or regression using linear regression.

Citation kNN (Wang, Zucker, & Daniel, 2000) algorithm works with multiple-instance problem to find kNN. This algorithm is laze as well.

Weka library supports clustering algorithms such as: Simple K Means, Cascade Simple K Means, CLOPE, Cobweb, DBScan, Simple EM (Expectation Maximisation), Farthest First, Hierarchical Clusterer, LVQ (Learning Vector Quantization), Make Density Based Clusterer, OPTICS (Ordering Points To Indentify the Clustering Structure), Self Organizing Map, and Filtered Clusterer.

Simple K Means (Arthur & Vassilvitskii, 2007) has two methods to calculate distance: Euclidean distance or the Manhattan distance. In case of Manhattan distance, centroids are computed as component-wise median, but in case of Euclidean, centroids are the mean.

Cascade Simple K Means (Calinski & Harabasz, 1974) algorithm selects the best k according to calinski-harabasz criterion".

CLOPE (Yang, Guan, & You, 2002) algorithm works with transactional data categorized by large volume.

Cobweb (Fisher, 1987) (Gennari, Langley, & Fisher, 1990) "algorithm compares the best host, adding a new leaf, merging the two best hosts, and splitting the best host when considering where to place a new instance".

DBScan (Ester, Kriegel, Sander, & Xu, 1996) is a density-based algorithm for discovering clusters in large spatial databases with noise.

Simple EM (Expectation Maximisation) (Weka 3: Data Mining Software in Java, 2013) algorithm will assign a probability distribution to each instance, and this will indicate the probability of belonging to each of the clusters. EM can automatically decide how many clusters to create by cross validation, or user may specify number of clusters manually.

Farthest First (Hochbaum & Shmoys, 1985) (Dasgupta, 2002) is a hierarchical algorithm uses Farthest First technique. It is approximation algorithm for the k-centre problem that tries to find optimal k-clustering under cost function and maximum cluster radius.

Hierarchical Clusterer (Weka 3: Data Mining Software in Java, 2013) algorithm “implements a number of classic agglomorative (i.e. bottom up) hierarchical clustering methods”.

LVQ (Learning Vector Quantization) (Weka 3: Data Mining Software in Java, 2013) algorithm “implements Learning Vector Quantization algorithm for unsupervised clustering”.

Make Density Based Clusterer (Weka 3: Data Mining Software in Java, 2013) algorithm makes clusters with ability to return a distribution and density. It is good for normal and discrete distributions.

OPTICS (Ordering Points To Indentify the Clustering Structure) (Ankerst, Breunig, Kriegel, & Sander, 1999) algorithm computes an augmented cluster-ordering of the database objects.

Self Organizing Map (Weka 3: Data Mining Software in Java, 2013) clusterer “implements Kohonen's Self Organizing Map algorithm for unsupervised clustering”.

Filtered Clusterer (Weka 3: Data Mining Software in Java, 2013) algorithm by pass data through filter before generate clusters for them.

Weka library supports association rules algorithms such as: Apriori, FPGrowth, GSP (Generalized Sequential Patterns), HotSpot, Predictive Apriori, Tertius, and Filtered Associator.

Apriori (Agrawal & Srikant, 1994) (Liu, Hsu, & Ma, 1998) algorithm implements an Apriori-type. It reduces the minimum support iteratively until finds the required number of rules with the given minimum confidence. The algorithm can mine class association rules.

FPGrowth (Han, Pei, & Yin, 2000) algorithm implements FP-growth algorithm for finding large item sets without generate list of candidates. It reduces the minimum support iteratively until finds the required number of rules with the given minimum metric.

GSP (Generalized Sequential Patterns) (Srikant & Agrawal, 1996) algorithm implements GSP algorithm to discover sequential patterns. User can identify the distinct data sequences by determine respective option. The results can be restricted by specify one or more attributes in each element of a sequence.

HotSpot (Weka 3: Data Mining Software in Java, 2013) algorithm displays set of rules in a tree-like structure, and learns maximize and minimize a target value of interest.

Predictive Apriori (Scheffer, 2001) algorithm searches by increase support threshold to find best 'n' rules. Algorithm adds a rule to the output of the 'n' best rules if the expected predictive accuracy of this rule is among the 'n' best rules.

Tertius (Flach & Lachiche, 1999) algorithm finds rules according to confirmation measure (Tertius-type algorithm).

Filtered Associator (Weka 3: Data Mining Software in Java, 2013) algorithm passes data through filter before find association rules.

2.3 Contribution and Research Approach

Many standards have been developed for metadata to describe learning objects. Main purpose of metadata is to support indexation and search in learning object repositories. The repository provides tools for users to search for learning objects. Effective search tool is important to serve users in efficient way (Najjar, Ternier, & Duval, 2004).

It has been found that during searching of learning objects, searchers select one or two metadata element to perform their queries, and they invest little time to build their search (Najjar, Ternier, & Duval, 2004). Search tools need to be effective even with this small information in query, and has ability to use data from additional resources to find appropriate learning objects.

Current search tools are still difficult to use and do not provide the user with relevant results. There is a need for a flexible technique to achieve high level of search results quality (Najjar, 2008)

Search tools may use interactive ways in the searching process. Some researches proposed a tool that can match users' tasks and contexts to provide user with recommendation for relevant learning objects. Users will not find normal results only, but also will receive enhanced results that took into consideration his context (Najjar, 2008).

In addition to basic learning object metadata, it is also important to collect how learning objects have been actually used across contexts. This data is called attention metadata (usage data) (Najjar, 2008).

The main contribution is the Consolidated Ranking and Recommendation Framework (CRRF). This framework has been designed, developed, and evaluated. Main component in the proposed framework is the User Profile. Attributes of user profile are listed, and this thesis explained how to find these attributes based on result of usage data analysis. Usage data is collected by Moodle2Cam module which is responsible for gathering of usage data from the Learning Management System (e.g. Moodle) and convert it to CAM format which is a standard format to store usage data. The data can be stored in XML files or database tables. The AutoProfileBuilder module is part of our proposed framework and analyses collected usage data to build user profile automatically. Other contributions are Ranking and Recommendation modules that have their formulas to calculate rank and recommendation learning objects based on usage data.

In addition to above contributions, there is a contribution in evaluation of data mining algorithms within learning technology domain.

2.4 Conclusion

This chapter gives background information about topics in the research field. Learning technology and data mining are the main two topics in this chapter. Background for learning technology including learning objects, standards of learning objects metadata, most popular learning object repositories and their structure, learning object lifecycle, issues and challenges in learning technology domain, and attention metadata standards are discussed. Background on data mining including information about data mining techniques, data mining tools, and Weka library is also covered in the chapter.

Chapter Three

Literature Review

This thesis based on two major topics: searching learning objects and data mining. Below sections provide literature review about these two topics.

3.1 Searching Learning Objects

Search within learning objects and find relevant result for user is important factor in success of learning objects repositories, so many researchers tried to find solution for this problem, and improve learning objects reusability.

Tools and techniques used to improve search learning objects have been classified by (Burke, 2007) and (Zapata, Menendez, Prieto, & Romero, 2011) into six approaches: content-based, collaborative filtering, demographic, knowledge-based, community-based, and hybrid.

Content-based method recommends learning objects that are similar to other learning objects that user liked in the past. Similarity between learning objects can be calculated using different techniques. Recommendation will use two sources: learning object metadata and past rating that user gave to them.

Collaborative filtering recommends learning objects that have been liked by similar users in past. Recommendation will use information from users rating only.
Demographic approach recommends learning objects based on demographic profile of the user. On the other hand, knowledge-based recommends learning objects based on specific domain knowledge and meet user needs and preferences. Community-based technique recommends learning objects based on user friends preferences. Approach that depends on a combination of the aforementioned techniques called as hybrid.

Collaborative filtering has two approaches (Verbert, Drachsler, Manouselis, Wolpers, Vuorikari, & Duval, 2011). First approach is user-based collaborative filtering that finds users with similar rating patterns with current user, and then uses rating done by those users to calculate prediction for current user. Second approach is item-based collaborative filtering that finds items with similar rating patterns for current item, and then use this information to calculate prediction for current user.

Researchers used different methodologies to rank and recommend learning objects with aim to improve learning objects reusability. One approach uses text similarity between keywords provided by user for search and metadata of learning objects (Ochoa, 2008). Another approach compares user profile with learning objects (Ochoa, 2008) and (Fouad Ibrahim, 2012). In (Al-Khalifa, 2008) and (Niemann et al., 2012), explore relations between learning objects themselves, and then recommend learning objects depends on user interests. In (Sampson & Papanikou, 2009), author proposed general framework for learning objects reusability as first step for further techniques. Authors in (El Helou, Salzmann, & Gillet, 2010) collected user interactions, and ranked learning objects depend on ranking algorithms. In (Zapata, Menendez, Prieto, & Romero, 2011), they used hybrid recommendation method (content comparison, collaborative, and demographic searches). Another research tried to collect attention metadata from multiple repositories, and compare it with peer users to provide recommendation service (Govaerts, El Helou, Erik, & Gillet, 2011). Another approach (Nešić, Gašević, Jazayeri, & Landoni, 2011) is based on build semantic information about learning objects and links between them to be used for search learning objects. Data mining techniques used as well to classify and rank learning objects (Batista, Pintado, Gil, Rodriguez, & Moreno, 2011). Concentrate on indexation and cataloguing for learning objects to improve search capabilities is the methodology in (Barcelos & Gluz, 2011). Ontology-based model is also used by providing many features to improve learning objects exploratory (Sridharan, Deng, & Corbitt, 2011). Personalization learning objects using user's past history and preferences used in (Sree Dharinya & Jayanthi, 2012). Other methodology used recommendation system which is flexible to changes in users' preferences (Chen & Huang, 2012). Finally, integrate Semantic Web technologies with LOM to develop learning objects finder is proposed in (Hsu, 2012).These methodologies explained in more details in below sections:

3.1.1. Metrics for Learning Objects (Learnometrics) (Ochoa, 2008)

According to (Ochoa, 2008), current learning object repositories use three strategies to provide ranking functionality:

Ranking based on human review:

Group of expert users review objects and grade them. Main advantage for this strategy is the explanation about decision behind the grade, and possible scenarios where it could be useful.

Disadvantage is the need for long working hours to complete this manual work, as a result, the object with low rank will appear higher than object not ranked yet regardless of its relevant. Second disadvantage is that ranking is relative to expert user doing the measurement, and cannot be adapted to other users easily. This approach is not scalable.

Ranking based on text similarity:

This approach depends on content to calculate rank value. Algorithms will be used to calculate similarity between terms in query and text in learning objects metadata. Advantage of this approach is its ability to calculate rank for all objects easily.

But one of disadvantages is that amount of metadata for learning objects normally low, and this will provide ranks with low performance. Another disadvantage, this approach will rank learning object depends on text, but will not provide information about relevance of the object itself. This approach doesn't guarantee quality.

Ranking based on user profile:

Compare information in user profile with object classification, and closer in value between user and object, will get higher rank and relevance.

One disadvantage of this approach is that user must explicitly select his interest before search. Another disadvantage is that ranking can be applied to objects that have been classified only. In addition to that, this approach doesn't integrate well in normal workflow of the user.

The research proposed different Ranking Metrics for learning objects. Topical relevance metric uses learning goal. Another metric is personal relevance that uses learning motivation, culture, language, education level, and accessibility. Last metric is situational relevance that uses learning setting, time of learning, place of learning, conditions, and limitations.

3.1.2. Ad Hoc Recommendation Engine (Al-Khalifa, 2008)

Ad Hoc Recommendation Engine (called Marifah) has been developed to support search for learning objects in Arabic language. This search engine is built using collaborative filtering scheme depends on user behaviour to recommend learning objects. There are two types for collaborative filtering: user-based and item-based. This engine used item-based scheme that depends on relations between objects without need to explore relation with users. Three factors participated in compute recommendation value: rate, number of downloads, and number of track backs. Recommendation engine model in Figure 6 shows overview for this algorithm. It supposes that user has list of LOs that he rated and saved in his history. Also for each LO saved in user history, algorithm will calculate LO similarity and recommendation value to find rank of LOs.

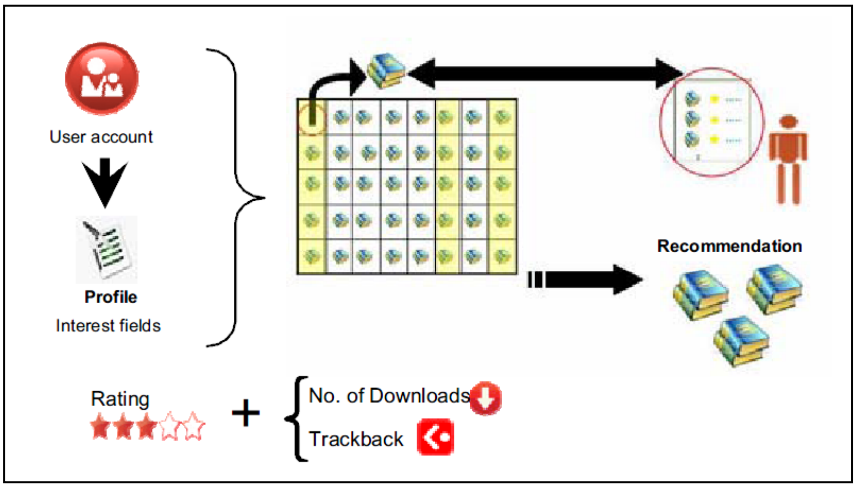


Figure 6: Recommendation Engine Model (Al-Khalifa, 2008).

3.1.3. Framework for LOs Reusability (Sampson & Papanikou, 2009)

This research discussed existing learning objects reusability frameworks, and suggested a new one to improve LOs reusability. This research proposed process and steps in the framework other than real algorithm or technique that can be used to improve search learning objects.

This framework proposed as essential step for development of real techniques and algorithms to improve learning objects reusability.

3.1.4. The 3A Recommender System (El Helou, Salzmann, & Gillet, 2010)

The system models user interactions in a heterogeneous graph, and then used ranking algorithms (personalized, contextual, and multi-relational) to rank actors, activities, and assets.

The 3A system ranks three entities: actors, assets, and activities. Actor can be regular user or agent. Asset is the resource that used by actor such as learning object. Activity is the operation done by actor on an asset.

Recommendation approach in this system consists of four steps. First step is graph construction where system forms heterogeneous and multi-relational directed graph taking actors, activities, and assets as nodes, and inter-relations between them as edges. Second step is context definition by collect information about user such as tags and activities. Third step is computation of importance that uses graph and context, in addition to PageRank (Page, Brin, Motwani, & Winograd, 1999) algorithm will be used to find importance computation for all nodes. Final step is extraction of ranked lists by extract list of actors, activities, and assets according to importance computation to be recommended.

3.1.5. Hybrid Recommender (DELPHOS) (Zapata et al., 2011)

This research proposed hybrid recommendation method for learning objects by use many types for filtering and techniques based on content comparison, collaborative, and demographic searches.

This method uses learning objects metadata, management activities on learning objects, and user profiles.

Sort and filter search result will provide user with recommendations and provide personalization for this task. Hybrid recommender implemented in DELPHOS system.

Architecture for proposed hybrid recommender method in Figure 7 shows that search engine for learning objects will be used to find pre-selection list by search within learning object metadata and compare it with provided search query, then this list will be filtered using four types of techniques (content similarity, learning object usage, evaluation for quality of learning object, and user profile similarity). Result of this method will be ranking for learning objects depends on predefined criteria and calculation scheme for all filtering techniques. Users will be able to select recommendation criteria that he needs to apply from four types mentioned above.

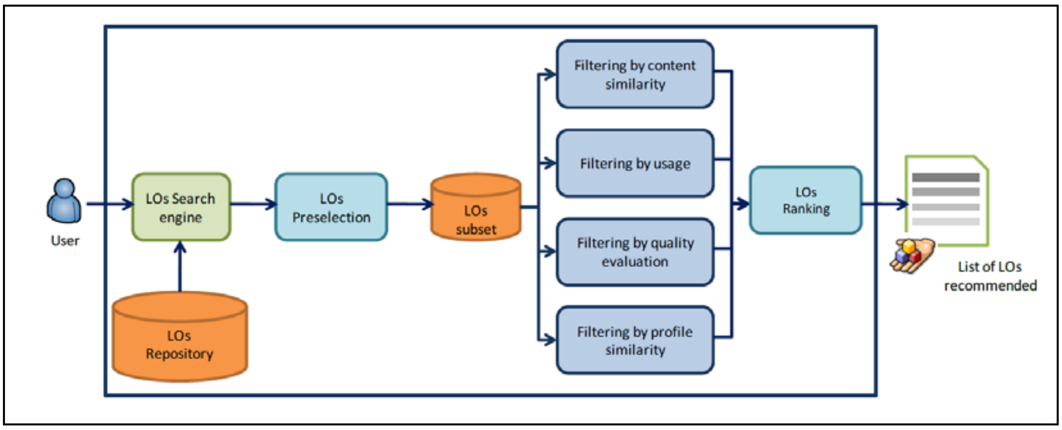


Figure 7: Architecture of hybrid recommender method (Zapata, Menendez, Prieto, & Romero, 2011).

Search result will be ranked from most relevance first, and also some statistics about each learning object like how many times object was saved, found in search result, and liked by other users. Also system will recommend some related objects for each learning object. User will have the ability to rate a learning object.

This method used learning objects usage only as source for attention metadata, and it didn't use attention metadata from other sources like social elements.

3.1.6. A Federated Search Widget (Govaerts, El Helou, Erik, & Gillet, 2011)

Small web application (widget) has been developed in this research to allow users in learning environments to search learning objects over multiple repositories.

Widget will be used to search for learning objects by collect information from social media sites and repositories. Then widget will use personalized social recommendation service to select relevant resources for user. Recommendation service will rank learning objects according to their global popularity and their popularity in user's social network.

Widget will capture attention metadata and user interactions such as liking, disliking, sharing, etc.

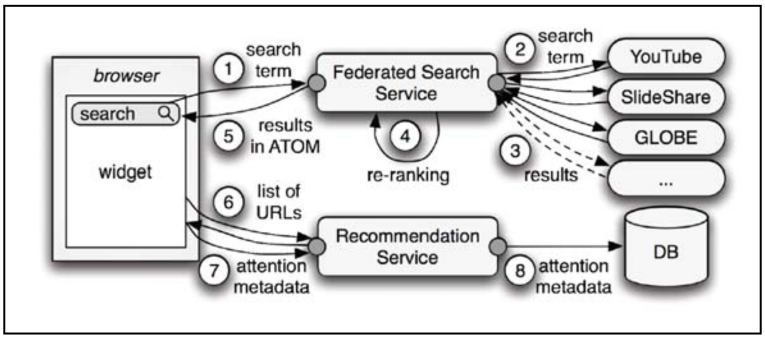


Figure 8: Architecture of federated search (Govaerts, El Helou, Erik, & Gillet, 2011).

Widget will get search request from user, then send this query to federated search service in client-server architecture. Federated search service will send query to multiple repositories and return search result to widget. This search result will not be shown to user yet, but instead will be sent to recommendation service that will use attention metadata to rank learning resources. Finally, ranked learning resources will be shown to user. Figure 8 shows architecture of federated search.

Main function in recommendation service is to rank relevance learning resources depends on attention metadata collected by peer users (like, dislike, share). Peer users can be determined from different social networks or users using same tool in same context. Background process is working to collect attention metadata and peers information.

Ranking algorithm inspired by PageRank algorithm (Page, Brin, Motwani, & Winograd, 1999) that used to rank web pages. Main idea in PageRank is that web pages are important when other important web pages have links to it. Federated search algorithm has random function that will work and rank resources according to their popularity for group of users.

3.1.7. Semantic Document Architecture (SDArch) (Nešić, Gašević, Jazayeri, & Landoni, 2011)

This research used semantic technologies and social networking to design new architecture (SDArch) in learning technologies to improve semantic search and personalization of learning content.

This framework works by build social network around some topic, then collects data from learning documents and form semantic integration and linking between them, and link documents in social network with different users. After that search local and shared collection of semantic links are applied, and finally navigate across semantic links to discover more data in interest.

SDArch is three-tier architecture (SOA) composed of the data layer, the service-oriented middleware, and user interface layer.

Data layer contains semantic document repository in RDF and binary data repositories. Also this layer contains text index to allow search over RDF data and binary data. This layer can be accessed via SPARQL endpoint using HTTP.

Service-oriented middleware is working as integration layer between user interface and data layer.

Presentation layer provides user interface layer for SDArch services. This layer is platform independent and can access middleware layer using SOAP over HTTP.

Figure 9 shows architecture of SDArch.

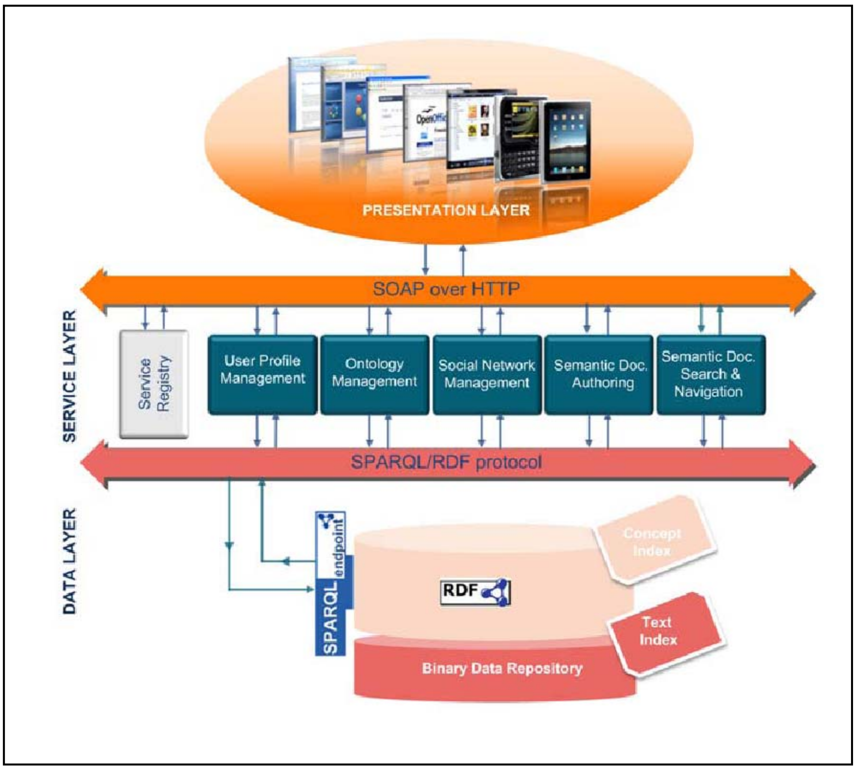


Figure 9: Architecture of SDArch (Nešić, Gašević, Jazayeri, & Landoni, 2011).

3.1.8. Multi-label Classification (Batista, Pintado, Gil, Rodriguez, & Moreno, 2011)

This research classifies and ranks learning objects using multi-label data mining techniques to help user select learning material by provide suitable choice. Learning process is supervised using multi-label classification and label ranking.

RAKEL was used for both multi-label classification and label ranking. This classification and ranking used in search of learning objects.

3.1.9. Agent-based Federated Search (AgCAT) (Barcelos & Gluz, 2011)

This research concentrated on cataloging and indexation process for learning objects to enhance process of finding appropriate object. They proposed an agent-based federated catalog of learning objects (AgCAT system). According to this research, correct cataloging will improve find learning objects by search engines, and incorrect cataloging will cause inefficacy in search process especially in distributed repositories.

Figure 10 shows architecture of AgCAT system.

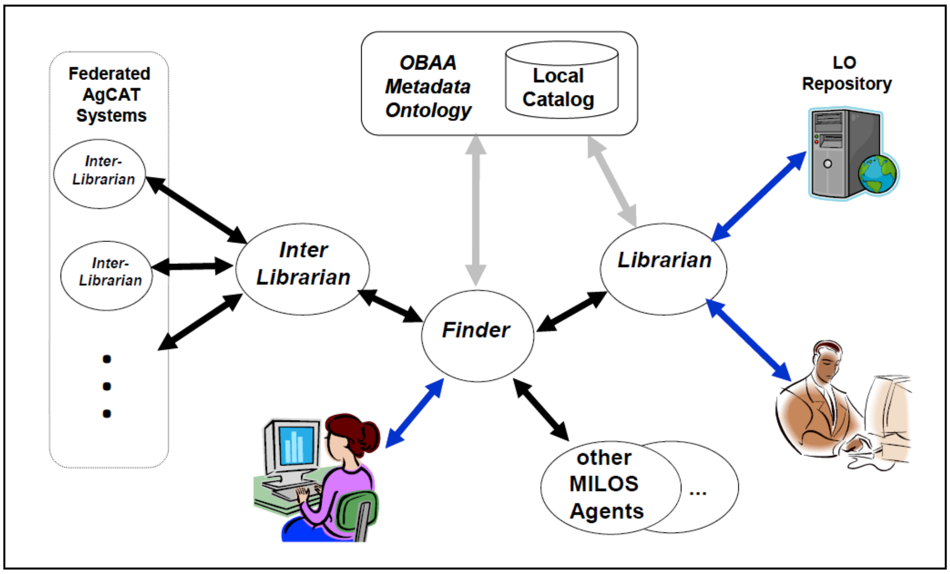


Figure 10: Architecture of AgCAT (Barcelos & Gluz, 2011).

AgCAT system is part of MILOS infrastructure (Multi-agent Infrastructure for Learning Object Support), and supports OBAA metadata proposal. OBAA metadata proposal is a Brazilian initiative to support multi-platform adaptability, compatibility with other metadata standards, and learning objects accessibility. Main goal of MILOS is to support all functions specified in OBAA metadata standards (Barcelos & Gluz, 2011).

AgCAT system consists of three types of software agents: Finder, Librarian, and InterLibrarian. Finder provides search service for AgCAT users and other MILOS agents, while Librarian collects metadata from LO repositories and store it in local catalog. InterLibrarian is responsible about establish of federated AgCAT systems.

3.1.10. An Ontology-Based Learning Resources (Sridharan, Deng, & Corbitt, 2011)

This framework proposed five features for learning objects exploratory. These feature are: authentication of retrieved resources, automatic ontology-based query refinement, reuse-oriented management of retrieved resources, adaptive retrieval of learning resources based on preference of individual learners, and synthesis of retrieval and management activities for creating reusable learning repositories.

The framework is an ontology-based conceptual model to enhance exploratory of learning resources by provide context-specific resources, validation resources, flexible to differences in learning styles and individual preferences, synthesis retrieval and management activities to create quality learning object repository.

3.1.11. Preferred Personalization Learning Object Model (PPLOM) (Sree Dharinya & Jayanthi, 2012)

Personalization learning objects can be attained based on user preference and learning style. Granularity of learning objects has direct impact on reusability. Granularity is the size of learning object and coupling with other objects. Fine granularity and reduce size of learning object, expect to increase reusability. Personalization can be effective, when reusability is effective.

Preferred Personalization Learning Object Model (PPLOM) proposed in this research to have effective and adaptive personalization.

Model will use user's past history to identify his preferences, also it will be known for algorithm object prerequisites. Model will use preference based algorithm and correlation based algorithm. PPLOM will help users to find learning objects not only depends on keywords, but also using prerequisites information and additional search algorithms for object meaning and context. Feedback by use learning object can give indication about its granularity, and level of personalization.

Figure 11 shows PPLOM model.

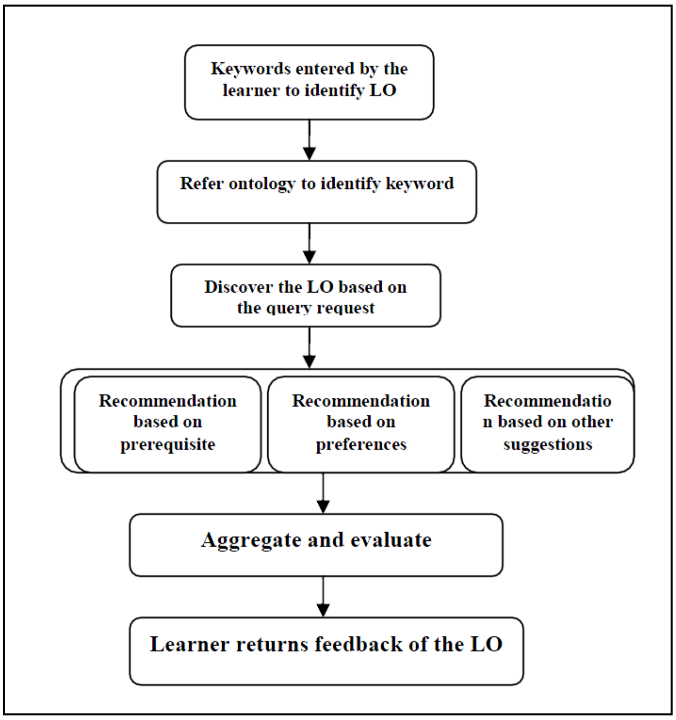


Figure 11: PPLOM model (Sree Dharinya & Jayanthi, 2012).

3.1.12. Clustering by Usage (Niemann et al., 2012)

This research concentrated on find semantic similarities between learning objects. Technique relies on usage-based relations between objects themselves without need for relations with user. Higher order co-occurrences that can create semantic clusters of learning objects will be considered similar.

This algorithm didn't use content of learning object or additional semantic metadata.

3.1.13. Recommendation for Interdisciplinary Applications (Chen & Huang, 2012)

This research concentrated on provides recommendation service for interdisciplinary learning applications, and taking into consideration changes in learner preferences and academic interests.

Figure 12 shows framework of interdisciplinary recommendation learning service system.

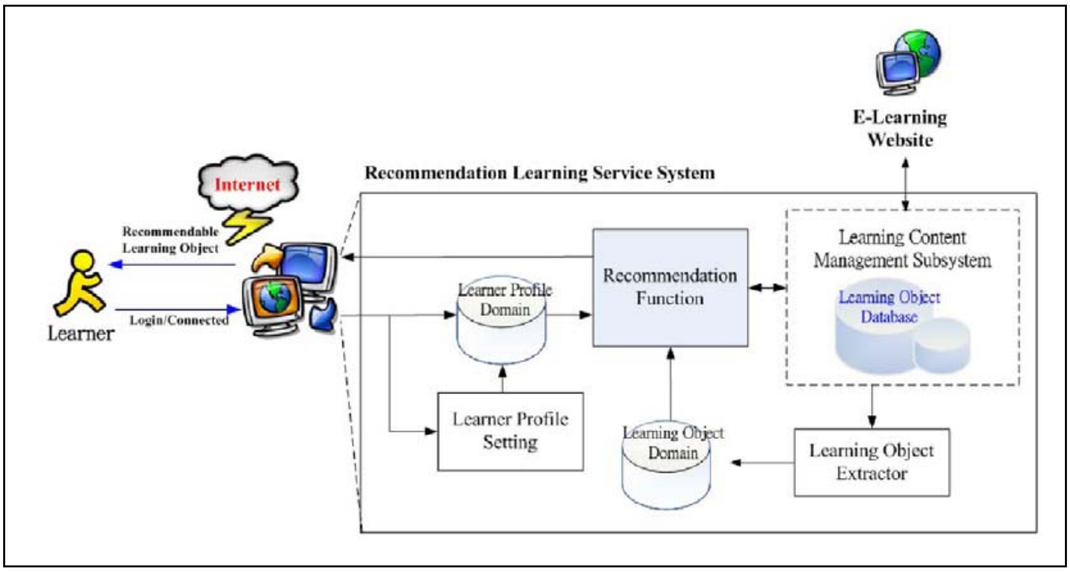


Figure 12: Framework of interdisciplinary recommendation learning service system (Chen & Huang, 2012).

Recommendation services use two types of ratings. Explicit rating that collect information from learners' input in course, quality of learning objects, and difficulty of teaching materials. Accuracy of this type is high. Second type of rating is implicit rating which automatically record learning paths and activities for analysis. This can be used to develop personalized recommendations.

Learning objects will be stored in Learning Content Management Subsystem, and Learning Object Extractor will transform learning objects into vector-based representations to be stored in Learning Object Domain. New learners will fill survey to check their interests and establish personalized profiles. Old users will be sent directly to Learner Profile Domain. Recommendation Function uses Learner Profile Domain and Learning Object Domain to recommend most appropriate learning objects for users.

3.1.14. Semantic Web Technologies (LOFinder) (Hsu, 2012)

This research proposed Multi-layered Semantic LOM Framework (MSLF) for integrate Semantic Web technologies into LOM by develop LOFinder system. The purpose from this system to enhance finding relevance learning objects.

LOFinder uses three approaches to find dynamic correlation of learning objects: learning objects metadata base, ontology base, and rule base.

LOM-based describes learning objects but cannot locate relevance learning objects, but OWL-based ontologies describes logic of LOM and this enhance semantic capabilities of LOM. Rule-based inference supports ontology-based by provide complementary capabilities. Using these three approaches can allow intelligence based on find relevance learning objects.

Figure 13 shows architecture of LOFinder.

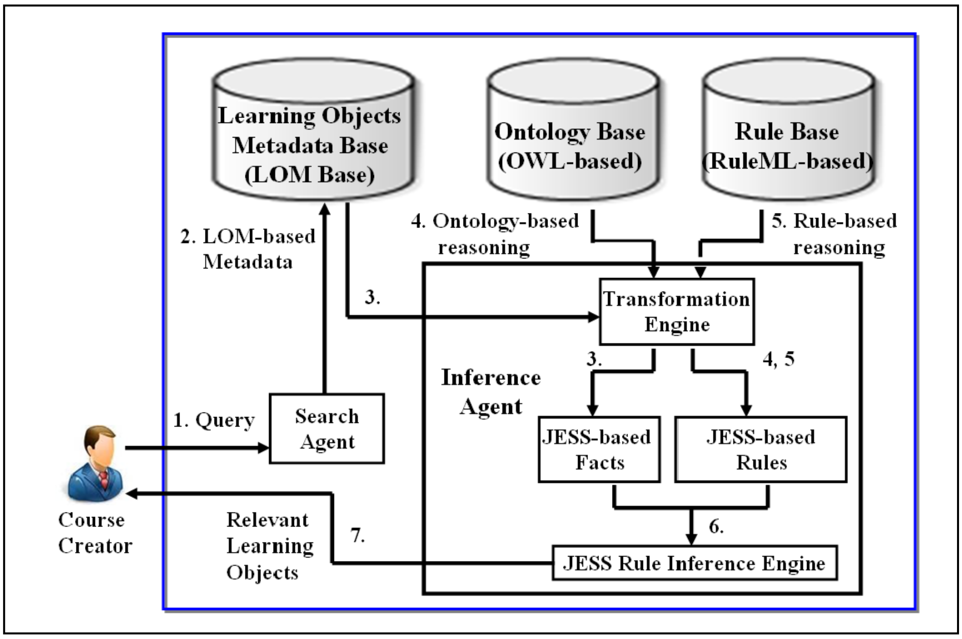


Figure 13: LOFinder Architecture (Hsu, 2012).

3.1.15. Recommendation in Adaptive E-Learning (Fouad Ibrahim, 2012)

This paper proposed system for personalized retrieval and recommendation of the learning objects based on learner profile. Learning objects metadata compared with learner profile to give recommendations. Semantic expansion for the query keywords used to find semantic similarity with learning objects.

Architecture in Figure 14 shows that main key for this system is to compare learning objects metadata with learner profile to give recommendations on learning objects. Semantic query expansion for learner query (keywords used in search) will be compared with learning objects metadata to improve recommendations.

Figure 14 shows architecture of Adaptive E-Learning.

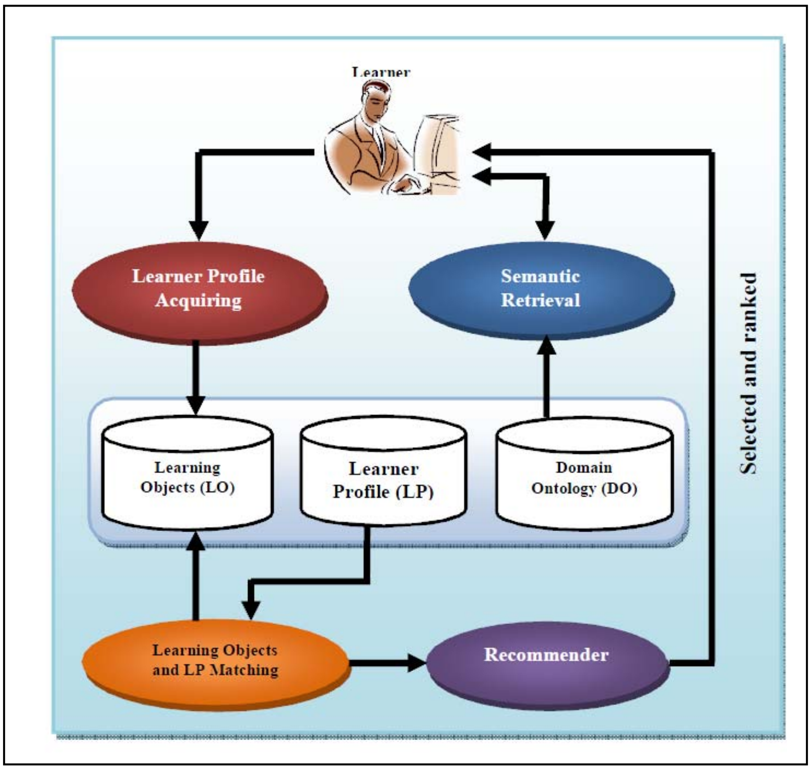


Figure 14: Architecture of Adaptive E-Learning (Fouad Ibrahim, 2012).

3.2 Data Mining for Learning Technology

Data mining has many applications, and used in different sectors. Various methods and techniques are in use within learning technology sector.

Table 3: Related work in data mining technology.

No.	Model	Classification	Description
1	Personalization Based on Web Usage Mining (Khribi, Jemni, & Nasraoui, 2009)	Web usage mining	Build learner and content models using data mining techniques, then use them to recommend learning objects

2	Educational Data Mining (Hung, Rice, & Saba, 2012)	Data visualization, clustering, relationship mining, and prediction	This work is review for literature in educational data mining
3	Data Mining in Virtual Learning (Gaudioso & Talavera, 2004)	Prediction and unsupervised	Analyse students' actions and assess their behaviours
4	Intelligent Learning Management System (ILMS) (Ueno, 2004)	Classification and text mining	Analysis for collaborative learning
5	Cross-level Frequent Pattern Mining (Huang, Chen, & Cheng, 2007)	Frequent pattern mining	Hierarchical scheme for learning suggestions
6	Time Dynamic Model Using Data Mining (Sharma, Jain, & Katare, 2011)	Clustering	Collect usage data for learners' behaviours to evaluate their performance
7	Data-Mining Technology for Material Recommendation (Liu & Shih, 2010)	Data association	Recommendation of learning materials
8	Data Mining in Context of E-Learning (ALMazroui, 2013)	Classification, regression, clustering, prediction, relationship mining, and visualization	Investigate contexts to identify problems using data mining

Table 3 outlines related work in data mining, and below sections describes these models. In comparison with our proposed work, the research (Khribi, Jemni, & Nasraoui, 2009) proposed to build learner profile using content and collaborative filtering approaches based on web usage mining, but this work didn't propose consolidated framework that gives flexibility, and didn't use wide range of data mining techniques to build user profile. In the research (Gaudioso & Talavera, 2004), data mining used to analyse students' behaviours to support tutoring and evaluation but not for ranking and recommendation of learning objects. In another research (Ueno, 2004), author proposed framework that uses text mining to let teachers and learners assess their behaviours in learning, but ranking and recommendation of learning objects not included in this research. The research in (Huang, Chen, & Cheng, 2007) is just a framework to suggest learning materials based on hierarchical structure, but no general framework for ranking and recommendation was proposed. The research (Sharma, Jain, & Katare, 2011) uses clustering technique to analyse students' behaviours to evaluate their performance without any ranking or recommendation functions. Authors in the research (Liu & Shih, 2010) used data association and collaborative filtering for recommendation of learning materials, but they didn't use usage data nor built user profiles.

3.2.1. Personalization Based on Web Usage Mining (Khribi, Jemni, & Nasraoui, 2009)

This approach has offline module which processes data and uses data mining techniques to build learner and content models. Online module uses these models to recommend learning objects. Offline module uses web usage mining techniques by gather web sessions, and applies clustering approach on these sessions. Learners in the same cluster have similar interests. In addition to that, association rules for clustered sessions will be used to find what learning objects associated with each other. This framework uses crawling and indexing of learning resources where each keyword mapped to set of pages that contained in them. The proposed process for this framework is shown in Figure 15.

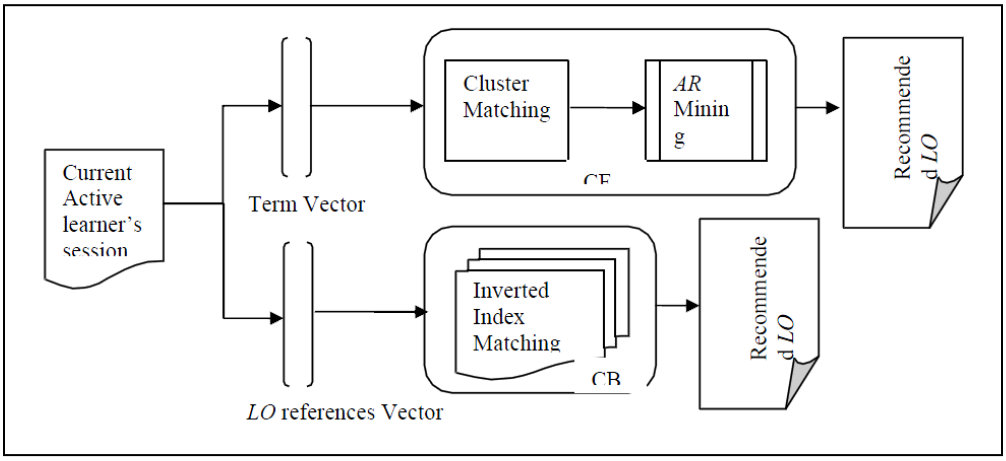


Figure 15: Recommendation process based on web usage mining. (Khribi, Jemni, & Nasraoui, 2009, p. 35)

3.2.2. Educational Data Mining (Hung, Rice, & Saba, 2012)

This paper is review for literature in educational data mining, and also proposed data mining model for education. This paper didn't propose any framework, but instead it presented four data mining methods that can be used in educational frameworks: data visualization (2D or 3D data visualization), clustering (K-mean or Hierarchical), relationship mining (association, sequential association, path analysis), and prediction (decision tree, regression, neural network). The paper explained detailed steps for data mining model (data source, analytic tool, data extraction, data pre-processing, data transformation, and data mining techniques).

3.2.3. Data Mining in Virtual Learning (Gaudioso & Talavera, 2004)

Data mining techniques used to analyse students' actions and assess their behaviours. The paper defined process in six steps: identify problem and data source, collect the data, clean the data by reprocess, build model in needed form, evaluate the model, and deploy the results. Paper proposed two models: predictive and descriptive. Predictive model used to find problems in previous courses to make improvements in future ones. Several prediction algorithms used in the research such as: J48, PART, JRip, and Naive Bayes. Descriptive model is unsupervised method with no particular variable to predict, but instead with goal to discover the structure of data. As an example for descriptive model, the behaviours of students can be clustered to find best groups for course project.

3.2.4. Intelligent Learning Management System (ILMS) (Ueno, 2004)

The paper proposed new data mining and text mining technologies to analyse data from collaborative learning. Mining functions in proposed work are: detection of irregular learning processes, content analysis, Decision Trees to analyse learners' historical data, Belief networks to analyse learners' historical data, analyses discussions using Markov, Entropy analyses for discussions, and text mining using expanded Correspondence Analysis for discussions.

3.2.5. Cross-level Frequent Pattern Mining (Huang, Chen, & Cheng, 2007)

Paper proposed methodology to suggest next course using mining frequent patterns for learners' behaviours. Cross-level learning suggestions for hierarchical scheme by provide students with multiple levels of suggestions instead of single level of frequent pattern result. This has been achieved by use frequent pattern mining (FP-tree) on multi-level hierarchy information.

3.2.6. Time Dynamic Model Using Data Mining (Sharma, Jain, & Katare, 2011)

This paper proposed time dynamic model to collect usage data for learners' behaviours and analyse it to evaluate learners' performance. Proposed model consists of six layers:

raw data layer, fact data layer, data mining layer, measurement layer, metrics layer, and pedagogical application layer. Clustering mining methods used to create measurements and metrics.

3.2.7. Data-Mining Technology for Material Recommendation (Liu & Shih, 2010)

Paper proposed application of data mining technology on recommendation of e-learning materials. Data mining algorithms used in this paper to improve material recommendations are: data association (Apriori, DIC, DHP).

3.2.8. Data Mining in Context of E-Learning (ALMazroui, 2013)

This paper tried to investigate contexts where data mining used in learning technology, and identify problems that can be solved using data mining. The paper listed various data mining methods used in e-learning and grouped them into six categories (Scheuer & McLaren, 2011). First category is supervised model induction where machine learning techniques used to predict known target attribute. Before use model, it is needed to train model using training instances. Examples of prediction models are classification and regression. Second category is unsupervised model induction. This model tries to find patterns and structures in data without define target attribute. Clustering is a widely used approach in this model. Third category is parameter estimation that used to predict the probability of events of interest. This method needs parameter to use such as mean and variance. Fourth category is relationship mining which used to identify relationships between variables. Examples of relationships are association, correlation, sequential, or casual in nature. Fifth category is distillation of data for human judgment that uses statistics, visualization methods, and interactive information interfaces to represent data by intelligent way. Sixth category is discovery with models that uses already created models to discover new information without need to create model from scratch.

In addition to above categories, more methods listed in (Barahate, 2012). These methods are: outlier detection, text mining, and Social Network Analysis (SNA). Outlier detection used to discover data with differences from the rest. This method can be used to detect students with learning problems, detect irregular learning processes, or detect deviations in learners' actions. Second method is text mining which is related to web content mining and used in collaborative mining and discussion evaluation, and also for documents grouping according to topics and similarities. SNA is to study relationships between entities within network structure. For example, it can be used to assess online interactions, group activities, and analyse structure and contents of online educational communities.

3.3 Conclusion

Literature review for this research includes information about related work in searching learning objects, and how each research tried to solve the problem. Also the chapter includes information about related work for data mining in learning technology domain, and how data mining used to solve learning technology problems.

Chapter Four

Methodology

4.1 Introduction

Finding relevant learning objects is very important for the existence of a learning objects repository. Many researchers tried to find appropriate solutions for this problem, and this led to the availability of large number of methods, yet a problem is not solved completely and there is some room for improvement. Combining and consolidating these methods can lead to a more effective solution, and improve the current situation.

Consolidated Ranking and Recommendation Framework is proposed in this thesis and it abstracts different methodologies used in this research problem, and gives the ability of join these methods. The framework allows other researchers to include their work into single and central framework to improve the search results problem.

Ranking and recommendation modules use user profiles to deliver relevant learning resources to the right users in the right context; right person in the right place and right time.

4.2 Relevant Research Work

There was a need to do a survey in two fields to be able to start the work on this thesis. First part is data mining techniques and how they can be best used in learning technology and what are the best data mining algorithms to use in each technique to achieve the highest correctness within the shortest time. Second part is searching learning objects frameworks, and how these frameworks improved reusability of learning objects, and to what extent they helped users find most relevant learning materials. Research on related work is presented in previous chapter.

There are several data mining techniques with a variety of tools that support many applications. Research has been done on available techniques and checked their ability to analyse usage data and build user profile. Finding a tool to support data mining using API to allow the system to create user profiles automatically was the key criterion to selection of the tool to use within this framework. Time cost to build and use data mining models, in addition to correctness of results were also key criteria to select the data mining algorithms to use within this framework. So, algorithms with the highest correctness and the lowest processing time are the best choice. High correctness will assure more accurate result for generated data mining models, and this will give us more relevant learning objects during search on learning objects or use the learning management system. Processing time to build and use data mining model is important factor as well, because it allows the framework to rank and recommend learning objects based on up-to-date usage data and deliver the result within short time.

It was important to build on the results of other researches that have been done around finding relevant learning objects. This thesis joined most important, to our knowledge, efforts already done in this domain, and proposed a consolidated framework to have the ability to be used by all researchers.

4.3 Quantitative Methodology

The aim of this research was clear in advance, but our recommendations appeared at the end of the research. We collected numerical data and statistics by analyse system logs and from survey filled by participated users. All details of the study were carefully designed, and then we analysed the collected data during our research.

We looked for efficient methodology to be able to test hypothesis, and prove results by numbers and statistics. So, the quantitative methodology has been used during this research.

Comparing ranking and recommendation results with other related work cannot be done because of different learning object repositories used in different researches. For that reason, this can be an exception for the quantitative methodology used in this research.

4.4 Evaluation of Data Mining Techniques

The main advantage of the proposed framework is its ability to create user profiles automatically based on usage data, and then use these profiles in ranking and recommendation. A flexible data mining tool with ability to support multiple algorithms and can deliver correctness with high quality and within reasonable time cost is needed.

The performance of data mining algorithms is very sensitive to the number of attributes, number of instances, and type of data (string, number, nominal, etc.). So, it was important to evaluate all available data mining algorithms using real usage data from learning management system with exactly the attributes and type of data that is going to be used in the real proposed system. During evaluation of data mining algorithms, we collected real usage data from learning management system, and then duplicated it to have big number of records. Collected usage data used as input for each algorithm, then build and use data model were executed.

Weka library has been used to implement data mining techniques, and there are many techniques used in the proposed framework (classification, anomaly detections, clustering, and association rules).

4.5 Design of Consolidated Ranking and Recommendation Framework (CRRF)

The framework is flexible since it allows researchers and other stakeholders to integrate their methods into this system, and contribute to improving the ranking and recommendation of learning objects. The framework has an internal method to combine results of all methods into one single result.

This thesis proposed a ranking and recommendation algorithm based on usage data (attention metadata). Algorithms collect usage data, create user profiles using data mining techniques, and then use it for ranking and recommendation. The proposed algorithm is the first contribution in CRRF.

The framework supports web services and SOA to allow easy integration with other systems such as learning object repositories and learning management systems.

4.6 Evaluation of CRRF

The framework and algorithm were used to develop a search engine for learning objects with ranking and recommendation functions. Validation of proposed framework and the used algorithms has been performed by involving end users for ten days, in usability sessions, to measure the required improvements.

Two methods were used to evaluate CRRF. The first method is the well-known information retrieval Precision and Recall measures to analyse logs, and calculate True Positive, False Positive, True Negative, and False Negative to find Precision and Recall for Ranking and Recommendation. The second method is a usability Survey using SUS (System Usability Scale). The survey used to collect feedback on usability of the proposed solution from representatives and users (Sauro, 2011). This survey has been used in several related work, and used to compare usability of proposed search engine with others. In addition to SUS, special survey (3 questions only) used to measure user opinion about best criteria to use for Ranking and Recommendation. For example, if the user prefers ranking according to courses he is enrolled in, or prefers to use search objectives. Details of the used survey exist in Appendix 7.

These methods helped to evaluate the work, find good points and recommend them, and find weakness points to improve them.

4.7 Conclusion

This chapter explains the methodology used in this research and how it is mainly different from other researches in the same domain. The research methodology is quantitative where numerical data is collected and analysed to evaluate the proposed framework. Also the chapter explains the evaluation methodology for data mining techniques and the proposed framework.

Chapter Five

Consolidated Ranking and Recommendation Framework (CRRF)

5.1 Introduction

Finding relevant learning objects is very important for successful reuse in learning objects repositories.

Consolidated Ranking and Recommendation Framework (CRRF) has been proposed which abstracts different methodologies used in this research area, and gives the ability to integrate these methods. The framework will allow researchers to integrate their work into single and central framework to improve the result, and contribute to improve ranking and recommendation of learning objects using internal methods to combine results of all methods into one single result.

Ranking means to give a mark for learning objects to be retrieved in a search engine or interface, and view learning objects with highest mark first, and identify it as most relevant. Ranking itself can be identified as a specific case of recommendation as well. In this thesis, we will define recommendation as the learning objects that the system suggests to the user during their normal navigation without need for explicit formulation of search queries by the user himself.

5.2 CRRF Architecture

The main purpose of the new proposed framework is to find abstract framework that can be used by stakeholders of learning management systems in this domain and allow them to contribute in ranking and recommendation of learning objects.

Ranking and recommendation process is summarized in Figure 16. The main source for any method is learning objects including learning object metadata (LOM). Additional information needed in some algorithms to rank and recommend learning objects. Result of this process is a list of ranked and recommended learning objects.

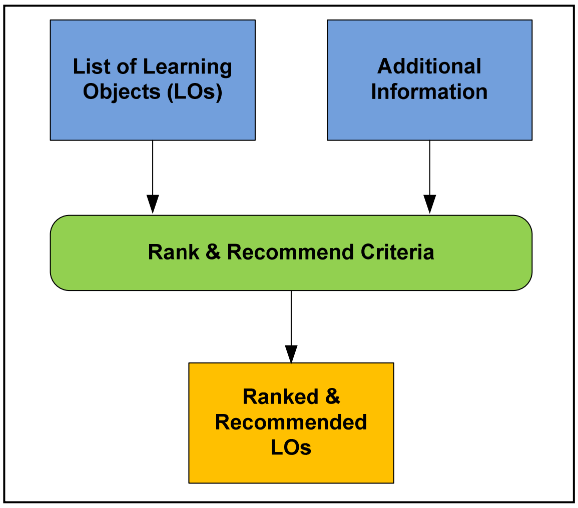


Figure 16: General Ranking and Recommendation Process.

The framework receives list of learning objects and additional information from different data sources as input and produce ranked and recommended learning objects as an output of implementing rank and recommend criteria which utilise different algorithms. Also some other techniques depend on manual review by humans for learning objects to classify them.

CRRF architecture tries to provide all available data sources in this domain to be used together in consolidated form. Also framework will be flexible for any additional information that can be provided in future researches, for example, researcher may add information from social networks about the user to get an idea about user's interests and use them in ranking and recommendation functions. Algorithms and criteria can be customized to use any data source or combination of data sources.

Ranking and recommendation techniques use multiple data sources such as: learning objects (LOs) which is binary files contain learning objects, learning objects metadata (LOM), users of the system, usage data that can be collected from interactions between users and learning objects in addition to usage data from other systems, User Profiles that contains information about the user to be used for context recommendation, semantic knowledge that can be extracted from learning objects or its metadata, and additional information that makes this framework flexible.

Ranking and Recommendation can be manual or dynamic. Manual methods depend on human review of learning objects, but dynamic methods use algorithms and criteria to calculate ranking and recommendation automatically. CRRF supports both methods, and researchers can integrate both methods into this framework.

CRRF supports complex algorithms and criteria to calculate ranking and recommendation. Formulas can use information from multiple data sources, and can use values from other formulas. Data from manual ranking and recommendation and result of other dynamic methods can be used as data source for other algorithms. For example, when a user rates a learning object as relevant for him, the recommendation algorithm can read this data, and suggest a learning object to peer users. Flexibility of possible algorithms and criteria enrich this framework with all tools needed to calculate ranking and recommendation. This feature allows other researchers to integrate their work in this framework easily.

Figure 17 shows data flow diagram of CRRF.

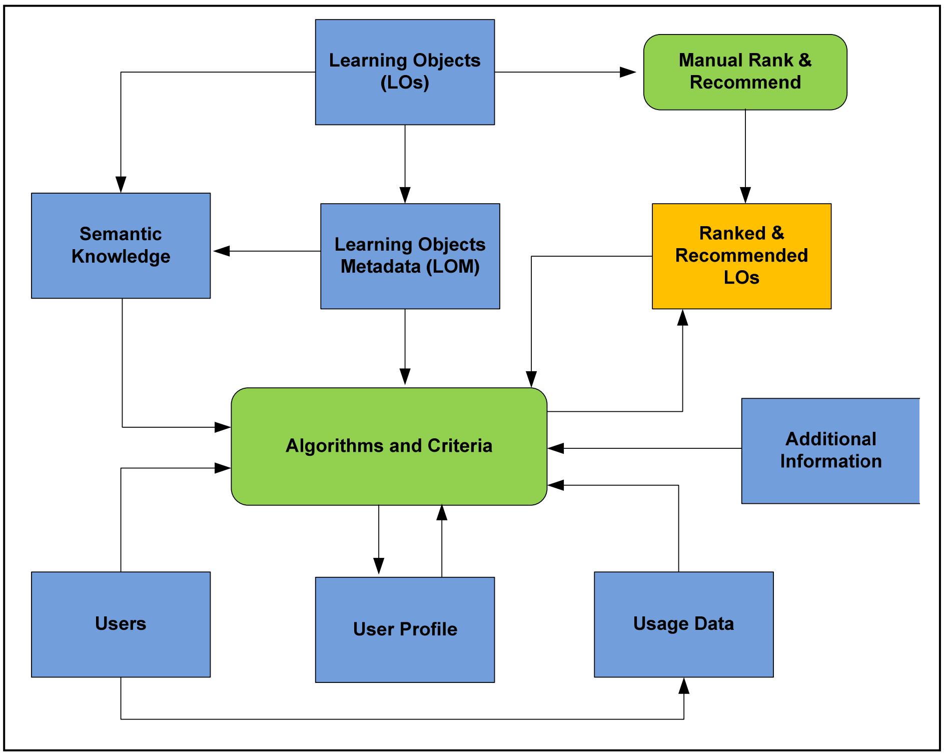


Figure 17: Data Flow of Consolidate Ranking and Recommendation Framework (CRRF).

CRRF formulas support different calculations such as basic mathematics operations (addition, subtraction, division, multiplication), statistical operations (average, standard deviation, mean), logical operations, and data mining algorithms using Weka library.

The CRRF checks results of ranking and recommendation from all participated techniques and join them together to have a more accurate and appropriate result.

Support consolidated techniques in ranking and recommendation learning objects is the key feature of CRRF, so service-oriented architecture (SOA) will be used in this framework. Figure 18 shows how the proposed framework supports SOA.

Figure 18 shows SOA architecture for CRRF.

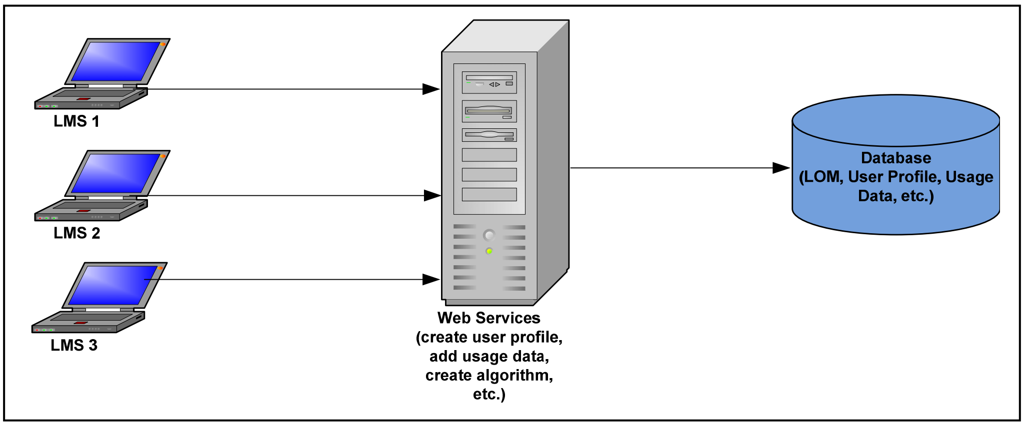


Figure 18: SOA architecture for CRRF.

The system functions are installed in web services (SOAP) server to support any learning system (platform independent), and allow easy integration with CRRF. Web services will abstract complexity of framework, and provide common interface with high flexibility. Researchers and learning systems will be able to use the web service to configure algorithm and criteria for ranking and recommendation, create setup data for algorithm such as user profile, and access ranking data result from configured algorithm or ranking result from other techniques including consolidated results.

5.3 Ranking and Recommendation Based on Usage Data

Figure 19 shows the architecture of the proposed system that will be added to the proposed consolidated framework.

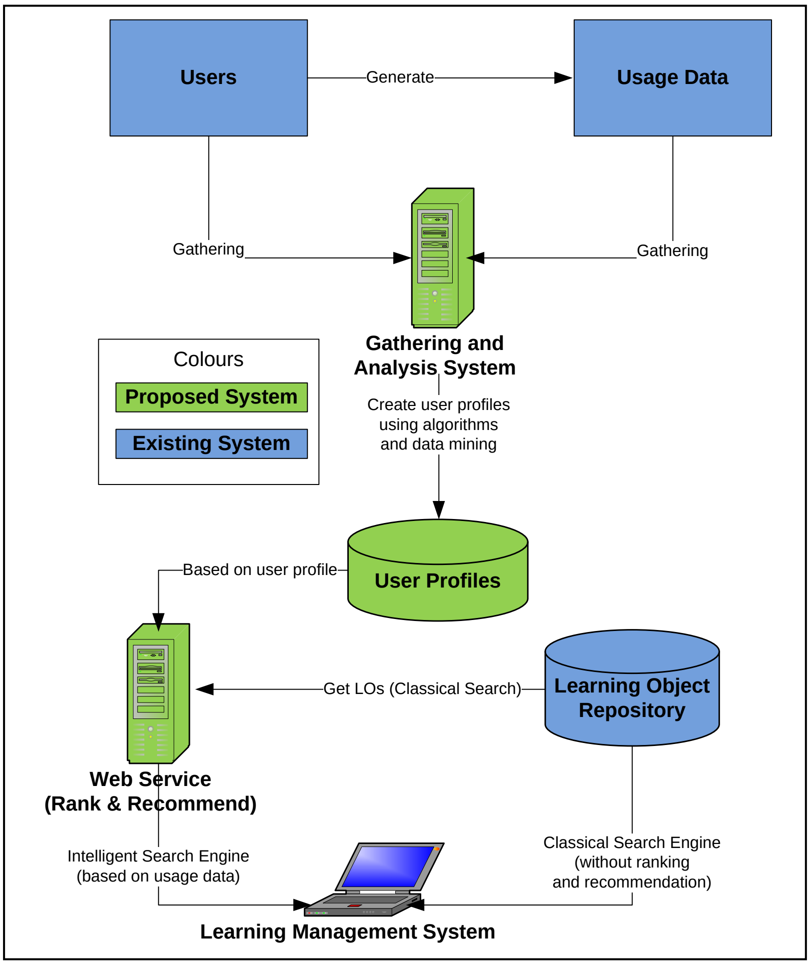


Figure 19: Architecture of ranking and recommendation based on usage data.

The proposed system consists of nodes in green, while other nodes are existing systems and resources already exist. Gathering and Analysis System (GAS) will collect data about users and their usage data, and apply algorithms and data mining techniques to create user profiles. These profiles will contain information about users that is needed during the ranking and recommendation process. The gathering and analysis process will be a background jobs done in the back-end of the system in asynchronous mode to have user profile ready for use when user starts interact with search engine. The system will also contain ranking and recommendation web service to replace classical search engine. The web service will provide intelligent search engine that depends on the created user profile and classical search engine.

The result of the intelligent search engine is a ranked learning objects list with high relevance first. In addition, the framework will recommend learning objects for the user during system navigation.

5.3.1. User Profile

All collected raw usage data cannot be used directly in ranking and recommendation process, so user profile will be created to achieve this purpose. Analysis of usage data using data mining techniques produces information in the user profile. Ranking and recommendation processes will use the produced user profile.

The user profile contains information about contexts, learning interests, search objectives, and relations. Contexts will be defined based on the content of courses in learning management system (LMS) that the user is enrolled in, and institution where LMS (e.g. Moodle) is installed. On the other hand, learning interests will be defined based on user's interactions with courses and LMS sections. Search objectives are user's interactions with search engine, and will be defined using keywords used during search on learning objects, and according to selected learning objects. This property is different from learning interests in the data source, search objectives represent interactions with search engine, but learning interests gathered from interactions with learning management system. In addition to that, data sources used to define search objectives contain small text (such as keywords or learning objects title), but interactions with courses and learning management system may produce large text fragments. The user profile has information about relations. Three factors are used to define the relations between users. First factor is users enrolled in the same courses and institution where LMS is installed. Second factor is the interactions with search engine, for example, search using same keywords, rate same learning objects, or select same learning objects. Third factor is the interactions with the LMS and how it was used, for example, users interact with the LMS in same way (rarely open the LMS, use forums and chat, upload attachments in similar topic, etc.), users open LMS from same location, users know same language, or belong to same country.

Figure 20 shows entity relationship diagram for user profile, and relations between tables including primary keys and indexes.

The entity crrf_user_profile is the core entity in this diagram. The attribute user_id is an id for the user given by LMS (e.g. Moodle). This id with institution id will be used to identify user in the system. The attribute last_update is date and time when the profile of the user was updated last time. The attribute interest_model_version is the version of learning interests model (one-class classification). The model file will be stored in file system where CRRF is installed. The attribute objective_model_version is the version of learning objectives model (one-class classification). The model file will be stored in file system where CRRF is installed. The attribute institution_id is to link the user profile with his institution to allow system support multiple institutions with similar user id.

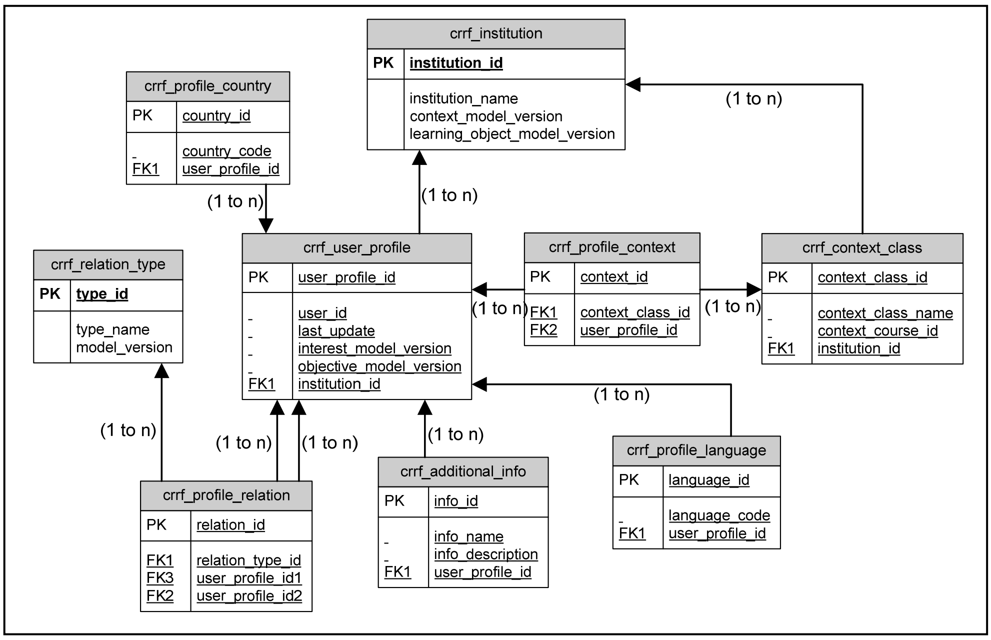


Figure 20: Entity relationship diagram of user profile.

The entity crrf_institution is to store all instances of learning management systems and were installed. The attribute institution_id is the primary key. This id will be used by institutions to interact with CRRF system. The attribute context_model_version is the version of contexts model (text classification). The model file will be stored in file system where CRRF is installed. The attribute learning_object_model_version is the version of learning object model (nearest neighbour search). The model file will be stored in the file system where CRRF is installed.

The entity crrf_profile_country stores all countries associated with a user profile. The attribute country_code based on ISO 3166-1 alpha-2 (Country Codes - ISO 3166, 2013). For example, PS is country code for State of Palestine. Attribute user_profile_id is foreign key to crrf_user_profile.

The entity crrf_profile_language stores all languages associated with a user profile. The attribute language_code based on ISO 639-1 (ISO 639 Language Codes, 2013). For example, “ar” is code for Arabic, and “en” for English. The attribute user_profile_id is foreign key to crrf_user_profile.

The entity crrf_context_class stores all context classes. The attribute context_class_name is usually the course name used by learning management system. The attribute context_course_id is the id of the course associated with the context and used by learning management system. While the attribute institution_id is a foreign key to crrf_institution to allows support different contexts per each institution.

The entity crrf_profile_context is used to link a user profile with all context classes associated to it. The attribute context_class_id is a foreign key to crrf_context_class. The attribute user_profile_id is a foreign key to crrf_user_profile.

The entity crrf_relation_type is to store relation types between user profiles that generated using clusters. The attribute type_id is the primary key and will be added manually when configure new relation type. The attribute type_name is a string to describe cluster model. The attribute model_version is the version of relation model (clustering). The model file will be stored in file system where CRRF is installed. Currently there are three relation types: course clusters, search clusters, and learning clusters.

The entity crrf_profile_relation is to define relations between user profiles. The attribute relation_type_id is a foreign key to crrf_relation_type. The attribute user_profile_id is a foreign key to crrf_user_profile. The attribute group_id is the cluster id, and it will be used to identify what user profiles belong to same cluster, in other words, what user profiles within relation.

The entity crrf_additional_info can be used to store any additional information on user profiles, and then use it in ranking and recommendation modules. The attribute info_name is any text to be used for information. The attribute info_name is any text to be used for information. The attribute info_description is any text to describe additional information. The attribute user_profile_id is a foreign key to crrf_user_profile.

5.3.2. Gathering and Analysis System (GAS)

The GAS is responsible for gathering usage data from learning management systems, and uses it to build user profiles automatically. Since Moodle was the development environment in this thesis, we will use it as source of the data.

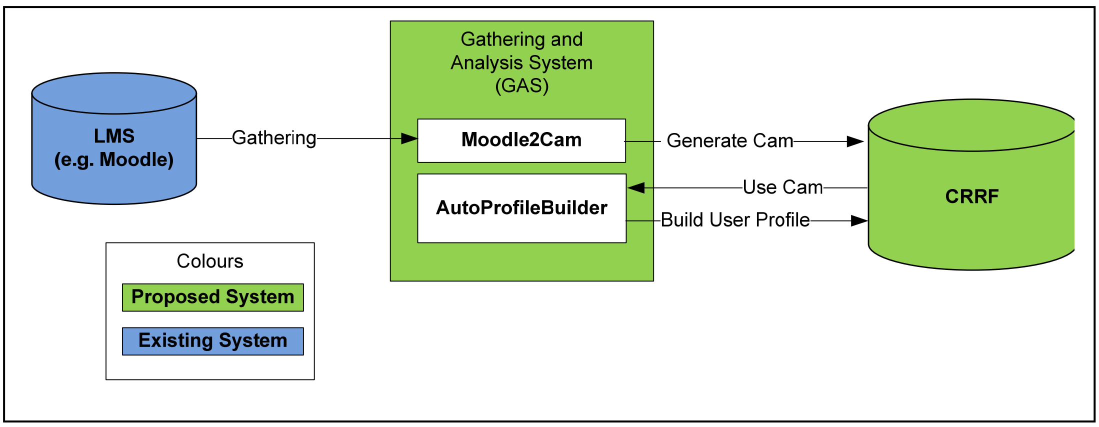


Figure 21: Gathering and Analysis System (GAS) Modules.

GAS has two modules: Moodle2Cam and AutoProfileBuilder. Both modules work in background and are calculated in asynchronous jobs because they take long time to be executed. Moodle2Cam module will work first to gather usage data, and after complete gathering, AutoProfileBuilder will work to use gathered usage data to build user profiles.

5.3.2.1. Moodle2Cam:

Moodle2Cam is the main data source in the framework, and will collect usage data from Moodle and store it into CAM format to be used later to create user profiles.

Moodle2Cam module has three functions: (1) gathering usage data from learning management system, (2) converts usage data to CAM format, (3) and stores CAM data in relational database tables inside CRRF system.

Moodle2Cam will use four Moodle's data sources to collect CAM. First data source is log data from Moodle and all associated sources with the log. Second data source is user's enrolments into courses. Another data source is attachments and files loaded into course such as lectures, assignments, answers, etc. Fourth data source is user's interactions with search engine (widget) such as search, rank, and select learning objects.

Below sections describe in details the mapping between above data sources in Moodle and how to map them into CAM format.

Moodle Log Data:

Log data from Moodle is the main source of users' actions. All interactions with Moodle will be stored in a log such as courses, assignments, chats, choices, forums, glossaries, lessons, quizzes, surveys, wikis, workshops, etc. Additional resources will be also collected even this field doesn't exist in CAM, but it is necessary to have more information for data mining use. Examples of additional resources are course name, discussion topic, user language, user country, forum subject, course section name, resource name, module name, assignment subject, glossary name, quiz title, survey title, workshop subject, book title, URL description, choice title, chat subject, etc.

User's Enrolments:

User's enrolments into courses are collected to know users' context and interests also will be used to define peer users. Enrolments information will be loaded from LMS database.

Attachments and Files:

Moodle2Cam module will parse all attachments and files loaded into Moodle, and extract text from them. The text will be used to define user's contexts using text mining techniques.

Several libraries and APIs are used to support different file formats and enable extract text from all possible resources. Libraries used to extract text are: Apache POI (http://poi.apache.org), Apache PDFBox (http://pdfbox.apache.org), and Apache Tika (http://tika.apache.org). Apache POI (Java API for Microsoft Documents) supports work with Office Open XML standards (OOXML) and Microsoft's OLE 2 Compound

Document format (OLE2). This library supports text extraction from Excel, Word, PowerPoint, Publisher, and Visio. Apache PDFBox (Java PDF Library) supports work with PDF documents and text extraction including Arabic language using ICU4J (http://site.icu-project.org). Apache Tika (Content Analysis Toolkit) library extracts context text and metadata from various file formats. This library supports the extract of text from HTML, XML, ODF (OpenDocument Format used in OpenOffice), EPUB (Electronic Publication Format), RTF, Compression and packaging formats, Text formats, Audio formats, Image formats, Video formats, Java class files and archives, and mbox format. Figure 22 shows these libraries, and supported formats.

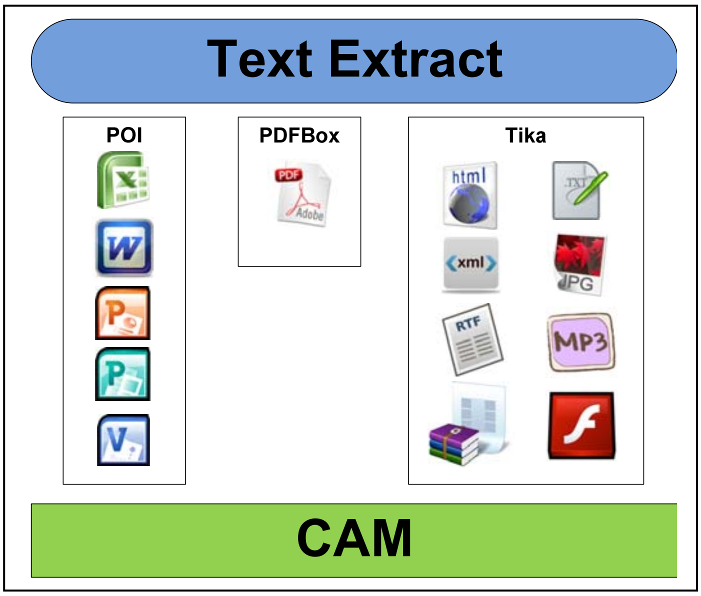


Figure 22: Text extract support from attachments and files.

User's Interactions with Search Engine:

Interactions with search engines (widgets) such as searching actions, rating learning objects, and selecting learning objects. These actions will be collected by the widget, and saved into CAM format.

5.3.2.2. AutoProfileBuilder:

AutoProfileBuilder module is responsible about analysis of usage data stored in CAM, and collected from learning management system and search engine, and then build user profiles. The process will be automatic by collecting data, perform analysis, and build models for user profile. For example, if subscriber enrolled in Java Programming course, Moodle2Cam will collect this attention, and saves it into CAM. Later, AutoProfileBuilder will load this attention, and add it to user contexts and interests by modifying the models. These models will be used in ranking and recommendation services which will have effect on the result of search.

Data mining techniques and algorithms used to build the user profile are: text classification, one-class classification / anomaly detection, nearest neighbour search classification, association rules, and clustering. Text classification will be used to classify all courses' content, and define users' contexts. A model for all courses will be generated. This model will be used to define the relevance between a topic and user's contexts. Contexts will be categorized according to the institution where the LMS is installed. One-Class classification / anomaly detection is used to classify user's learning interests and search objectives. One model for user's learning interests and another one for user's search objectives will be generated. These models will be used later to decide if some topic is within user's learning interests or search objectives or how it is relevant to them. Nearest neighbour search classification model will contain context, language, country, IP address, and learning object. Model will be used later to decide the nearest or the most relevant learning object for a user using these parameters. Association rules model will measure the association between learning objects, and what learning objects usually used together. In addition to that, association between users depends on relations between them can be generated to recommend learning objects for users depends on selected or rated objects by peer users. Clustering will generate three clusters to define relations between users. Clusters are: courses and institutions cluster, search engine actions cluster, and LMS actions cluster. Users in the same cluster will be considered with relation, and relation level will increase if exist in more clusters, for example, share same courses and search engine clusters.

Figure 23 shows the architecture of AutoProfileBuilder module, how integrates with other modules, and data mining techniques used during build user profile. The figure shows input data for each data mining technique and the output attribute within the user profile.

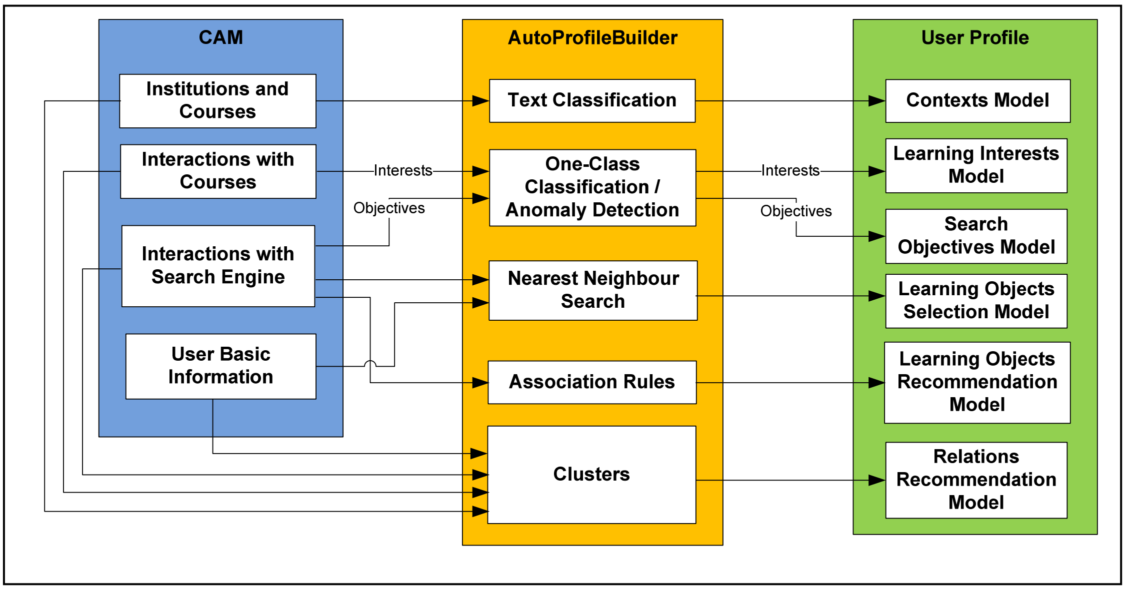


Figure 23: Architecture of AutoProfileBuilder module.

Figure 23 shows how AutoProfileBuilder uses CAM data to build user profiles. Text classification uses full text of courses extracted by Moodle2Cam, and builds a context model for each LMS instance. A user profile will be associated to contexts model for the same LMS instance, and for courses that user enrolled in them. Learning interests model will be built using interactions with learning management system and user's courses; e.g., when user download lectures, upload files, participate in forums, and upload answer of assignment. All these actions will be gathered, and used to build a user's learning interests model. Search objectives model will be build in the same way, but using search keywords used during interactions with search engine. Learning objects selection model is a way to find the most suitable learning objects using user's basic information as input. This model will be built based on selected learning objects by users and their basic information. The purpose of this model is to find learning objects that are being selected frequently by group of users with similar language, country, or enrolled in same courses. Learning objects recommendation model is simply used to find those learning objects that were selected frequently together in same session. Clustering algorithm will be used to group users, and then consider all users in the same group as peer users, and recommend learning objects according to peer users' actions.

5.3.3. Ranking and Recommendation Modules

The ranking module is responsible for ranking search results according to their relevance to the user. The most relevant learning objects, according to user profile, appear first. Ranking module uses the contexts model, learning interests' model, and search objectives model to rank learning objects.

Recommendation module works while users browse learning objects and learning management system. The module recommends learning objects that are relevant to the user without explicitly searching for them. The module provides users with relevant learning objects even without performing any search query. Recommendation module uses learning objects selection model, learning objects recommendation model, and relations recommendation model to recommend learning objects.

Ranking and Recommendation modules have been designed to support flexibility in building ranking and recommendation criteria and formulas using user profiles.

5.3.3.1. Entity Relationship Diagram:

Figure 24 shows ER diagram for Ranking and Recommendation modules.

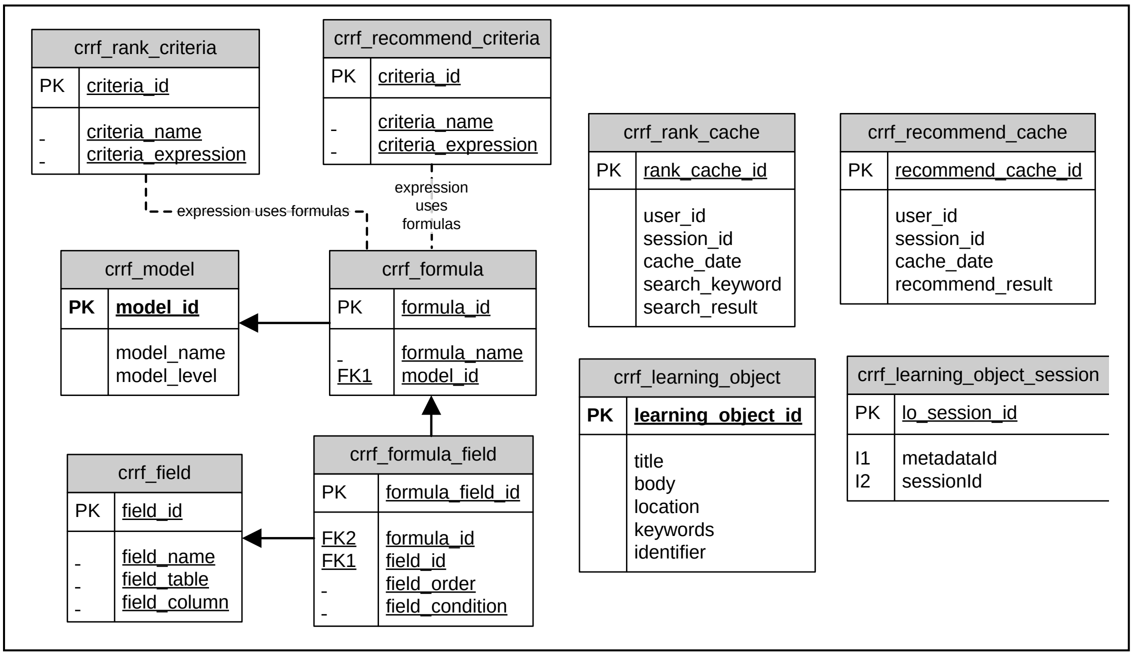


Figure 24: Entity Relationship (ER) Diagram of Ranking and Recommendation modules.

The entity crrf_rank_criteria and crrf_recommend_criteria are the first step for Ranking and Recommendation algorithms. The attribute criteria_name describes ranking and recommendation criterion. The attribute criteria_expression is a mathematical expression to calculate ranking and recommendation. All mathematical operations can be used within this expression, also constant numeric values. An expression can have many formulas in format of Fx, where x is formula_id in table crrf_formula. For example, expression can be F1+(F2*2).

The entity crrf_formula provides flexibility in calculate ranking and recommendation criteria. The attribute formula_name describes the formula. The attribute model_id is a foreign key to crrf_model.

The entity crrf_model is list of all models available in system and can be used by ranking and recommendation formulas. This table provides high flexibility for the system to use models for different formulas. The attribute model_id is the primary key to be added manually when configuring a new model. The attribute model_name is used by the entire system to identify the model. The attribute model_level identifies to what level the model belongs. For example, INSTITUTION, USER, or PUBLIC. In case of institution, one model will be created for each institution such as context model and learning object model. For user level, one model will be created for each user such as learning interests model and search objectives model. In case of public level such as clustering, one model will be created all over the system and for all institutions.

The entity crrf_formula_field has all input parameters and database fields that will be used to calculate the formula. The system will get these input parameters and database fields, and then pass them to ranking and recommendation handlers where the real calculation will be applied. The attribute formula_id is a foreign key to table crrf_formula to define to what formula this field belongs. The attribute field_id is a foreign key to table crrf_field where details of the field exist. The attribute field_order specifies in what order to pass fields to ranking and recommendation handler. The attribute field_condition is used when field to be retrieved from database. The condition attribute is the normal condition statement in SQL database without 'where' statement.

The entity crrf_field contains details about the fields. The attribute field_name describes the field. The attribute field_table is name of input parameter or table name from database. Input parameters are list of learning objects to make ranking or recommendation for them. For example to indicate that field is parameter from list of input learning objects, then use ":INPUT_LO" as value for this attribute. If this attribute value doesn't equal to ":INPUT_LO", then the system will consider it as table name from database. Many tables can be configured in this attribute if the user intends to get data from more than one table, and can do this by set name of tables with coma separator and alias for each table to be used field_column attribute or field_condition attribute in table crrf_formula_field. For example, "table1 t1, table2 t2". The attribute field_column is input parameters or database fields to be retrieved and passed to ranking and recommendation handlers. If attribute (field_table) equal to ":INPUT_LO", then this attribute can have input parameter from learning object. Available learning object parameters are: ":INPUT_LO_TITLE" for learning object title, ":INPUT_LO_BODY" for learning object description, or ":INPUT_LO_KEYWORDS" for learning object keywords. There are two input parameters that can be used in field_condition without a need to define them in any place, and the system will get them from logged user and input list of learning objects that need to be ranked or recommended. These two parameters are: ":INPUT_LO_ID" for learning object id, and ":INPUT_USER_ID" for user id sent ranking or recommendation request. If value for field_column attribute doesn't start with ":", then it will be considered as column names from database tables mentioned in field_table attribute. The user can retrieve many columns by set name of columns with coma separator. System will pass these database columns to ranking and recommendation handlers using same order provided in this column. For example, user can use "t1.column1, t1.column2, t2.column1" to retrieve three columns from table1 and table2. We can notice here that usually select statement has three parts "select columns from tables where conditions". User can configure "columns" in crrf_field.field_column, and "tables" in crrf_field.field_table, and "conditions" in crrf_formula_field.field_condition. Tables crrf_field and crrf_formula_field provide high flexibility for ranking and recommendation handlers to use any data for their functions.

The entity crrf_rank_cache is used to manage cache of ranking result for a user. This allows system to deliver fast paging for search result instead of repeat ranking process for every page. The attribute user_id is a user for the cache belongs to it. The attribute session_id is HTTP session where the cache created in. The attribute cache_date is time of cache, and this time will be used to help cache manager to decide if use this cache or mark it as expired, and launch ranking operation again. The attribute search_keyword is the keyword used in search, and it will be used by cache manager to decide if user is doing search for same data or paging the result. The attribute search_result is the full ranked list of learning objects using json format. Ranking module will use this list, and get required subset for required page.

The entity crrf_recommend_cache is used to manage cache of recommendation result for a user. This allows system to deliver fast recommendation instead of repeat its process for every page. The attribute user_id is a user for the cache belongs to it. The attribute session_id is the HTTP session where the cache created in. The attribute cache_date is time of cache, and this time will be used to help cache manager to decide if use this cache or mark it as expired, and launch recommendation operation again. The attribute recommend_result is the full recommended list of learning objects using json format. Recommendation module will use this list, and get required subset for required page.

The entity crrf_learning_object has information about used learning objects by system's users. This entity is used when system uses external repository for learning objects, and it allows system to recommend learning objects within short time. Proposed system will store information in this entity when users do their search, and this information will serve during recommendation instead of get information from repository. The attribute learning_object_id is the primary key for learning object that used by external repository. The attribute title is the learning object title. The attribute body is the learning object description. The attribute location is the URL for learning object. The attribute keywords is learning object keywords. The attribute identifier is learning object identifier produced by learning object repository.

The entity crrf_learning_object_session is simple to store all selected learning objects by system users and in what sessions they were selected. This entity exists for performance issue, and to allow (frequently selected together) recommendation function to work within short time. The attribute metadata ID is the selected learning object id. The attribute session ID is the HTTP session where learning object is selected.

Current configured data for Ranking and Recommendation criteria listed in Appendix 3.

5.3.3.2. Class Diagram:

Abstract class GeneralTechnique is the root class for all data mining techniques, and has common attributes and functions that are used by all techniques. This class implements Serializable to allow system to serialize all data mining models into file system by GAS, and to be deserialized by Ranking and Recommendation modules.

GeneralTechnique class is the parent class for all data mining techniques. This class has most attributes and methods used by other classes. Attributes of GeneralTechnique class and its methods listed in Appendix 4.

Figure 25 shows class diagram for Ranking and Recommendation modules, and how classes depend on each other.

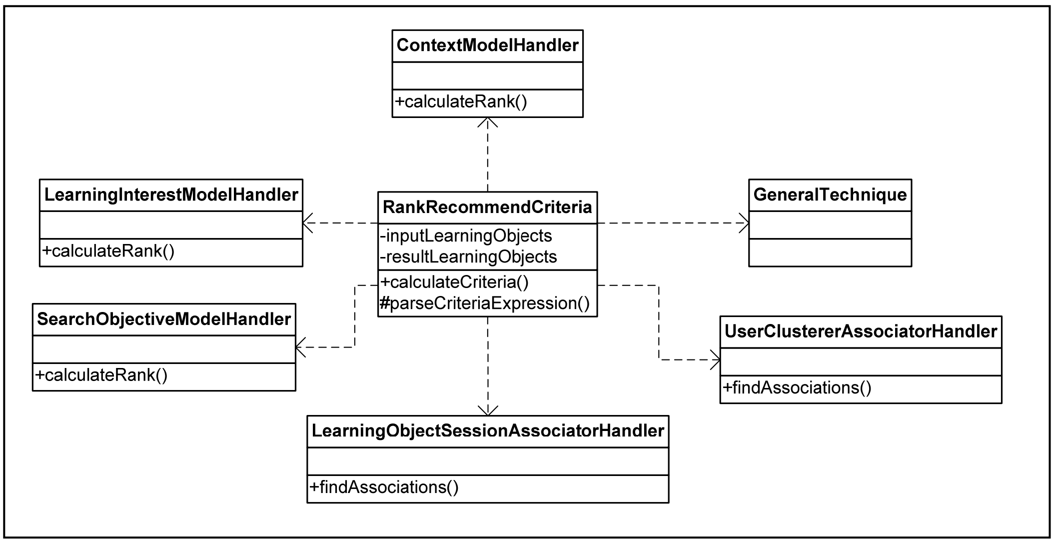


Figure 25: Class Diagram of Ranking and Recommendation modules.

Abstract class GeneralAssociator is a general class for all association rules techniques. The class has associator attribute to define which associator to use. Also has buildIfNeeded method to find all required association rules. The class LearningObjectSessionAssociator is the only available sub-class to be used for association rules. The class has apriori attribute as associator, and setupAssociator method that configure technique attributes. The class has three sub-classes: TextClassifier, LearningObjectClassifier, and OneClassTextClassifier.

Figure 26 shows class diagram for data mining techniques used in the system.

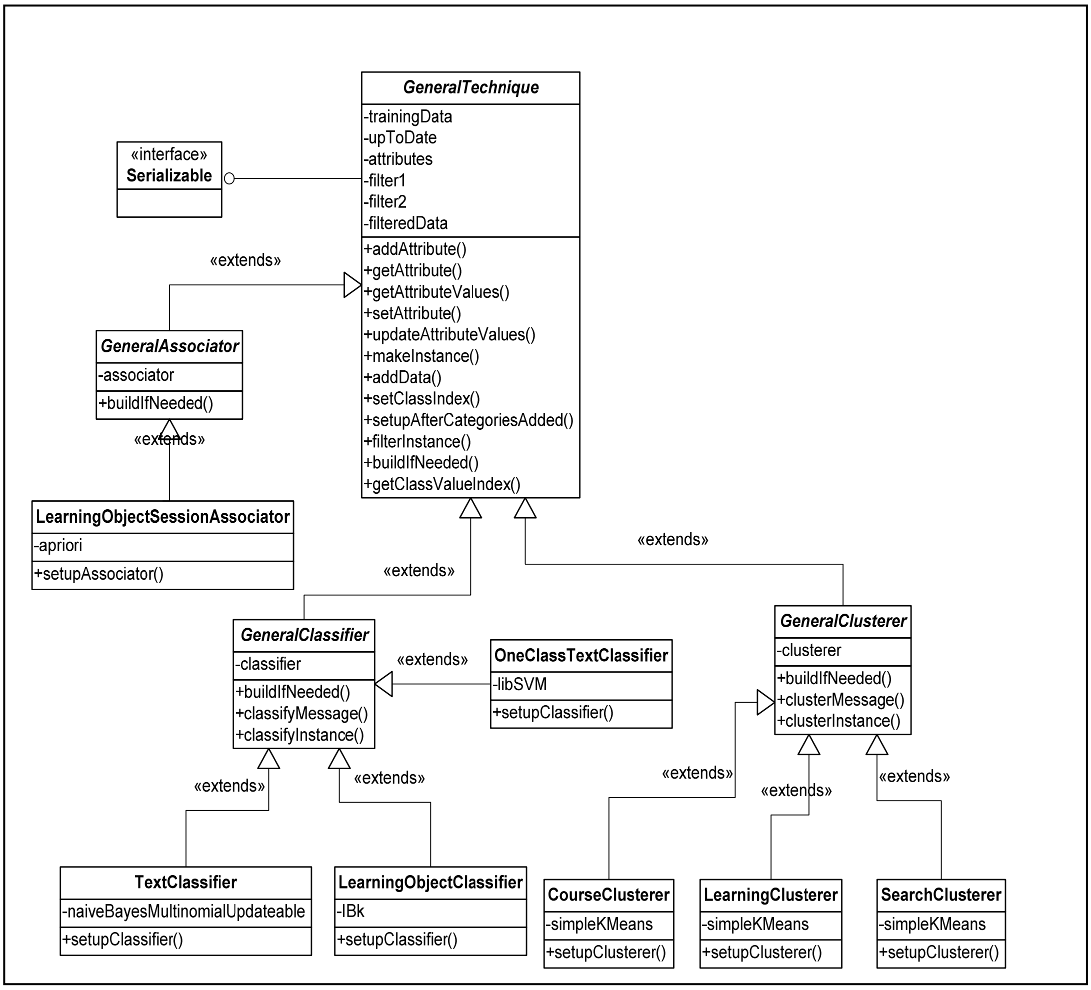


Figure 26: Class Diagram of data mining techniques.

Abstract class GeneralClassifier is a general class for all classifiers. The class has classifier attribute to define which classifier to use. GeneralClassifier class has three methods: buildIfNeeded, classifyMessage, and classifyInstance. The method buildIfNeeded builds classifier before using it (if not already built). On the other hand, method classifyMessage classifies instance by returning probability of instance in all available classes. Third method is classifyInstance which classifies instance by return class with most probability.

The class TextClassifier is used for context model, and has naiveBayesMultinomialUpdateable attribute to define which classifier to use for context model, and has setupClassifier method to configure classifier attributes.

The class LearningObjectClassifier is used for learning objects selection model that uses nearest neighbour search, and has IBk attribute to define which classifier to use for nearest neighbour search, and has setupClassifier method to configure classifier attributes.

The class OneClassTextClassifier is used for learning interests and search objectives models, and has libSVM attribute to define which classifier to use for one-class classification models, and has setupClassifier method to configure classifier attributes.

Abstract class GeneralClusterer is a general class for all clustering techniques. The class has clusterer attribute to define which clusterer to use. This class has three sub-classes: CourseClusterer, LearningClusterer, and SearchClusterer. GeneralClusterer class has three methods: buildIfNeeded to build clusterer before use it (if not already built), clusterMessage method to cluster instance by return probability of instance in all available clusters, and the method clusterInstance to cluster instance by return cluster with most probability.

Classes (CourseClusterer, LearningClusterer, and SearchClusterer) are used for relations recommendation model, and has simpleKMeans attribute to define which clusterer to use for relations recommendation model, and has setupClusterer method to configure clusterer attributes.

Appendix 1 lists of all attributes for all data mining techniques with their types and description. These attributes are configured in setup method in leave sub-class.

Ranking and Recommendation Classes:

Classes in Ranking and Recommendation modules outlined in Table 4, and explained below in more details.

Table 4: Ranking and Recommendation classes.

Class Name	Class Description
RankRecommend Criteria	This is the core class in ranking and recommendation process. System handles ranking and recommendation by the same way because details will be in handler classes.
ContextModelHa ndler	Rank learning objects using context model.
LearningInterest ModelHandler	Rank learning objects using learning interests model.
SearchObjective ModelHandler	Rank learning objects using search objectives model.

Class Name	Class Description
LearningObjectSessionAssociatorHandler	Recommend learning objects using learning objects recommendation model.
UserClustererAssociatorHandler	Recommend learning objects using relations recommendation model.

Class RankRecommendCriteria has calculateCriteria that will receive criteriaExpression, userId, institutionId, and list of learningObjects. The system will use these input parameters and do five main steps to calculate ranking or recommendation. First step is to parse criteriaExpression and find all formulas need to be calculated. Second step is finding all fields for each formula. Third step is find field values for each field, and pass this info to specific handler associated with each formula. Fourth step is use handler to calculate rank for each learning object in case of ranking, or recommend learning objects in case of recommendation. Last step will be applied if function is ranking, and in this case system will do three actions. First rank action is to calculate the final rank using criteriaExpression for all formulas and fields. After that, ranks will be changed for all learning objects into scale from (0) to (5). Learning objects with maximum rank will be assigned to (5) rank, and learning object with minimum rank will be assigned to (0) rank. The concept in reference (Scale range of numbers, 2011) has been implemented. Last ranking action is sorting list of learning object descending according to rank by place high rank first.

Class ContextModelHandler has calculateRank method that receives context model object (built by GAS), userId, institutionId, and list criteriaFieldValues (title and description of learning objects). The system will classify title and description of learning object using context model, then find the calculated probability for contexts users is enrolled in, and finally find summation of rank result from title and description to have final rank.

Class LearningInterestModelHandler has calculateRank method that receives learning interests model object for a user (built by GAS), userId, institutionId, and list criteriaFieldValues (learning object title). The system will classify title of learning object using learning interests model. This classification model returns (1) if learning object is within user's learning interests or (0) if not. Final rank will be (0) or (1).

Class SearchObjectiveModelHandler has calculateRank method that receives search objectives model object for a user (built by GAS), userId, institutionId, and list criteriaFieldValues (learning object title). The system will classify title of learning object using search objectives model. This classification model returns (1) if learning object is within user's search objectives or (0) if not. Final rank will be (0) or (1).

Class LearningObjectSessionAssociatorHandler has findAssociations method that receives learning objects recommendation model object (empty model) and list criteriaFieldValues (IDs of learning objects and their HTTP sessions for all learning objects that were selected together with a specified learning object). The system will add this information into the provided empty associator, and find the best association rules. System will load information about learning objects found in association rules, and return them for the user.

Class UserClustererAssociatorHandler has findAssociations method that receives association model object (empty model) and list criteriaFieldValues (IDs of learning objects and users for all learning objects were selected by peer users that defined by relations recommendation model). Relations between users will be defined by GAS during analysis phase using relations recommendation model. The system will add this information into the provided empty associator, and find the best association rules. System will load information about learning objects found in association rules, and return them for the user.

5.3.3.3. Cache Management:

Ranking and Recommendation functions use different resources to perform their work, and may take long time to be completed. The system has cache management to improve performance of its operations.

There are three types of cache supported in the system: ranking result, recommendation result, and data mining techniques models. The system caches search keywords used by a user, and result of ranking process. If the user performed same search again, or during paging of ranked search result for same search operation, the system will use the cached ranked search result. The system recommends learning objects using relations recommendation model in every page in the system while user browsing the LMS. List of recommended learning object may change every one hour (configurable period). So, there is no need to find recommended learning objects for every page, and system caches this recommendation list, and use it for all pages. Third cache type is for data mining techniques models, and because GAS works every one hour (configurable period) to gather data and analyse it. During this time, there will be no updates on data mining models. System will deserialize data mining models and cache them in system, then use them when needed.

Cache management has configurable value to use for how much time to keep cache in system. When this period of time is elapsed, system will launch process (ranking, recommendation, or deserialize data mining model), and cache result again. Figure 27 shows the process of cache management used in system.

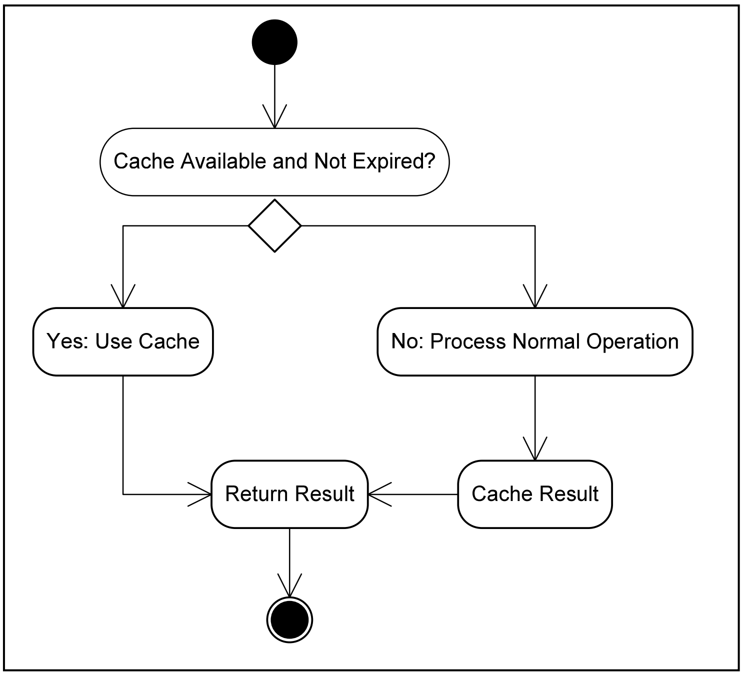


Figure 27: Cache Management Process.

5.4 Conclusion

Consolidated Ranking and Recommendation Framework (CRRF) designed to support general process in ranking and recommendation, and can use multiple data sources in its operations. The framework uses service-oriented architecture (SOA) to facilitate integration with other systems.

The CRRF consists of two parts: Gathering and Analysis System (GAS), and Ranking and Recommendation modules. GAS gathers data (Moodle2Cam), save it into CAM format, and analyses it (AutoProfileBuilder) to create user profiles in dynamic way. AutoProfileBuilder generates context model, learning interests' model, search objectives model, learning objects selection model, learning objects recommendation model, and relations recommendation model. These data mining models will be used during ranking and recommendation processes.

Ranking and Recommendation modules have flexible criteria that can use formulas to calculate ranking and find recommendations. Formulas can be configured using input parameters and database fields. These modules support cache management to improve performance of its operations.

Chapter Six

Evaluation and Results

6.1 Introduction

The evaluation approach consists of two parts. Firs part was data mining algorithms that will be used to create user profile. Second part was evaluation of ranking and recommendation framework, and the improvement to this framework made on searching of learning objects.

Evaluation of data mining algorithms includes experiments on large data to measure performance of algorithms.

Evaluation of ranking and recommendation framework includes remote usability testing for search engine, and evaluation for search result.

6.2 Data Mining Algorithms

Evaluation of data mining algorithms covered different techniques: text classification, one-class classification/anomaly detection, nearest neighbour search classification, association rules, and clustering.

Criteria selected in evaluation of data mining algorithms are the time needed to build the model, the time to use the model, and the correctness of the result. Acceptable time to build and use the model is different from one problem to another, for example, some problems like text classification and clustering will be executed to build user profile in offline mode, so it is acceptable to build and use the model within few minutes, but in the other hand, association problem will be used to recommend learning objects for the users, so there is a need to find association rules within seconds. Algorithm with higher correctness is better, but take into consideration the time needed to build the model is important in decide what algorithm to use.

We generated real usage data in Moodle (about 4,000 transactions), then duplicated it to about (130,000 transactions) to be used for evaluation of data mining techniques. Generated data were collected about activities of: 5 teachers (3 computer science, 1 chemistry, 1 business administration), 12 students (6 computer science, 3 chemistry, 3 business administration), 19 courses including full text books and lectures (10 master in computer science, 3 bachelor in computer science , 3 chemistry, 3 administration). Different levels of courses are used (bachelor and master) to increase number of participants in the evaluation, and these levels have no other special purpose in the research.

Appendix 5 shows software and hardware specifications used to execute experiments.

Same data used for all experiments within same data mining technique. Steps in experiments are: loading training data from database, build model, and finally test the model. Build model is the time needed to reflect usage data on models and make them available for ranking and recommendation modules. Test model is the time needed by ranking and recommendation modules to perform their operations, and deliver result to user.

6.2.1. Text Classification

Experiments for text classification were executed on 8,211 instances with total size 902 MB of text. These transactions generated by extracting text from full textbooks, lectures, and assignments for 19 courses.

Experiments for text classifications executed on below algorithms:

Table 5: Data mining algorithms for text classification.

Algorithm Name	Short Name
Bayes Net	BN
Complement Naïve Bayes	CNB
DMNB Text (Discriminative Multinomial Naïve Bayes)	DMNBT
Naïve Bayes	NB
Naïve Bayes Updateable	NBU
Naïve Bayes Multinomial	NBM
Naïve Bayes Multinomial Updateable	NBMU
Naïve Bayes Multinomial Text	NBMT
LibSVM (Library for Support Vector Machines)	LibSVM
SMO (Sequential Minimal Optimization)	SMO
OneR	OneR

Our analysis shows many performance factors for above algorithms such as: build model time, test model time, correctness of classification, Kappa statistic, Precision, Recall, and FP Rate (False-Positive).

Figure 28 shows comparison for build model time.

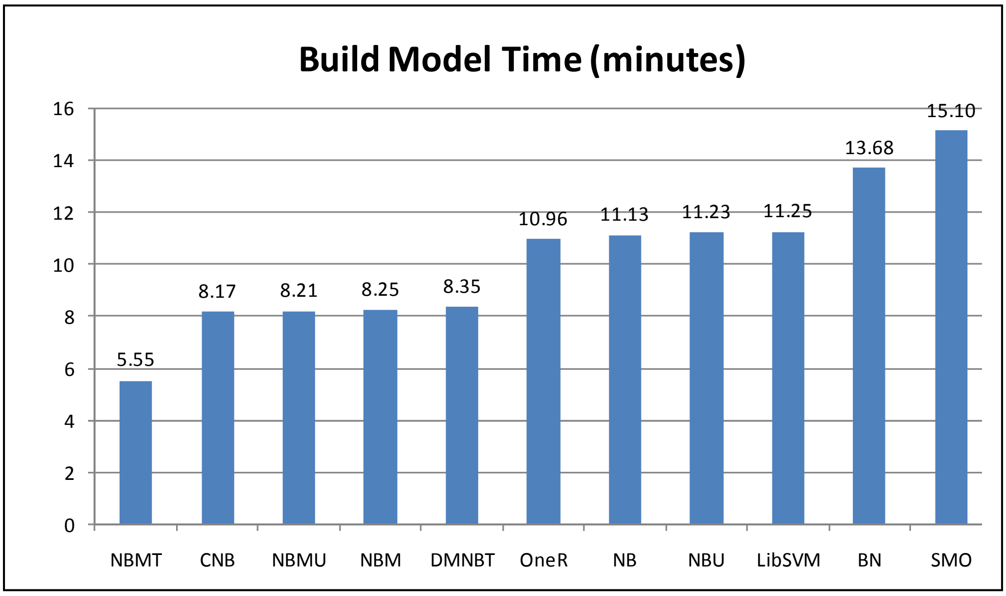


Figure 28: Build model time for text classification algorithms.

Execution time for algorithms to build model varies from 5.55 minutes for Naïve Bayes Multinomial Text algorithm as fastest algorithm to 15.10 minutes for SMO as slowest algorithm. Naïve Bayes algorithms have excellent performance, but Naïve Bayes Multinomial Text algorithm works directly with text without need to filter attributes or transfer them to different data types, so it was the fastest one. SMO is a sequential algorithm and this makes it the slowest one.

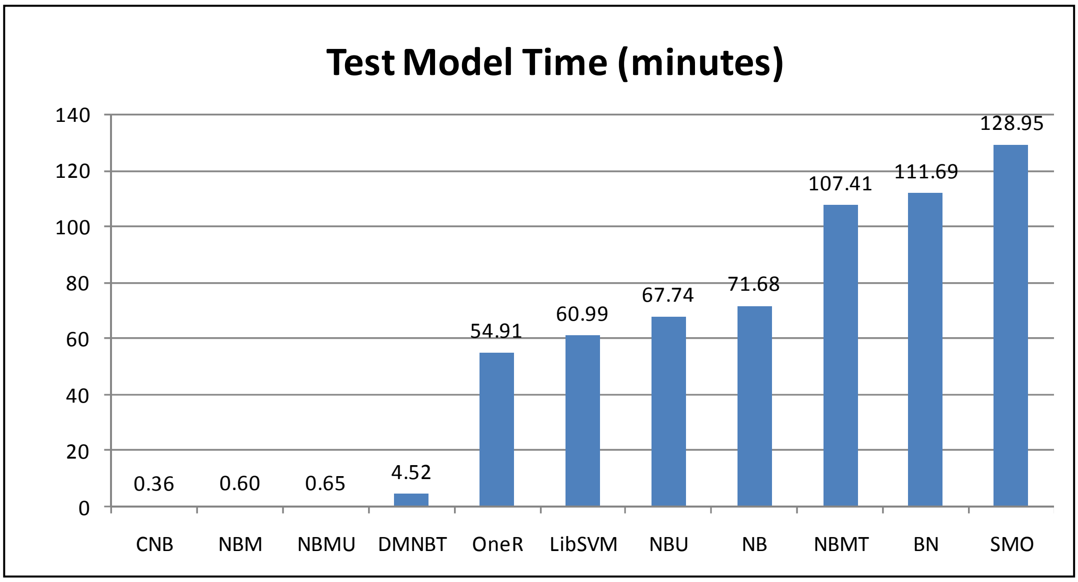


Figure 29: Test model time for text classification algorithms.

Execution time for algorithms to test model varies from 0.36 minutes for Complement Naïve Bayes algorithm as fastest algorithm to 128.95 minutes for SMO as slowest algorithm. The gap in test model time was very clear between algorithms. Three algorithms considered having acceptable time for test the model, these algorithms are: Complement Naïve Bayes, Naïve Bayes Multinomial, and Naïve Bayes Multinomial Updateable. Complement Naïve Bayes uses normalization transforms, and Multinomial algorithms depend on number of words in each document for classification. These two factors allowed these algorithms to perform testing in short time. It is clear that other algorithms especially those completed testing within more than 50 minutes are not acceptable for proposed applications.

Figure 30 shows correctness of text classification algorithms. Training set that used to build model, also used to test the model.

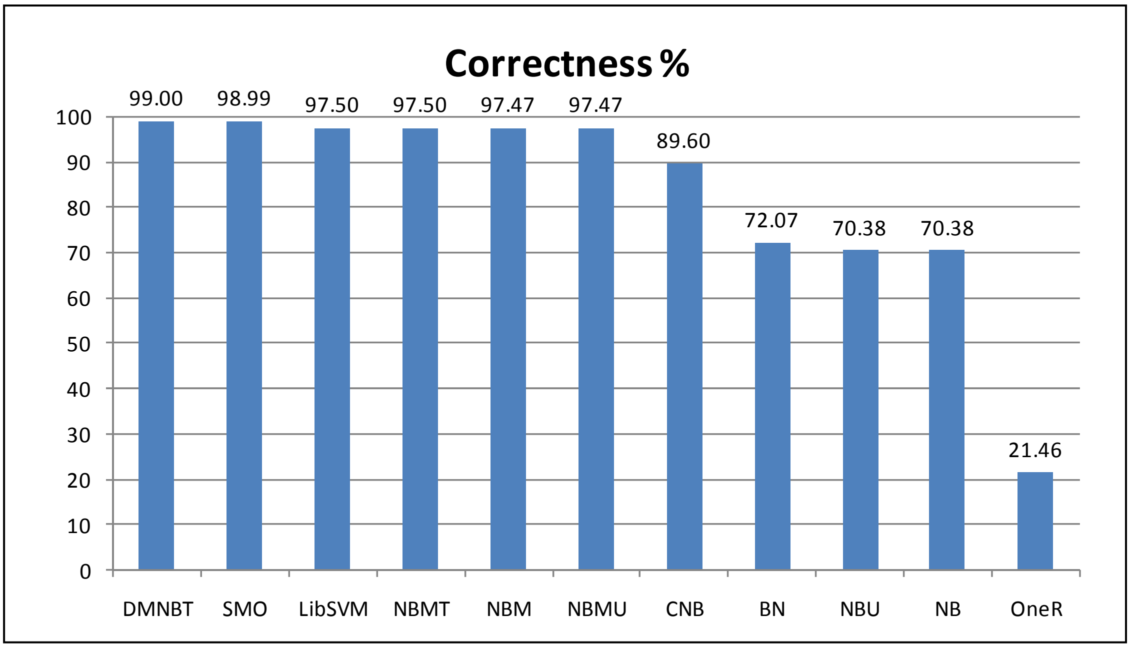


Figure 30: Correctness of text classification algorithms.

Experiments show that most accurate algorithm is DMNB Text with correctness equal to 99%. Algorithm OneR with correctness equal to 21.46% was the worse one. Multinomial algorithms (DMNBT, NBM, NBMU, and NBMT) use the number of words in each document in classification, and this allowed these algorithms to achieve high correctness. SMO algorithm achieved high correctness, but it is slow in build and test model. LibSVM is a library for support vector machines and achieved high correctness. LibSVM is faster in both build and test models than SMO, but its performance still not acceptable for proposed applications. Other Naïve Bayes algorithms (NB, NBU, BN, and CNB) failed to achieve acceptable correctness because these algorithms do not consider number of words during classification (not Multinomial). OneR algorithm is a simple algorithm, and couldn't achieve good correctness for text classification.

Table 6 shows many factors that result for experiments executed on algorithms.

Table 6: Experiments result of text classification algorithms.

Classifier	**Build Model Time (minutes)**	**Test Model Time (minutes)**	**Correctness %**	**Kappa statistic**	**Precision**	**Recall**	**FP Rate**
DMNBT	8.35	4.52	99.00	0.989	0.991	0.990	0.002
SMO	15.10	128.95	98.99	0.989	0.990	0.990	0.002
LibSVM	11.25	60.99	97.50	0.973	0.979	0.975	0.004
NBMT	5.55	107.41	97.50	0.973	0.978	0.975	0.002
NBM	8.25	0.60	97.47	0.973	0.975	0.975	0.003
NBMU	8.21	0.65	97.47	0.973	0.975	0.975	0.003
CNB	8.17	0.36	89.60	0.888	0.969	0.896	0.002
BN	13.68	111.69	72.07	0.700	0.896	0.721	0.019
NBU	11.23	67.74	70.38	0.684	0.911	0.704	0.012
NB	11.13	71.68	70.38	0.684	0.911	0.704	0.012
OneR	10.96	54.91	21.46	0.089	0.075	0.215	0.129

6.2.2. One-Class Classification / Anomaly Detection

Experiments for One-Class classification / anomaly detection executed on 68,256 instances with total size 903 MB of text. These transactions generated by interaction of 17 users with Moodle and search engine.

Experiments for one-class classifications executed on below algorithms:

Table 7: Data mining algorithms for one-class classification.

Algorithm Name	Short Name
LibSVM One-Class (Library for Support Vector Machines)	LibSVM
One Class Classifier	OneClass

Our analysis shows many performance factors for above algorithms such as: build model time, test model time, and correctness of classification.

Figure 31 shows comparison for build model time.

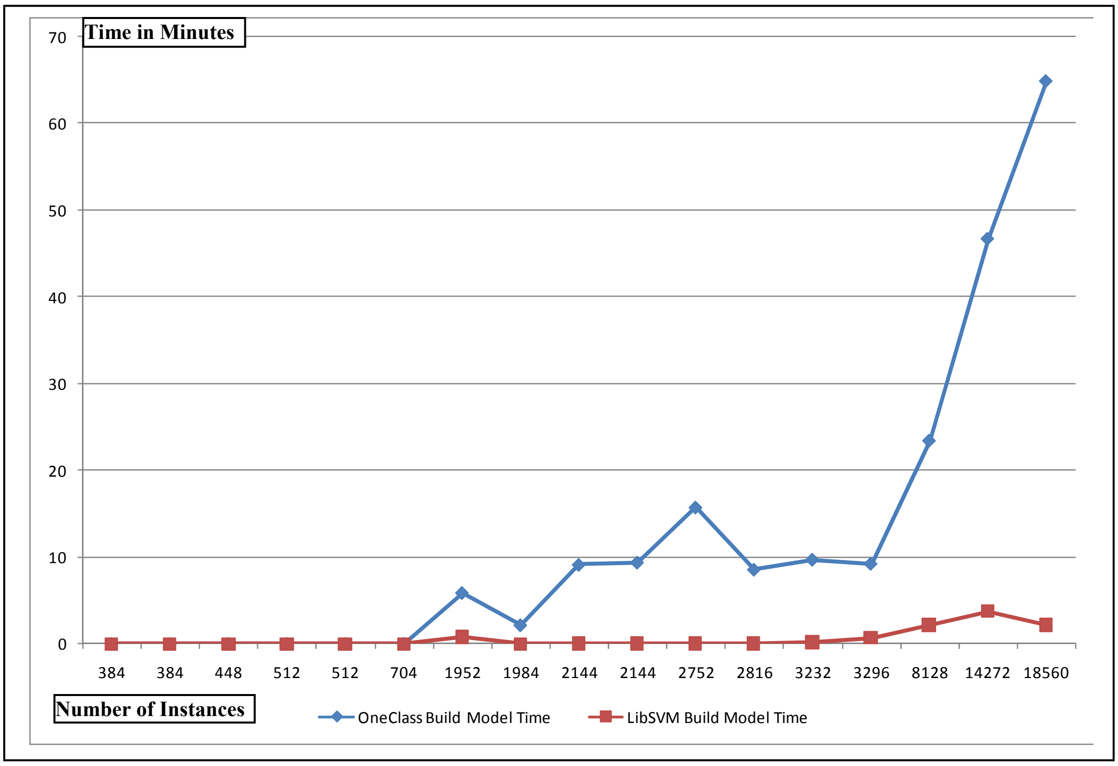


Figure 31: Build model time for one-class classification algorithms.

Algorithms used to build learning interests and search objectives for 17 users. In figure, x-axis shows number of instances, and y-axis shows execution time in minutes. It is clear from figure that LibSVM is scalable, and can build model in short time for different number of instances. In comparison, One-Class algorithm became very slow when deal with big number of instances. LibSVM is a library for support vector machines, originally designed in C++ language, and them it has been translated to Java. Figure 32 shows comparison for test model time.

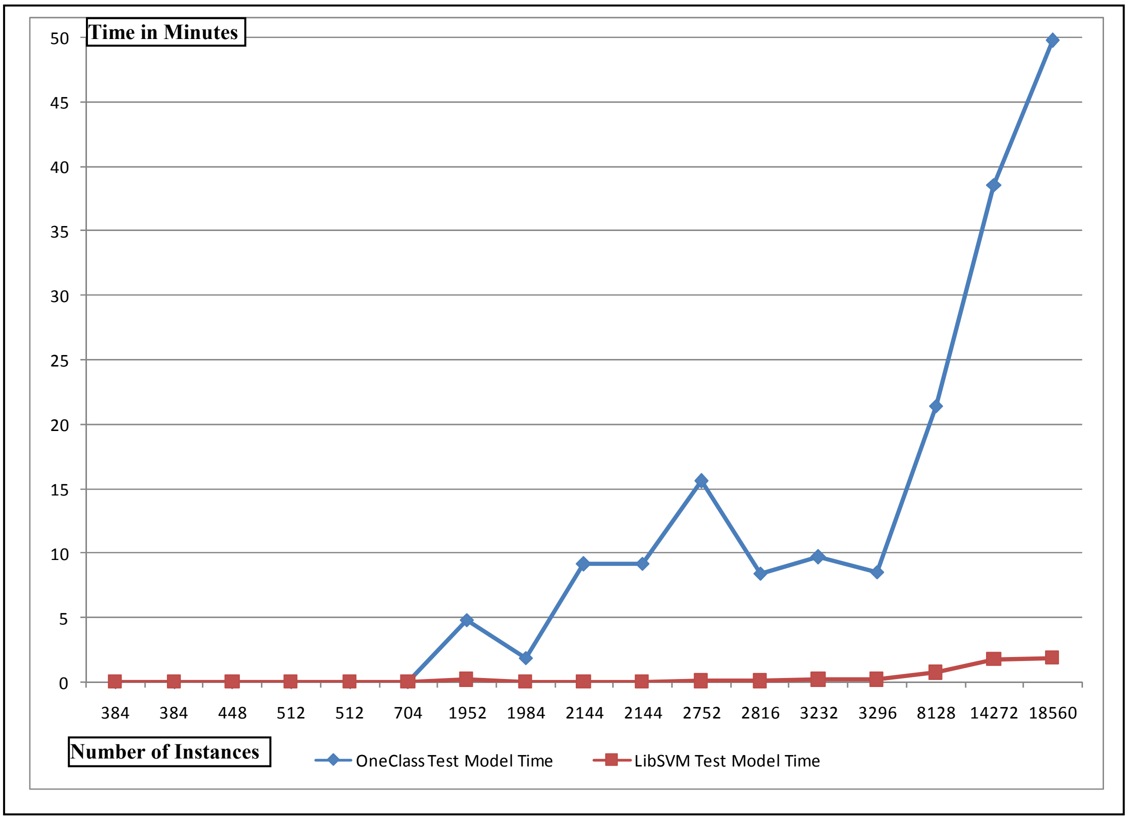


Figure 32: Test model time for one-class classification algorithms.

In figure, x-axis shows number of instances, and y-axis shows execution time in minutes. It is clear from figure that LibSVM is scalable, and can test model in short time for different number of instances. In comparison, One-Class algorithm became very slow when deal with big number of instances.

Figure 33 shows correctness of one-class classification algorithms. Training set that used to build model, also used to test the model.

Correctness doesn't depend on number of instances, but instead it depends on data itself. But in general, Figure 33 shows that LibSVM succeeded to achieve better correctness from One-Class classifier. These algorithms may produce more accurate results if system provided them with information about other classes. But for our applications (learning interests and search objectives of a user), it is not possible to add information about other users because interests and objectives of users may intersect.

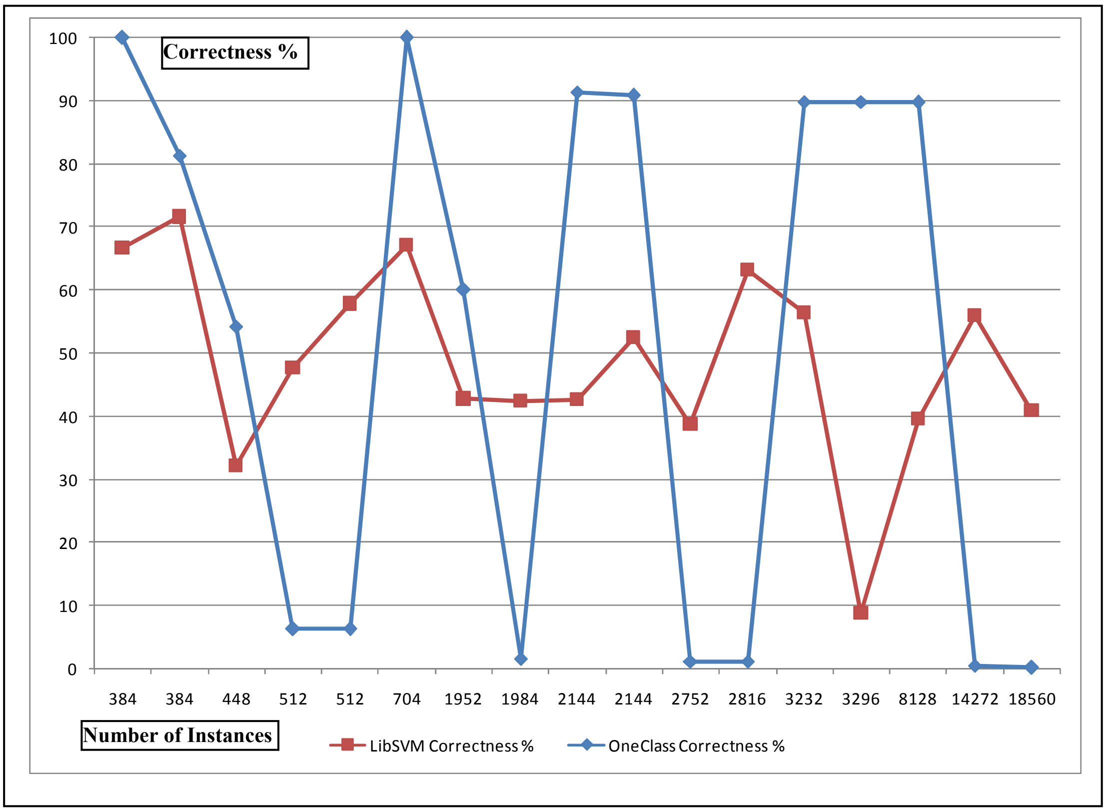


Figure 33: Correctness of one-class classification algorithms.

6.2.3. Nearest Neighbour Search Classification

Experiments of nearest neighbour search executed on 45,184 instances. Each transaction has information about (context, language, country, IP address, and selected learning object). The technique will try to find nearest learning objects using context, language, country, and IP address. Found nearest learning objects will be recommended for a user.

Experiments for nearest neighbour search executed on below algorithms:

Table 8: Data mining algorithms for nearest neighbour search.

Algorithm Name	Short Name
MLP Classifier (ANN) (Multi Layer Perceptron)	MLPClass
RBF Classifier (Radial Basis Function Networks Classifier)	RBF
Multilayer Perceptron (ANN)	MLP
IBk (Instance-based K-nearest neighbours classifier)	IBk
KStar	KStar
LBR (Lazy Bayesian Rules)	LBR
LWL (Locally Weighted Learning)	LWL
Citation kNN	CitationKNN

Figure 34 shows comparison for build model time.

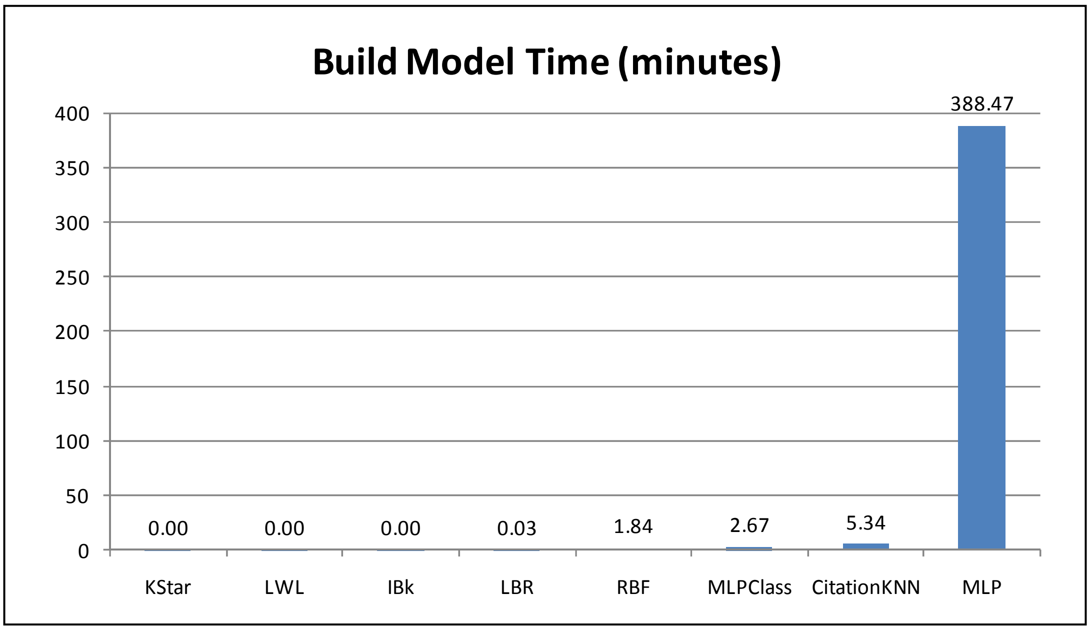


Figure 34: Build model time for nearest neighbour search algorithms.

Execution time to build model for lazy algorithms (KStar, LWL, IBK, and LBR) was zero, because these algorithms do not build module at all, and wait until receive request to calculate nearest neighbour. Multilayer Perceptron (ANN) algorithm was very slow to build model in comparison with other algorithms due to big number of nominal values.

Figure 35 shows comparison for test model time.

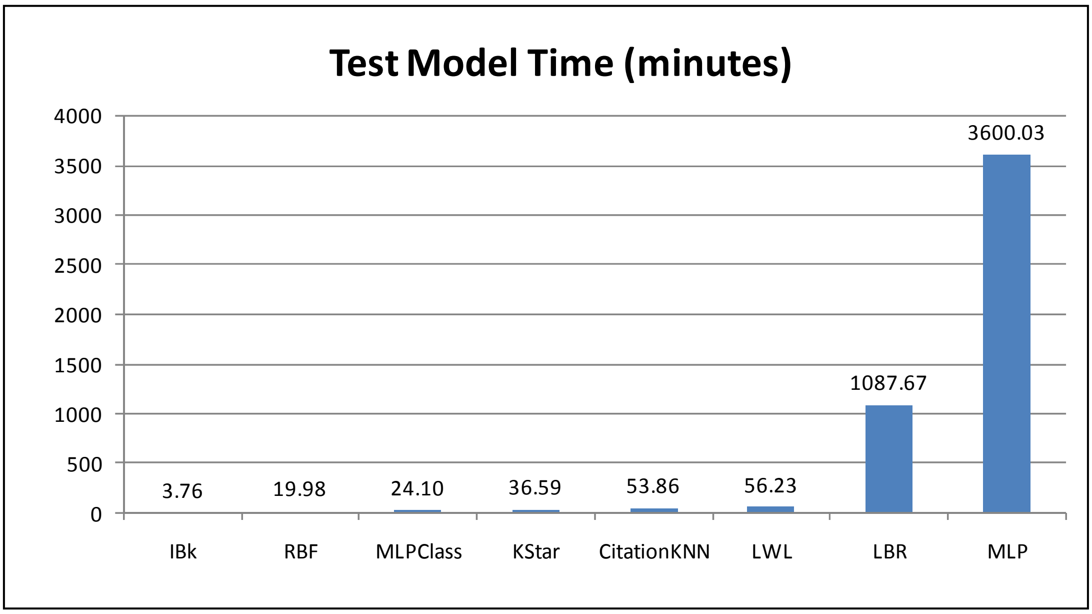


Figure 35: Test model time for nearest neighbour search algorithms.

Execution time for algorithms to test model varies from 3.76 minutes for IBk as fastest algorithm to 3600.03 minutes for Multilayer Perceptron (ANN) as slowest algorithm. Algorithm LBR completed within 1087.67 minutes, which also considered as very slow. Other algorithms completed in less than 57 minutes.

Table 9 shows build model time and test model time for nearest neighbour search algorithms.

Table 9: Experiments result of nearest neighbour search algorithms.

Nearest Neighbour Search	Build Model Time (minutes)	Test Model Time (minutes)
IBk	0.00	3.76
RBFClassifier	1.84	19.98
MLPClassifier	2.67	24.10
KStar	0.00	36.59
CitationKNN	5.34	53.86
LWL	0.00	56.23
LBR	0.03	1087.67
Multi Layer Perceptron	388.47	3600.03

6.2.4. Association Rules

Experiments for association rules have been executed on 8,864 instances. Each transaction has information about (session and selected learning object). The technique tried to find most selected learning objects within same session. All algorithms have been configured to find best (100) association rules using minimum support equal to (0.003). Experiments for association rules executed on below algorithms:

Table 10: Data mining algorithms for association rules.

Algorithm Name	Short Name
Apriori	Apriori
Filtered Associator	FA
FPGrowth	FPGrowth
GSP (Generalized Sequential Patterns)	GSP
HotSpot	HotSpot
Predictive Apriori	PA
Tertius	Tertius

Algorithm GSP (Generalized Sequential Patterns) failed to build model within 67 hours. So it was cancelled and not included in below results.

Figure 36 shows comparison for build model time.

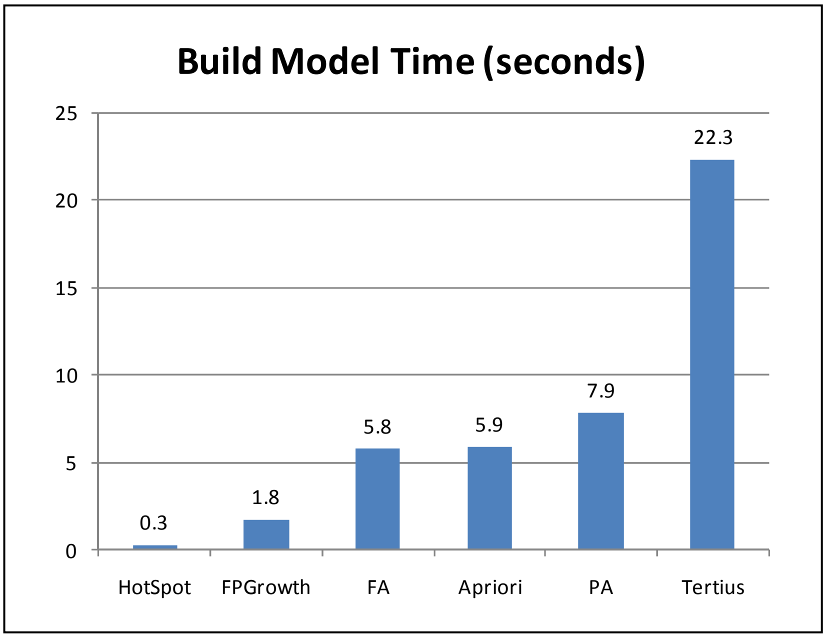


Figure 36: Build model time for association rules algorithms.

This data mining problem doesn't need a lot of time to be completed. So, most algorithms completed in short time. But still 22 seconds used to build model for Tertius algorithm is long time, especially that online calculation for association rules will improve recommendation function and make it near real time.

Figure 37 shows comparison for test model time. Fastest algorithm (build and test model) was HotSpot, and slowest algorithm was Tertius. Almost order of algorithms to build and test model was the same.

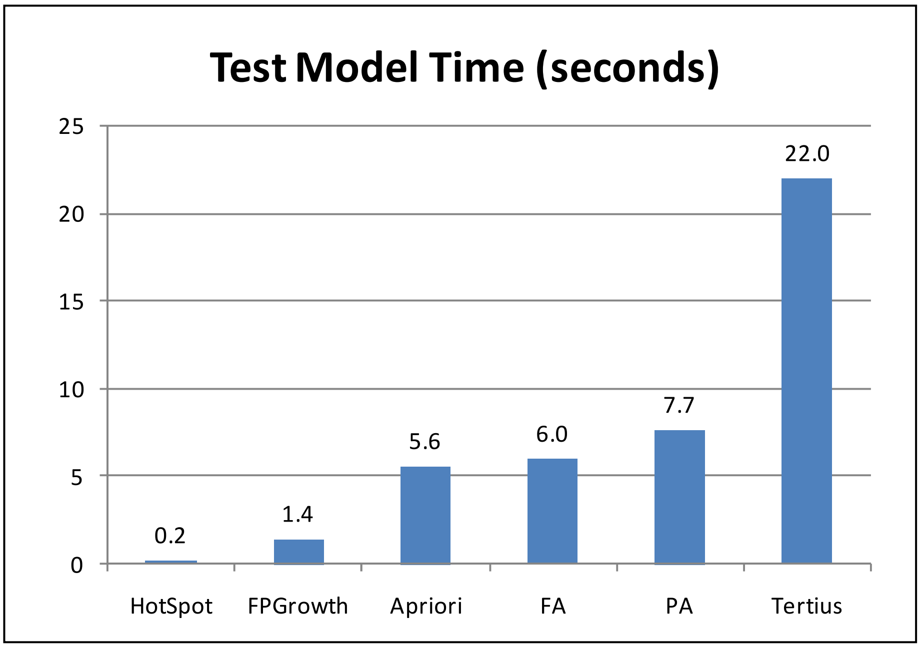


Figure 37: Test model time for association rules algorithms.

Figure 38 shows correctness of association rules algorithms. Correctness is the ability of the algorithm to find the correct (100) association rules. Some algorithms found (100) association rules, but failed to found the correct frequent patterns.

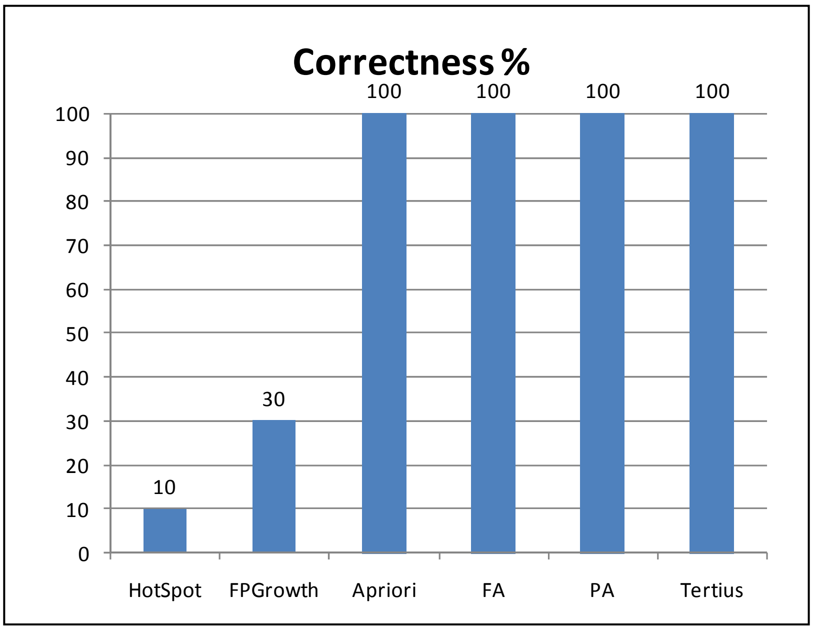


Figure 38: Correctness of association rules algorithms.

HotSpot and FPGrowth algorithms failed to find the correct (100) association rules, but all other algorithms (Apriori, FA, PA, and Tertius) succeeded to find the required (100) association rules correctly.

Table 11 shows build model time, test model time, and correctness for association algorithms.

Table 11: Experiments result of association rules algorithms.

Associator	Build Model Time (seconds)	Test Model Time (seconds)	Correctness %
HotSpot	0.3	0.2	10
FPGrowth	1.8	1.4	30
Apriori	5.9	5.6	100
Filtered Associator	5.8	6.0	100
Predictive Apriori	7.9	7.7	100
Tertius	22.3	22.0	100

6.2.5. Clustering

Experiments for clustering executed on 399,648 instances. Learning interests cluster (LMS actions) has been used in these experiments. Each transaction has information about (event name, event type, IP address, language, country, and user). The technique tried to cluster interactions with learning management system, and use it to decide the relation level between users. All algorithms were configured to distribute transactions into 200 clusters.

Experiments for clustering executed on below algorithms:

Table 12: Data mining algorithms for clustering.

Algorithm Name	Short Name
Cascade Simple K Means	CSKM
CLOPE	CLOPE
Cobweb	Cobweb
DBScan	DBScan
Simple EM (Expectation Maximisation)	EM
Farthest First	FF
Filtered Clusterer	Filtered
Hierarchical Clusterer	Hierarchical
LVQ (Learning Vector Quantization)	LVQ
Make Density Based Clusterer	MDBC
OPTICS	OPTICS
Self Organizing Map	SOM
Simple K Means	SKM
XMeans	XMeans

Algorithm (Hierarchical Clusterer) failed to build model due to heap space error. Even other algorithms succeeded to perform operation using same heap space. Below algorithms failed to build clusterer model within long time.

Algorithm Name	Failed to Build Model within (Hours)
Cobweb	23
CLOPE	45
Self Organizing Map	54
OPTICS	63

Two algorithms, DBScan and LVQ took very long time to build the model, 1049.8 and 642.25 minutes respectively. Algorithms (FF, FC, and Simple K Means) succeeded to build model in less than one minute. Other algorithms (MDBC, CS K Means, EM, and XMeans) built the model within 2.26 to 24.21 minutes.

Figure 39 shows build model time for clustering algorithms.

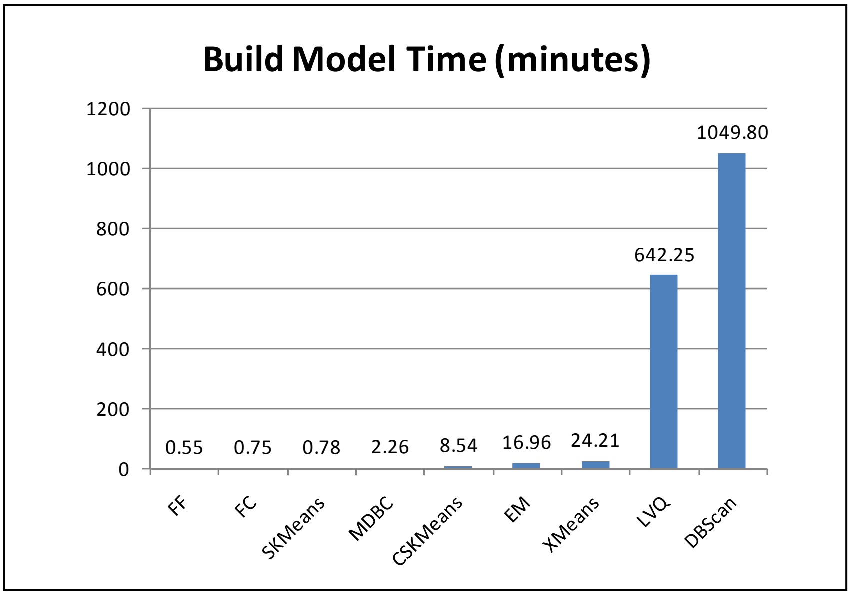


Figure 39: Build model time for clustering algorithms.

Figure 40 shows test model time for clustering algorithms.

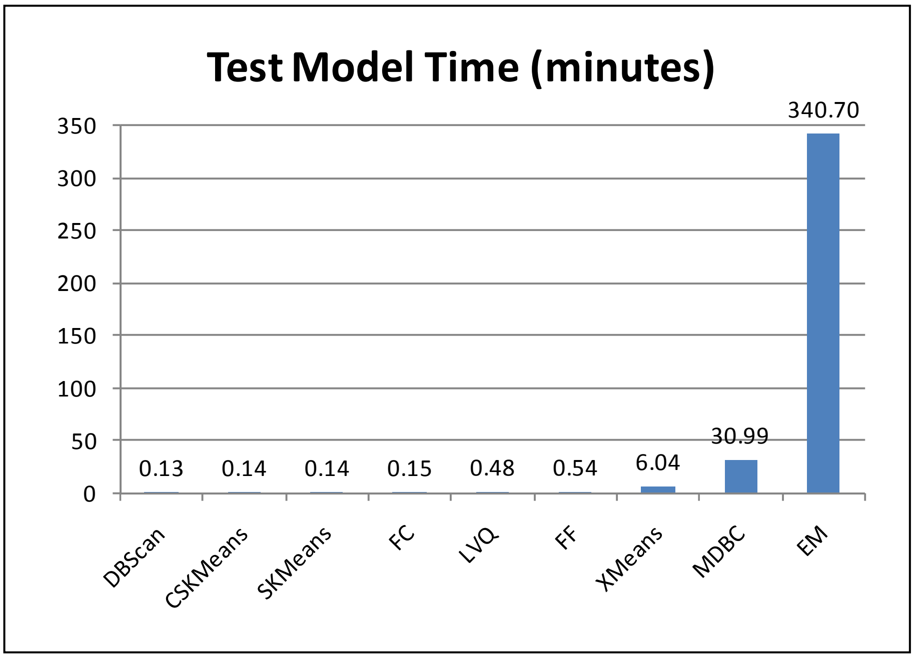


Figure 40: Test model time for clustering algorithms.

EM algorithm took very long time (340.7 minutes) to test the model. Algorithms (DBScan, CS K Means, Simple K Means, FC, LVQ, and FF) succeeded to test model in less than one minute. Other algorithms (XMeans, MDBC) built the model within 6.04 to 30.99 minutes.

All experiments configured to distribute data into 200 clusters. So accuracy of algorithms can be measured by find its ability to distributes data into these clusters. Figure 41 shows number of clusters that each algorithm succeed to generated and distribute data into them.

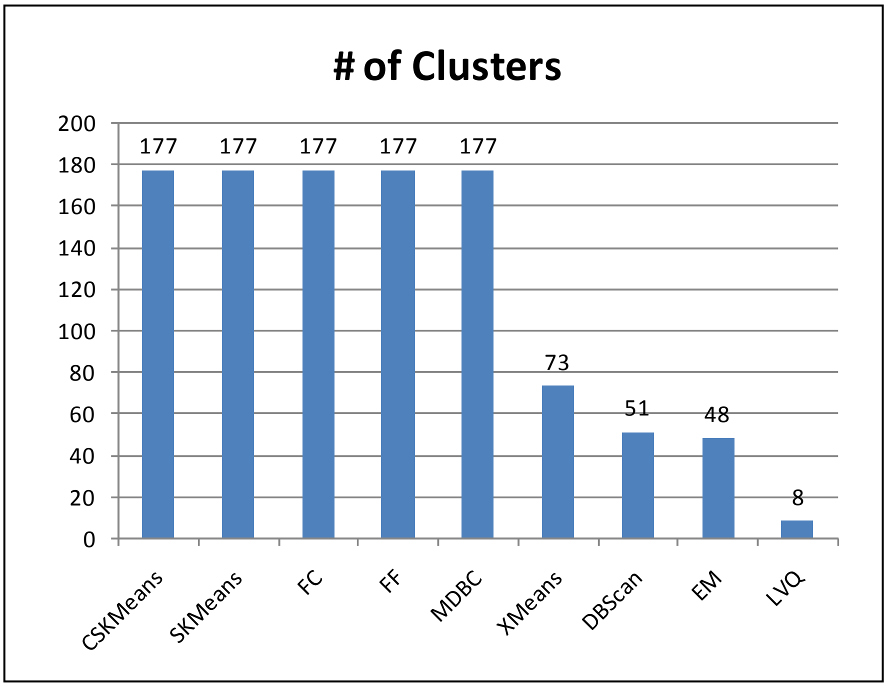


Figure 41: Number of created clusters using clustering algorithms.

Algorithms (CS K Means, Simple K Means, FC, FF, and MDBC) generated 177 clusters with similar number of instances stored in each cluster. Other clusters generated from 73 down to 8 clusters.

Table 13 shows build model time, test model time, and number of clusters that algorithm succeeded to generate them.

Table 13: Experiments result of clustering algorithms.

Clusterer	Build Model Time (minutes)	Test Model Time (minutes)	# of Clusters
CascadeSimpleKMeans	8.54	0.14	177
SimpleKMeans	0.78	0.14	177
FilteredClusterer	0.75	0.15	177
FarthestFirst	0.55	0.54	177
MakeDensityBasedClusterer	2.26	30.99	177
XMeans	24.21	6.04	73
DBScan	1049.80	0.13	51
EM	16.96	340.70	48
LVQ	642.25	0.48	8

6.3 Consolidated Ranking and Recommendation Framework

CRRF system has been published in internet for evaluation. CRRF configured to use Moodle system that has 13 courses in computer science (4 courses bachelor level, and 9 courses master level). The system has been used by 15 users, and each user enrolled in two courses. Evaluation guide in Appendix 6 was sent to users. It was required from users to use the system for short time for few days. The idea was to allow system gathers and analyses users' activities and include this usage data in user profiles and data mining models. During evaluation, users requested to do manual evaluation for learning objects to allow us later to compare the manual ranking with automatic ranking provided by the system, to calculate precision and recall. At the end of evaluation, users requested to fill survey in Appendix 7 that has SUS (System Usability Scale) survey to measure system usability, and questions about their opinion on importance of different factors on ranking and recommendation.

6.3.1. Precision and Recall

Precision and recall are used to measure accuracy in information retrieval. Precision is percentage of retrieved instances that are relevant from all retrieved instances. Recall is percentage of relevant instances that are retrieved from all relevant instances. (Bidgoli, 2003)

Precision and recall can be calculated by TP (True Positive are cases were positive and predicted positive), FP (False Positive are cases were negative but predicted positive), TN (True Negative are cases were negative and predicted negative), and FN (False Negative are cases were positive and predicted negative). (Precision and Recall Calculation, 2013)

According to above definitions:

$$Precision = TP / (TP + FP)$$

$$Recall = TP / (TP + FN)$$

In proposed framework, calculating of precision and recall is a bit hard. Ranking function within framework gives rank for each learning object (from 0 to 5, 0 is not relevant, and 5 is the most relevant), and then sort these objects and return most relevant in the beginning. Precision and recall will be calculated according to their successful of providing the correct rating according to user profile.

Users of the system requested to check rating given by the system, and then to give their own ranting. At the end of evaluation, ranking given by system is compared with ranking given by human to calculate precision and recall.

Table 14: Precision and Recall calculation parameters.

Parameter	**Condition**
TP	SR between (HR - 0.5) and (HR + 0.5)
FP	SR > (HR + 0.5)
FN	SR < (HR - 0.5)

Where;
TP: True Positive
FP: False Positive
FN: False Negative
SR: System Rank
HR: Human Rank

Table 14 shows how to find if a learning object that showed to a user is TP, FP, or FN. Condition column in above table needs rank done by system, and the rank provided by human. These two ranks evaluate the relevance of learning object for a user. True positive learning objects are those objects got ranking by system very close to ranking given by a user. System ranks learning objects using real numbers with fractions, but system shows rank for a user by round it for closest number from (0) to (5). So to avoid this confusion, we added (0.5) to human rank range. False positive learning objects are those objects that given rank by human lower than rank given by the system, so they are considered as relevant by system, but actually user evaluated them as not relevant. False negative learning objects are those objects that given rank by human higher than rank given by the system, so they are considered as not relevant by system, but actually user evaluated them as relevant.

Fifteen users participated in evaluation of the framework, and during evaluation period, users performed 159 search operations, and evaluated 281 learning objects.

Figure 42 shows precision scores for all participated users.

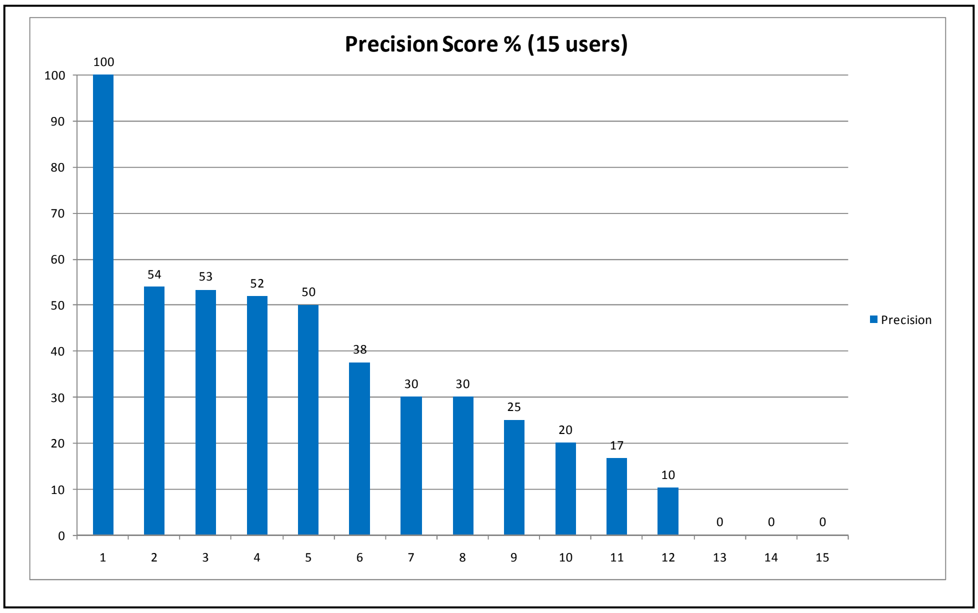


Figure 42: Precision score for ranking system.

The average precision score for all users was 32%. According to (Govaerts, El Helou, Erik, & Gillet, 2011), precision for Google during evaluating their search widget was 65%. Most users mentioned that number of provided learning objects in search result was limited; also quality of learning objects was poor. Evaluating the used repository and quality of learning objects were not part of this thesis. Comparison with Google is just to give indication about achieved level of precision for this research, but there are a lot of differences between proposed framework ranking and Google search engine that making direct comparison is not correct.

The average recall score for all users was 51%. This is percentage of relevant instances that are retrieved from all relevant instances.

Figure 43 shows recall scores for all participated users.

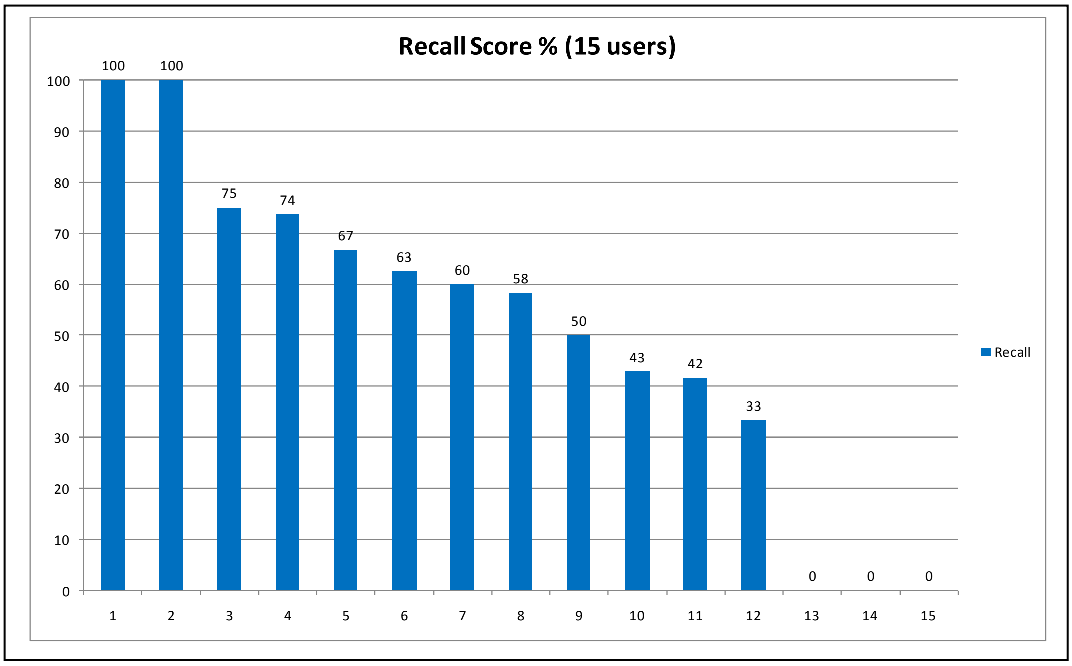


Figure 43: Recall score for ranking system.

6.3.2. SUS (System Usability Scale)

Real fifteen users were asked to use the system for few days, and perform search queries, evaluate ranking and recommendation results, and finally to fill SUS survey. Instructions and information were sent to users by email, and we met face to face with nine of them, and explained what exactly was required.

The purpose from this survey is to measure the usability of the search engine. Interface of search widget has been developed by eummena.org, and during work on this thesis, ranking and recommendation functions were added to it. Usability of search widget was not measured before, so we cannot check if added ranking and recommendation functions changed the usability of the widget. The purpose of this survey is to measure usability score of current search widget after add ranking and recommendation, and collect comments from users to improve it. This score will be used in future researches to measure improvement on the search widget.

Total average score for all users is 64.5%. According to (Sauro, 2011), the average score for 500 studies is 68%.

Figure 44 shows score for all users.

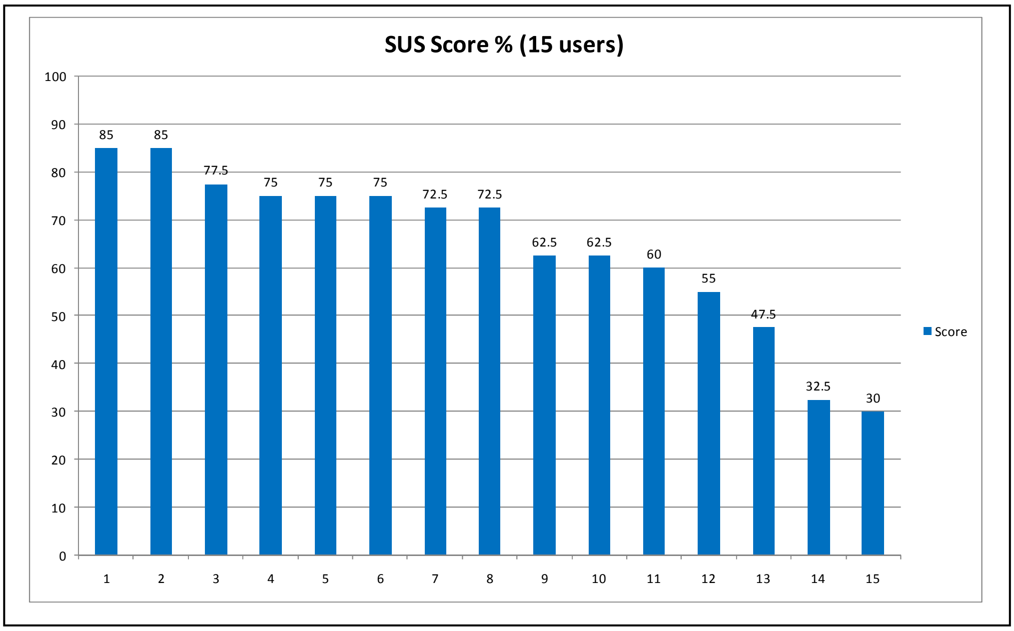


Figure 44: SUS Score for all users.

We can see that three of fifteen users voted for score less than 50, but other thirteen users voted with score higher than 50. If we calculated average score for top thirteen users, total average score for their SUS is 71.5%.

Users' comments about the search widget are useful, and may improve the widget usability, and achieve higher SUS score in next version. When a user performs a search and the system shows the result, the user may confuse result of search with list of learning objected mentioned in (Recommended by Peer Users) section. Both results exist in same column. Some users prefer to split these two lists from each other. Second comment is when the system shows loading icon during ranking process. Some ranking processes may take up to 20 seconds. Users prefer to show progress bar instead of loading icon, where progress bar will show, in roughly, the percentage of completion in ranking process. Third comment is about the main page, search result shows first six items, and gives user link to (More Results...) to allow him view rest of the result. The link (More Results...) uses same format and colour like normal search result in first six items. It is better to change the format of this link to make it easier to be distinguished. Fourth comment is about manual ranking, that available for users in details page, uses five starts. For some learning objects, system will show the first four stars in one row and the fifth start in another row. View starts in two rows may confuse the user, and let him rank learning object as level 4, while he is thinking that he rated object with the best score. Fifth comment is about the option in details page that allow user to choose section and do import for it. It is not clear for what purpose this function can be used. Another comment is about More Results page which has no text search. Provide this page with simple text search function will improve search within search result. Last comment is that users cannot control number of results to show in each page. Allow users to do that may help them to show number of results they prefer, and find what they want easier.

6.3.3. Ranking and Recommendation Factors

Three factors were used in framework to rank learning objects. These factors are: contexts (courses users enrolled in), learning interests (interactions with learning management system), and search objectives (keywords used during search).

Current framework uses below formula to calculate the total rank:

Total Rank = F1 + F2 + (F3 * 0.025) + (F4 * 0.025)

F1 is the rank of learning object title using context model. F2 is the rank of learning object description using context model. F3 is the rank of learning object title using learning interests model. F4 is the rank of learning object title using search objectives model. In current framework, F3 (learning interests) and F4 (search objectives) have 5% weight. While F1 and F2 (context) have 95% of the weight. The main reason for this formula is the high correctness for text classification algorithm used for context model, and the low correctness for one-class classification used for learning interests and search objectives models.

There was part in survey to allow users evaluate the importance of these factors in rank learning objects.

Figure 45 shows the results of users' evaluation for importance of ranking factors. It shows similar importance for all ranking factors.

Three recommendation factors were used in the framework to recommend learning objects. These factors are: learning objects selected by users enrolled in the same courses, learning objects selected by users with similar learning interests, and learning objects selected by users with similar search objectives. Recommendation criteria use equal weight for all recommendation factors.

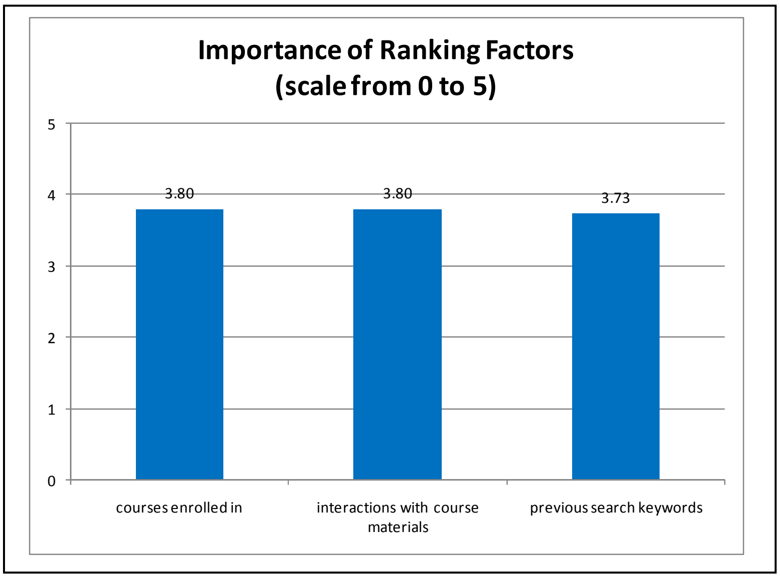


Figure 45: Importance of Ranking Factors.

Figure 46 shows evaluation of users for importance of recommendation factors. It shows similar importance for all recommendation factors.

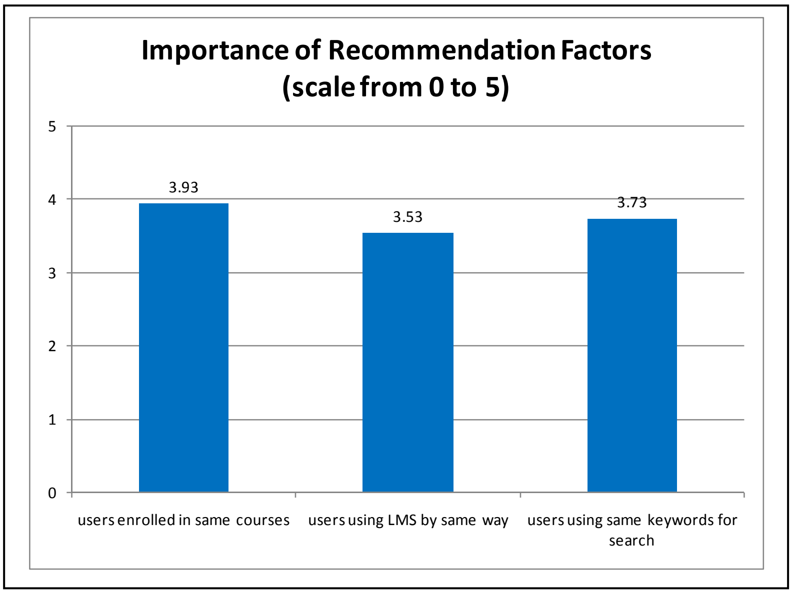


Figure 46: Importance of Recommendation Factors.

6.4 Comparison with Related Work

This section compares the proposed framework with related work that is mentioned in the previous chapter. The comparison is based on eight factors: ranking, recommendation, based on usage data, consolidated framework, user Profiles, data mining, evaluation using SUS, and evaluation using precision and recall. These factors are the main features in the proposed framework.

Most other researchers have only one feature of ranking or recommendation, but the proposed work has both of these features with clear definition for them. On the other hand, the proposed framework is the only system has the consolidated feature that allows combine multiple works into one single system. Some of other frameworks proposed the user profile as part of ranking or recommendation processes, but no one of them proposed clear properties for the profile, and how to map these properties to usage data, and ranking and recommendation processes. Only two researchers from related work used data mining techniques in their frameworks, but the proposed framework concentrated on usage of data mining algorithms to build the user profile.

The proposed framework tries to include features from all related work to empower the ranking and recommendation processes, also tries to answer all pending questions remained unanswered by other researchers

Table 15 shows availability of all factors in all related work.

Table 15: Comparison between CRRF and Related Work.

Framework / Feature	**Ranking**	**Recommendation**	**Based on usage data**	**Consolidated Framework**	**User Profile**	**Data Mining**	**Evaluation using SUS**	**Evaluation using Precision and Recall**
CRRF	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Metrics for Learning Objects (Ochoa, 2008)	Yes	No	Yes	No	Yes	No	Yes	Yes
Ad Hoc Recommendation Engine (Al-Khalifa, 2008)	No	Yes	Yes	No	Yes	No	No	Yes
The 3A Recommender System (El Helou et al., 2010)	Yes	No	No	No	Yes	No	No	Yes
Hybrid Recommender (DELPHOS) (Zapata et al., 2011)	Yes	No	Yes	No	No	No	Yes	No
A Federated Search Widget (Govaerts et al., 2011)	Yes	No	Yes	No	No	No	Yes	Yes
Semantic Document Architecture (SDArch) (Nešić et al., 2011)	No	Yes	No	No	Yes	No	Yes	No

Framework / Feature	Ranking	Recommendation	Based on usage data	Consolidated Framework	User Profile	Data Mining	Evaluation using SUS	Evaluation using Precision and Recall
Multi-label Classification (Batista et al., 2011)	Yes	No	No	No	No	Yes	No	Yes
Agent-based Federated Search (AgCAT) (Barcelos & Gluz, 2011)	Yes	No	No	No	No	No	No	No
An Ontology-Based Learning Resources (Sridharan et al., 2011)	Yes	No	No	No	No	No	Yes	No
Preferred Personalization Learning Object Model (PPLOM) (Sree Dharinya & Jayanthi, 2012)	Yes	No	Yes	No	No	No	No	Yes
Clustering by Usage (Niemann et al., 2012)	Yes	No	Yes	No	No	Yes	No	No
Recommendation for Interdisciplinary Applications (Chen & Huang, 2012)	No	Yes	No	No	Yes	No	No	No

Framework / Feature	**Ranking**	**Recommendation**	**Based on usage data**	**Consolidated Framework**	**User Profile**	**Data Mining**	**Evaluation using SUS**	**Evaluation using Precision and Recall**
Semantic Web Technologies (LOFinder) (Hsu, 2012)	No	Yes	No	No	No	No	No	No
Recommendation in Adaptive E-Learning (Fouad Ibrahim, 2012)	Yes	Yes	No	No	Yes	No	No	No

6.5 Conclusion

Experiments executed on data mining algorithms for text classification show that Multinomial Naïve Bayes algorithms are good in build model time and test model time. In addition to that, updateable algorithms have one more advantage by allow update on data model instead of building it from scratch.

One-Class classification is difficult data mining problem. LibSVM is the best available algorithm in Weka library.

Lazy nearest neighbour search algorithms do not need time to build model, but some of them will take long time to find the nearest neighbours. IBk algorithm achieved best time to find nearest neighbours without need for any time to build the model.

Apriori is the most suitable algorithm in finding association rules. This algorithm completed build and test model within acceptable time, and also provided 100% of correctness.

Clustering problem in this framework is not complex, so there was no need for a complex algorithm to cluster instances. Simple K Means algorithm has been selected due to efficiency in time cost and ability to cluster instances.

Precision and recall for ranking system are acceptable as first version, and need more improvements. Average precision was 32%, and average recall was 51%.

Average score of system usability scale was 64.5%. This result is very close to average score. All required comments and needed enhancements have been collected.

Result of research to find importance of ranking and recommendation factors showed that all factors have similar importance.

Chapter Seven

Conclusions and Recommendations

7.1 Introduction

This chapter summarizes conclusions result from this thesis, and provides recommendations for further researches. Work in this thesis can be divided into two parts: consolidated ranking and recommendation framework, and proper data mining techniques to be used. Below sections discuss conclusions and recommendations for proposed framework.

7.2 Main Results

This thesis proposed a Consolidated Ranking and Recommendation Framework (CRRF) for learning objects based on usage data. The framework designed to be flexible to support all work in this domain that already done in previous researches, and also to allow future researchers to include their work within this framework to improve ranking and recommendation result.

CRRF has flexibility to use common data for ranking and recommendation, in addition to its ability to use any external data that can be added to the framework. The framework supports complex formulas to allow researchers to configure their ranking and recommendation criteria and embed it in the framework. For example, system formulas support basic mathematical operations, statistical operations, logical operations, and data mining techniques. The framework has been built using SOA to easily be integrated with other systems.

While the framework has been designed to be flexible in its support for ranking and recommendation formulas, first contribution in this framework has been proposed in this thesis. Ranking and recommendation based on usage data is the first contribution in the consolidated framework. Proposed framework gathers usage data from learning management systems and analyses it by GAS module, and then builds user profiles automatically using data mining techniques.

User profiles will have information about contexts, learning interests, search objectives, relations with other users, and other basic information. Any additional information can be added to user profile in a flexible way.

GAS has two modules: Moodle2Cam and AutoProfileBuilder. Gathering usage data from LMS is responsibility of Moodle2Cam module that gathers data and stores it in CAM format. Moodle2Cam gathers information from LMS logs, users' enrolments into courses, attachments, and interactions with search engine (widget). Three libraries (POI, PDFBox, and Tika) are used to extract text from attachments, and these libraries provide support for most file types used in learning management systems. AutoProfileBuilder analyses data that collected by Moodle2Cam, and builds user profiles automatically. AutoProfileBuilder uses different data mining techniques (text classification, one-class classification, nearest neighbour search, association rules, and clustering) to analyse data and build user profiles. Text classification generates context model, one-class classification generates learning interests and search objectives, nearest neighbour search generates learning objects selection model, association rules generates learning objects recommendation model, and clustering generates relations recommendation model. AutoProfileBuilder uses Weka library for data mining techniques. This library is powerful, and has many data mining algorithms available for use. All these models will be used by Ranking and Recommendation modules to do their functions.

Ranking and Recommendation modules has flexible design by support of configuration feature of criteria, formulas, and fields that used in ranking and recommendation processes. All data available in user profile, gathered data by Moodle2Cam, models generated by AutoProfileBuilder, and any additional data provided can be used by ranking and recommendation formulas. Proposed framework is flexible for easy to change in used data mining techniques by allowing filtering for data and easy addition for new data mining algorithms, and develop customized ranking and recommendation handlers for a model. Ranking and recommendation modules will use configured criteria to calculate rank and provide recommendation of learning objects. Modules have cache management function to improve system performance. Cache management supports ranking, recommendation, and data mining models.

Use of effective data mining techniques that can provide high correctness within reasonable time cost is one of main research areas in this thesis. Weka library has been used for data mining techniques because of its support for Java language which is platform independent, and for its support to various algorithms. Proposed framework uses many data mining techniques such as: text classification, one-class classification, nearest neighbour search, association rules, and clustering. Each one of these techniques has several algorithms support it.

Experiments on text classification for eleven algorithms show that Naïve Bayes algorithms can complete build model in short time in comparison with other algorithms. On the other hand, algorithms that support multinomial text classification succeeded to complete test model in short time, and achieve high correctness. Multinomial algorithms depend on number of words in each document during classification, and this helped these algorithms to have high correctness. Naïve Bayes Multinomial Updateable has been selected to be used in the framework because of its support for multinomial method that can achieve high correctness, and it's belonging to Naïve Bayes algorithms that have high performance. In addition to that, this algorithm supports Updateable interface that allows this algorithm to be updated after built instead of build it again from scratch. This feature allows fast rapid addition and build for algorithm.

One-class classification / anomaly detection is a difficult problem, and few algorithms support it. Most algorithms need information about other classes to work properly, but in our application, it is not possible to provide information about other classes that is not part of learning interests or search objectives of a user. Only two algorithms support one-class in Weka library. Both algorithms couldn't achieve acceptable correctness, but in general LibSVM was better and more appropriate.

Experiments on nearest neighbour search for eight algorithms show that artificial neural network algorithms cannot build or test model in reasonable time because of their need to build complex network in case of large data, but lazy algorithms do not need any time to build model, and also succeeded to calculate nearest neighbours in acceptable time. IBk algorithm has been selected in proposed framework because it was the fastest in test model in addition to that, it is lazy algorithm and doesn't need any time to build model. Artificial neural network are known to have good correctness and acceptable time cost in short data, but these algorithms failed to be executed in reasonable time for large data.

Association rules experiments executed on seven algorithms. Time needed to build and test association rules must be very short to allow system to give acceptable performance for users. After exclude some algorithms that failed to provide acceptable time for build and test models, and analyse correctness factor for these algorithms, Apriori algorithm has been selected to be used in the proposed framework.

Experiments on clustering for fourteen algorithms show that some algorithms failed to build model within long time such as one or two days. Some algorithms built model in more than 10 hours, but some of them succeeded to build and test model in less than one minute. Algorithm ability to cluster instances is very important, and some algorithms succeeded to distribute data into 177 clusters while others generated less than 75 clusters. Simple K Means algorithm has been selected to be used in proposed framework because of its ability to build and test model in less than one minute, also its ability to generate 177 clusters when requested to generate 200 clusters.

Evaluation for CRRF has been conducted by calculating precision and recall to measure its ability to provide relevant learning objects. The average precision for all users was 32%, while other study found that Google precision was 65%. Most users complained about quality of learning objects and limited number of results. The average recall for all users was 51%.

Search engine usability has been evaluated using SUS (System Usability Scale). The search engine has been developed by Eummena.org project, and during work in this thesis, ranking and recommendation functions has been added to it. It was important to evaluate its usability to make sure that added functions didn't complex the search engine. Total average score for all users was 64.5%, while the average of 500 studies is 68%. The score of search engine is very close to average score; also several comments have been collected to improve the search engine usability.

There are several factors that can control ranking result such as: courses enrolled in, interactions with courses, and previous search keywords. Recommendation factors on the other hand are: relations with users enrolled in same courses, relations with users using learning management system by same way, and relation with users using same search keywords. Evaluation of users for these factors shows that users gave similar importance for these factors. In proposed framework, correctness of data mining technique was the factor in select the importance. For example, in ranking module, text classification is more accurate than one-class classification, so more weight has been given to text classification.

7.3 Recommendations for Further Research

This thesis proposed a Consolidated Ranking and Recommendation Framework (CRRF) based on usage data, but it is the first step on this direction. Further researches are needed:

- One-class classification efficiency is not high. There is a need to do research on more algorithms to achieve high efficiency. Such algorithm will improve ranking result according to learning interests and search objectives.
- More optimizations for ranking algorithm are needed to allow system to have better precision and recall scores. Optimization can be: add filters to usage data before analyse it, aggregate or summation of usage data, or collect usage data from additional sources.
- Ranking and recommendation modules designed to be flexible, but it still doesn't support ontology, and also doesn't support RDF (Resource Description Framework) and UNL (Universal Networking Language) formats. More work is needed to provide this support for the framework.
- Some data mining techniques not used in this framework such as decision trees, predictions, and estimations. More research is needed to find ability to include them in framework.
- Search engine needs some improvements to enhance its usability. Several comments have been collected from users. Applying these comments in search engine may enhance its usability.
- Framework may generate many recommended learning objects, so it will be useful if a rank can be given to recommended learning objects, and framework shows them with highest rank first.

- Evaluation of data mining techniques is based on duplicated data because it is not easy to have large real data. It is recommended to collect large real data, and perform evaluation for data mining again to have more accurate results.

References

Agrawal, R., & Srikant, R. (1994). Fast Algorithms for Mining Association Rules in Large Databases. *20th International Conference on Very Large Data Bases*, (pp. 478-499).

Aha, D., & Kibler, D. (1991). Instance-based learning algorithms. *Machine Learning , 6*, 37-66.

Al-Khalifa, H. (2008). Building an Arabic Learning Object Repository with an AdHoc Recommendation Engine. *Proceedings of iiWAS2008*, (pp. 390-394). Linz.

ALMazroui, Y. A. (2013). A survey of Data mining in the context of E-learning. *International Journal of Information Technology & Computer Science (IJITCS) , 7* (3), 8-18.

Ankerst, M., Breunig, M., Kriegel, H.-P., & Sander, J. (1999). OPTICS: Ordering Points To Identify the Clustering Structure. *ACM SIGMOD International Conference on Management of Data*, (pp. 49-60).

Apache UIMA. (2013, March 11). Retrieved March 11, 2013, from Apache UIMA project: http://uima.apache.org

APML-A. (2012, July 21). Retrieved July 21, 2012, from Attention Profile Mark-up Language: http://www.apml.areyoupayingattention.com/endusers/overview

APML-B. (2012, July 21). Retrieved July 21, 2012, from Attention Profile Mark-up Language: http://apml.areyoupayingattention.com

ARIADNE-A. (2012, September 19). Retrieved September 19, 2012, from ARIADNE Home Page: http://www.ARIADNE-eu.org/content/about

ARIADNE-B. (2012, September 19). Retrieved September 19, 2012, from ARIADNE Infrastructure: http://www.ARIADNE-eu.org/content/infrastructure

ARIADNE-C. (2012, September 19). Retrieved September 19, 2012, from ARIADNE Technologies: http://www.ARIADNE-eu.org/content/technologies

ARIADNE-D. (2012, September 19). Retrieved September 19, 2012, from ARIADNE Finder - Search Educational Resources: http://ARIADNE.cs.kuleuven.be/finder/ARIADNE

Arthur, D., & Vassilvitskii, S. (2007). k-means++: the advantages of carefull seeding. *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, (pp. 1027-1035).

Atkeson, C., Moore, A., & Schaal, S. (1996). *Locally weighted learning.* AI Review.

Barahate, S. R. (2012). Educational Data Mining as a Trend of Data Mining in Educational System. *International Conference & Workshop on Recent Trends in Technology, (TCET) 2012*, (pp. 11-17).

Barcelos, C. F., & Gluz, J. C. (2011). An Agent-based Federated Learning Object Search Service. *Interdisciplinary Journal of E-Learning and Learning Objects , 7*, 37-54.

Batista, V. L., Pintado, F. P., Gil, A. B., Rodriguez, S., & Moreno, M. (2011). A System for Multi-label Classification of Learning Objects. In C. E. al. (Ed.), *SOCO 2011, AISC 87*, (pp. 523-531).

Bayesian Network. (2013, March 12). Retrieved March 12, 2013, from Bayesian Network Classifiers in Weka for Version 3-5-7: http://www.cs.waikato.ac.nz/~remco/weka.bn.pdf

Bidgoli, H. (2003). *The internet encyclopedia* (First ed., Vol. 3). Wiley.

Burke, R. (2007). Hybrid Web Recommender Systems. In I. B. P. (Ed.), *The Adaptive Web* (pp. 377-408). Berlin: Springer.

Butoianu, V., Verbert, K., Duval, E., & Broisin, J. (2010). User Context and Personalized Learning: a Federation of Contextualized Attention Metadata. *Journal of Universal Computer Science , 16* (16), 2252-2271.

Cafolla, R. (2002). Project Merlot: Bringing Peer Review to Web-based Educational Resources. In C. C. al. (Ed.), *Proceedings of Society for Information Technology and Teacher Education International Conference* (pp. 614-618). Chesapeake: VA: AACE.

Calinski, T., & Harabasz, J. (1974). *A dendrite method for cluster analysis.*

CAREO. (2012, September 21). Retrieved September 21, 2012, from Internet and Education Guide - CAREO: http://theguide.ntic.org/display_lo.php?oai_id=oai%3Aeureka.ntic.org%3A4c99175698cbe7.29003580

Chang, C.-C., & Lin, C.-J. (2013, March 12). *LIBSVM*. Retrieved March 12, 2013, from A Library for Support Vector Machines: http://www.csie.ntu.edu.tw/~cjlin/libsvm

Chen, H.-R., & Huang, J.-G. (2012). Exploring Learner Attitudes toward Web-based Recommendation Learning Service System for Interdisciplinary Applications. *Educational Technology & Society , 15* (2), 89-100.

Cleary, J., & Trigg, L. (1995). K*: An Instance-based Learner Using an Entropic Distance Measure. *12th International Conference on Machine Learning*, (pp. 108-114).

Collis, B., & Strijker, A. (2004). Technology and human issues in reusing learning objects. *Journal of Interactive Media in Education , 4*, 1-32.

Connexions-A. (2012, September 20). Retrieved September 20, 2012, from Connexions Home Page: http://cnx.org

Connexions-B. (2012, September 20). Retrieved September 20, 2012, from Basic architecture: http://cnx.org/aboutus/overview

Connexions-C. (2012, September 20). Retrieved September 20, 2012, from Connexions Search for Content: http://cnx.org/content

Country Codes - ISO 3166. (2013, March 15). Retrieved March 15, 2013, from ISO: http://www.iso.org/iso/country_codes.htm

Dasgupta, S. (2002). Performance Guarantees for Hierarchical Clustering. *15th Annual Conference on Computational Learning Theory*, (pp. 351-363).

*Data Mining Concepts and Techniques*2006San FranciscoUSAMorgan Kaufmann Publishers. Elsevier Inc.

Decker, S., Melnik, S., Van Harmelen, F., Fensel, D., Klein, M. C., Broekstra, J., et al. (2000). The semantic web: The roles of XML and RDF. *IEEE Internet Computing , 15* (3), 63-74.

Downes, S. (2004). The Learning Marketplace: Meaning, Metadata and Content Syndication in the Learning Object Economy.

Dublin Core. (2003). Retrieved September 17, 2012, from Dublin Core Metadata Element Set: http://dublincore.org/documents/2003/02/04/dces/

Duval, E. (2004). We’re on the road to. In L. Cantoni, & C. McLoughlin (Ed.), *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications* (pp. 3-8). Lugano: AACE.

Duval, E., & Hodgins, W. (2003). *A LOM research agenda*. Retrieved September 21, 2012, from In Proceedings of the twelfth international conference on World Wide Web: http://www2003.org/cdrom/papers/alternate/P659/p659-duval.html.html

Eap, T., Hatala, M., & Richards, G. (2004). Digital Repository Interoperability: Design, Implementation and Deployment of the ECL Protocol and Connecting Middleware. *ACM , 1-58113-912-8/04/0005*, 376-377.

EdNA-A. (2012, September 20). Retrieved September 20, 2012, from About Education Network Australia (EdNA): http://apps-new.edna.edu.au/edna_retired/edna/go/about.html

EdNA-B. (n.d.). Retrieved September 20, 2012, from Education Network Australia (EdNA) and Metadata: http://apps-new.edna.edu.au/edna_retired/edna/go/resources/metadata.html

Edutella. (2012, September 21). Retrieved September 21, 2012, from Edutella Home Page: http://www.edutella.org/edutella/edutella.shtml

El Helou, S., Salzmann, C., & Gillet, D. (2010). The 3A Personalized, Contextual and Relation-based Recommender System. *Journal of Universal Computer Science , 16*, 2179-2195.

ELKI Background

EL-Manzalawy, Y. (2013, March 12). *Artificial Intelligence Research Laboratory*. Retrieved March 12, 2013, from Artificial Intelligence Research Laboratory: http://www.cs.iastate.edu/~yasser/wlsvm

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *Second International Conference on Knowledge Discovery and Data Mining*, (pp. 226-231).

Fisher, D. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine Learning , 2* (2), 139-172.

Flach, P., & Lachiche, N. (1999). Confirmation-Guided Discovery of first-order rules with Tertius. *Machine Learning , 42*, 61-95.

Fouad Ibrahim, K. (2012). Semantic Retrieval and Recommendation in Adaptive E-Learning System. *ICCIT*, (pp. 609-614). Bangladesh.

Frank, E., Hall, M., & Pfahringer, B. (2003). Locally Weighted Naive Bayes. *19th Conference in Uncertainty in Artificial Intelligence*, (pp. 249-256).

Gaudioso, E., & Talavera, L. (2004). *Data mining to support tutoring in virtual learning communities: experiences and challenges.* Catalunya: Llenguatges i Sistemes Informatics, Universitat Politecnica de Catalunya.

Gennari, J., Langley, P., & Fisher, D. (1990). Models of incremental concept formation. *Artificial Intelligence , 40*, 11-61.

Govaerts, S., El Helou, S., Erik, D., & Gillet, D. (2011). A Federated Search and Social Recommendation Widget. *Workshop SRS11, In Conjunction with CSCW 2011.* Hangzhou.

Han, J., Pei, J., & Yin, Y. (2000). Mining frequent patterns without candidate generation. *Proceedings of the 2000 ACM-SIGMID International Conference on Management of Data*, (pp. 1-12).

Hastie, T., & Tibshirani, R. (1998). Classification by Pairwise Coupling. *Advances in Neural Information Processing Systems* .

Henricksen, K., & Indulska, J. (2005). Developing context-aware pervasive computing applications: Models and approach. *Pervasive and Mobile Computing* .

Hochbaum, & Shmoys. (1985). A best possible heuristic for the k-center problem. *Mathematics of Operations Research , 10* (2), 180-184.

Hodgins, W. (2002). The Future of Learning Objects. *Proceedings of e-Technologies in Engineering Education: Learning Outcomes Providing Future Possibilities* , 76-82.

Holte, R. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine Learning , 11*, 63-91.

Hsu, I.-C. (2012). Intelligent Discovery for Learning Objects Using Semantic Web Technologies. *Educational Technology & Society , 15* (1), 298–312.

Huang, Y.-M., Chen, J.-N., & Cheng, S.-C. (2007). A Method of Cross-level Frequent Pattern Mining for Web-based Instruction. *Educational Technology & Society , 10* (3), 305-319.

Hung, J.-L., Rice, K., & Saba, A. (2012). An Educational Data Mining Model for Online Teaching and Learning. *Journal of Educational Technology Development and Exchange , 2*, 77-94.

(2005). *IEEE Draft Standard for Learning Technology - Extensible Markup Language (XML) Schema Definition Language Binding for Learning Object Metadata.* New York: Institute of Electrical and Electronics Engineers.

ISO 639 Language Codes. (2013, March 15). Retrieved March 15, 2013, from International Information Centre for Terminology: http://www.infoterm.info/standardization/iso_639_1_2002.php

(2011). *ISO/IEC Information technology - Learning, education and training - Metadata for learning resources - Part 1:Framework.* ISO/IEC 19788-1, Switzerland.

(2011). *ISO/IEC Information technology - Learning, education and training - Metadata for learning resources - Part 3:Basic application profile.* ISO/IEC 19788-3, Switzerland.

jHepWork Home

John, G., & Langley, P. (1995). Estimating Continuous Distributions in Bayesian Classifiers. *Eleventh Conference on Uncertainty in Artificial Intelligence*, (pp. 338-345). San Mateo.

Keerthi, S., Shevade, S., Bhattacharyya, C., & Murthy, K. (2001). Improvements to Platt's SMO Algorithm for SVM Classifier Design. *Neural Computation , 13* (3), 637-649.

Khribi, M. K., Jemni, M., & Nasraoui, O. (2009). Automatic Recommendations for E-Learning Personalization Based on Web Usage Mining Techniques and Information Retrieval. *Educational Technology & Society , 12* (4), 30-42.

KNIME Home Page. (2013, March 11). Retrieved March 11, 2013, from KNIME - Professional Open-Source Software: http://www.knime.org

Liu, B., Hsu, W., & Ma, Y. (1998). Integrating Classification and Association Rule Mining. *Fourth International Conference on Knowledge Discovery and Data Mining*, (pp. 80-86).

Liu, F.-J., & Shih, B.-J. (2010). Application of Data-Mining Technology on E-Learning Material Recommendation. (S. Soomro, Ed.) *E-learning Experiences and Future* , 213-228.

Maricopa. (2012, September 21). Retrieved September 21, 2012, from The Maricopa Learning eXchange (MLX) Home Page: http://www.mcli.dist.maricopa.edu/mlx

Mattson, M., Norman, D., & Purdy, R. (2002). *CAREO Campus Alberta Repository of Educational Objects.* Learning Commons, University of Calgary.

Mccallum, A., & Nigam, K. (1998). A Comparison of Event Models for Naive Bayes Text Classification. *AAAI-98 Workshop on 'Learning for Text Categorization'.*

MERLOT. (2012, September 20). Retrieved September 20, 2012, from MERLOT Home Page: http://www.merlot.org/merlot/index.htm

MineSet. (2013, March 12). Retrieved March 12, 2013, from Purple Insight, MineSet: http://www.algorithmic-solutions.com/leda/projects/mineset.htm

Moodle. (2012, July 12). Retrieved July 12, 2012, from Moodle: http://moodle.org

Multilayer perceptron. (2013, March 12). Retrieved March 12, 2013, from Wikipedia: http://en.wikipedia.org/wiki/Multilayer_perceptron

Multilayer perceptron. (2013, March 12). Retrieved March 12, 2013, from Wikipedia: http://en.wikipedia.org/wiki/Multilayer_perceptron

Najjar, J. (2008). *Learning Object Metadata: An Empirical Investigation and Lessons Learned.* KATHOLIEKE UNIVERSITEIT, LEUVEN, Belgium.

Najjar, J., Klerkx, J., Vuorikari, R., & Duval, E. (2005). Finding appropriate learning objects: An empirical evaluation. *9th European Conference on Research and Advanced Technology for Digital Libraries. ECDL. 3652*, pp. 323-335. Verlag: Springer.

Najjar, J., Ternier, S., & Duval, E. (2004). User Behavior in Learning Objects Repositories: An Empirical Analysis. *ED-MEDIA 2004 World Conference on Educational Multimedia, Hypermedia and Telecommunications*, (pp. 4373-4378).

Najjar, J., Wolpers, M., & Duval, E. (2006). Attention Metadata: Collection and Management. *WWW2006 workshop on Logging Traces of Web Activity: The Mechanics of Data Collection.* Edinburgh.

Nešić, S., Gašević, D., Jazayeri, M., & Landoni, M. (2011). A Learning Content Authoring Approach based on Semantic Technologies and Social Networking: an Empirical Study. *Educational Technology & Society , 14* (4), 35-48.

Niemann, K., Schmitz, H.-C., Kirschenmann, U., Wolpers, M., Schmidt, A., & Krones, T. (2012). Clustering by Usage: Higher Order Co-occurrences of Learning Objects. *LAK12: 2nd International Conference on Learning Analytics & Knowledge.* Vancouver.

Ochoa, X. (2008). *Learnometrics: Metrics for Learning Objects.* LEUVEN, Belgium: KATHOLIEKE UNIVERSITEIT.

OER Commons. (2007, February 01). Retrieved April 28, 2013, from OER Commons: http://www.oercommons.org/

Oracle Data Mining. (2013, March 11). Retrieved March 11, 2013, from Oracle Data Mining: http://www.oracle.com/technetwork/database/options/advanced-analytics/odm/index.html

Orange Data Mining. (2013, March 11). Retrieved March 11, 2013, from Orange - Data Mining Fruitful & Fun: http://orange.biolab.si

Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). *The PageRank Citation Ranking: Bringing Order to the Web.* Stanford InfoLab.

Platt, J. (1998). Fast Training of Support Vector Machines using Sequential Minimal Optimization. In B. Schoelkopf, C. Burges, & A. Smola (Ed.), *Advances in Kernel Methods - Support Vector Learning.*

Precision and recall. (2013, March 18). Retrieved March 22, 2013, from Wikipedia: http://en.wikipedia.org/wiki/Precision_and_recall

Precision and Recall Calculation. (2013, March 08). Retrieved March 08, 2013, from Data Mining Community's Top Resource: http://www.kdnuggets.com/faq/precision-recall.html

Radial basis function network. (2013, February 27). Retrieved March 12, 2013, from Wikipedia: http://en.wikipedia.org/wiki/Radial_basis_function_network

RapidMiner Overview

RDF. (2012, September 22). Retrieved September 22, 2012, from Semantic Web Standards - Resource Description Framework (RDF) Home Page: http://www.w3.org/RDF

Rehak, D., & Mason, R. (2003). Keeping the learning in learning objects. In L. A. (Ed.), *Reusing Online Resources: A Sustainable Approach to E-Learning* (pp. 22-30). London: Kogan Page.

Rennie, J., Shih, L., Teevan, J., & Karger, D. (2003). Tackling the Poor Assumptions of Naive Bayes Text Classifiers. *ICML* , 616-623.

Sampson, D., & Papanikou, C. (2009). A Framework for Learning Objects Reusability within Learning Activities. *ICALT 2009* (pp. 32-36). IEEE.

Sauro, J. (2011, February 02). *System Usability Scale (SUS)*. Retrieved December 14, 2012, from Measuring Usability: http://www.measuringusability.com/sus.php

Scale range of numbers. (2011, March 14). Retrieved February 19, 2013, from stackoverflow: http://stackoverflow.com/questions/5294955/how-to-scale-down-a-range-of-numbers-with-a-known-min-and-max-value

Scheffer, T. (2001). Finding Association Rules That Trade Support Optimally against Confidence. *5th European Conference on Principles of Data Mining and Knowledge Discovery*, (pp. 424-435).

Scheuer, O., & McLaren, B. (2011). *Educational Data Mining*. Encyclopedia of the Sciences of Learning, Springer.

Sharma, K., Jain, P., & Katare, R. (2011). E-Learning by Time Dynamic Model Using Data Mining. *Global Journal of Computer Science and Technology , 11* (17).

Sokvitne, L. (2000). An evaluation of the effectiveness of current dublin core metadata for retrieval. *VALA (Libraries, Technology and the Future), Biennial Conference* (p. 15). Victoria: Victorian Association for Library Automation Inc.

Sosteric, M., & Hesemeier, S. (2004). A first step towards a theory of learning objects. In M. R. (Ed.). London: Routledge Falmer.

SPSS Modeler. (2013, March 11). Retrieved March 11, 2013, from SPSS Modeler: http://www-01.ibm.com/software/analytics/spss/products/modeler

SQL Server - Analysis Services. (2013, March 11). Retrieved March 11, 2013, from Microsoft SQL Server - Analysis Services: http://www.microsoft.com/en-us/sqlserver/solutions-technologies/business-intelligence/analysis.aspx

Sree Dharinya, V., & Jayanthi, M. K. (2012). An Approach Towards Redefining Granularity of Learning Objects for Effective and Adaptive Personalization. *Journal of Theoretical and Applied Information Technology , 41* (1), 98-102.

Sridharan, B., Deng, H., & Corbitt, B. (2011). An Ontology-Based Learning Resources Management Framework for Exploratory E-learning. *Asia Pacific Management Review , 16* (2), 119-132.

Srikant, R., & Agrawal, R. (1996). Mining Sequential Patterns. *Generalizations and Performance Improvements* .

Su, J., Zhang, H., Ling, C., & Matwin, S. (2008). Discriminative Parameter Learning for Bayesian Networks. *ICML* .

Ternier, S. (2008). *Standards based Interoperability for Searching in and Publishing to Learning Object Repositories.* LEUVEN, Belgium: KATHOLIEKE UNIVERSITEIT.

Ternier, S., Duval, E., & Neven, F. (2003). Using a P2P architecture to provide interoperability between Learning Object Repositories. In D. Lassner, & C. McNaught (Ed.), *World Conference on Educational Multimedia, Hypermedia and Telecommunications* (pp. 148-151). Honolulu: AACE.

The WEKA Data Mining Software: An Update2009*SIGKDD Explorations 11*110-18

Ueno, M. (2004). Data mining and text mining technologies for collaborative learning in an ILMS "Samurai". *IEEE International Conference on Advanced Learning Technologies (ICALT'04).*

Verbert, K. (2008). *An Architecture and Framework for Flexible Reuse of Learning Object Components.* LEUVEN, Belgium: KATHOLIEKE UNIVERSITEIT.

Verbert, K., Drachsler, H., Manouselis, N., Wolpers, M., Vuorikari, R., & Duval, E. (2011). Dataset-driven Research for Improving Recommender Systems for Learning. 1st International Conference Learning Analytics & Knowledge. *1st International Conference Learning Analytics & Knowledge.* Banff.

Wang, J., Zucker, & Daniel, J. (2000). Solving Multiple-Instance Problem: A Lazy Learning Approach. *17th International Conference on Machine Learning*, (pp. 1119-1125).

Weka 3: Data Mining Software in Java. (2013, March 12). Retrieved March 12, 2013, from Weka 3: Data Mining Software in Java: http://www.cs.waikato.ac.nz/ml/weka/

Wikipedia E-learning. (2012, September 15). Retrieved September 15, 2012, from Wikipedia: http://en.wikipedia.org/wiki/E-Learning

Wiley, D. (2002). Connecting Learning Objects to Instructional Design Theory: A Definition, a Metaphor, and a Taxonomy. *The Instructional Use of Learning Objects.* Bloomington.

Wolpers, M., Najjar, J., Verbert, K., & Duval, E. (2007). Tracking Actual Usage: the Attention Metadata Approach. *Educational Technology & Society , 10* (3), 106-121.

Yang, Y., Guan, X., & You, J. (2002). CLOPE: a fast and effective clustering algorithm for transactional data. *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, (pp. 682-687).

Zapata, A., Menendez, V., Prieto, M., & Romero, C. (2011). A Hybrid Recommender Method for Learning Objects. *Design and Evaluation of Digital Content for Education (DEDCE) 2011.* International Journal of Computer Applications (IJCA).

Zheng, Z., & Webb, G. (2000). Lazy Learning of Bayesian Rules. *Machine Learning , 4* (1), 53-84.

Appendices

Appendix 1: Learning Objects Metadata

Category groups for metadata fields in IEEE LOM:

No.	Category	Description
1	General	Describes general characteristics of the learning object (title, language, description, keyword, etc)
2	Life cycle	Elements affected learning object life cycle and its current status (version, status, contribute, etc)
3	Meta-Metadata	Data about the metadata entry (contribute, metadata schema, language, etc)
4	Technical	Technical characteristics and needs of the learning object (format, size, location, requirement, installation remarks, etc)
5	Educational	Pedagogical characteristics of the learning object (interactivity type, learning resource type, interactivity level, semantic density, context, difficulty, etc)
6	Rights	Copyright and conditions of use (cost, copyright and restrictions, etc)
7	Relation	Relations of the learning object with other learning objects (relation, kind, resource, etc)
8	Annotation	Comments and reviews on learning object (entity, date, description, etc)
9	Classification	Taxonomic path in classification system (purpose, taxon path, keyword, etc)

Elements for metadata in DCMES standard:

No.	Item	Definition
1	Title	A name given to the resource
2	Creator	An entity primarily responsible for making the content of the resource
3	Subject	A topic of the content of the resource
4	Description	An account of the content of the resource
5	Publisher	An entity responsible for making the resource available
6	Contributor	An entity responsible for making contributions to the content of the resource
7	Date	A date of an event in the lifecycle of the resource
8	Type	The nature or genre of the content of the resource
9	Format	The physical or digital manifestation of the resource
10	Identifier	An unambiguous reference to the resource within a given context

11	Source	A Reference to a resource from which the present resource is derived
12	Language	A language of the intellectual content of the resource
13	Relation	A reference to a related resource
14	Coverage	The extent or scope of the content of the resource
15	Rights	Information about rights held in and over the resource

"Data element specification" example (ISO/IEC Framework, 2011, p.12):

No.	Data element attribute	Value
1	Identifier (mandatory)	ISO_IEC_19788-3:2010::DES0300
2	Property name (mandatory)	format (eng)
3	Definition (mandatory)	file format of the learning resource (eng)
4	Linguistic indicator (mandatory)	non-linguistic
5	Domain (mandatory)	Learning Resource (ISO_IEC_19788-1:2010::RC0002)
6	Range (mandatory)	Literal
7	Content value rules (conditional)	RS_DES0300
8	Refines (conditional)	ISO_IEC_19788-2:2010::DES0900
9	Example(s) (optional)	video/mpeg text/html
10	Note(s) (optional)	-

Appendix 2: Mapping from Moodle to CAM

Mapping from Moodle (Log) to CAM:

CAM		**Moodle**	
Table	**Column**	**Table**	**Column**
event	datetime	mdl_log	time
event	name	mdl_log	action
event	sharinglevel	always "public"	
eventrelatedentity	role	mdl_log	module
relatedentity	id	mdl_log	id
relatedentity	entityid	mdl_log	userid
relatedentity	metadataid	mdl_log	course
relatedentity	metadatareference	mdl_log	info
relatedentity	mimetype	always "null"	
relatedentity	name	mdl_log	url
session	domain	institution id where moodle installed	

CAM		Moodle	
Table	**Column**	**Table**	**Column**
session	ipaddress	mdl_log	ip
session	sessionid	always "null"	
reference content		additional information about log. such as: course title, section name, forum subject, assignment subject, etc.	

Mapping from Moodle (User's enrolments) to CAM:

CAM		Moodle	
Table	**Column**	**Table**	**Column**
event	datetime	mdl_user_enrolments	timecreated
event	name	always "enrol"	
event	sharinglevel	always "public"	
eventrelatedentity	role	always "user"	
relatedentity	id	mdl_user_enrolments	id
relatedentity	entityid	mdl_user_enrolments	userid
relatedentity	metadataid	mdl_enrol	courseid
relatedentity	metadatareference	always "null"	
relatedentity	mimetype	always "null"	
relatedentity	name	always "null"	
session	domain	institution id where moodle installed	
session	ipaddress	always "null"	
session	sessionid	always "null"	

Mapping from Moodle (Attachments and Files) to CAM:

CAM		Moodle	
Table	**Column**	**Table**	**Column**
event	datetime	mdl_files	timecreated
event	name	depends on mdl_context.contextlevel: 30 = user 50 = course 70 = module	
event	sharinglevel	always "public"	
eventrelatedentity	role	always "file"	
relatedentity	id	mdl_files	id
relatedentity	entityid	mdl_files	userid
relatedentity	metadataid	mdl_course_modules	course
relatedentity	metadatareference	mdl_files	filename
relatedentity	mimetype	mdl_files	mimetype

CAM		Moodle	
Table	**Column**	**Table**	**Column**
relatedentity	name	mdl_files	contenthash
session	domain	institution id where moodle installed	
session	ipaddress	always “null”	
session	sessionid	always “null”	
reference content		parse text from physical file	

Mapping from Moodle (Interactions with Search Engine) to CAM:

CAM		Moodle	
Table	**Column**	**Table**	**Column**
event	datetime	date of interaction	
event	name	“search”, “rate”, or “select”	
event	sharinglevel	always “public”	
eventrelatedentity	role	always “widget”	
relatedentity	id	event	id
relatedentity	entityid	logged in userid from moodle	
relatedentity	metadataid	rate and select: learning object id in external repository search: always “id0” , means null	
relatedentity	metadatareference	always “lomv1.0” for search. information about learning object (title, body, location, and keywords) for rate and select.	
relatedentity	mimetype	always “text/html”	
relatedentity	name	rate: rated score. from 1 to 5 (bad to gorgeous) search: keywords used in widget search select: title, keywords, and description of learning object	
session	domain	institution id where moodle installed	
session	ipaddress	ip address for logged in user	
session	sessionid	session id for logged in user	

Appendix 3: Ranking and Recommendation Criteria

crrf_rank_criteria:

criteria_id	criteria_name	criteria_expression
1	Text classification ranking using context, and one-class classification using learning interests and search objectives	F1+F2+(F3*0.025)+(F4*0.025)

crrf_recommend_criteria:

criteria_id	criteria_name	criteria_expression
1	Recommendation using learning object and session	F5
2	Recommendation by users clusters	F6

crrf_formula:

formula_id	formula_name	model_id
1	Context formula by learning object title	1
2	Context formula by learning object body	1
3	Learning interests formula by learning object title	3
4	Search objectives formula by learning object title	4
5	Learning object session associator	8
6	Users clusterers associator	9

crrf_model:

model_id	model_name	model_level
1	ContextModel	INSTITUTION
2	LearningObjectModel	INSTITUTION
3	LearningInterestModel	USER
4	SearchObjectiveModel	USER
5	CourseClustererModel	PUBLIC
6	SearchClustererModel	PUBLIC
7	LearningClustererModel	PUBLIC
8	LearningObjectSessionAssociatorModel	PUBLIC
9	UserClustererModel	PUBLIC

crrf_formula_field:

formula_field_id	formula_id	field_id	field_order	field_condition
3	1	1	1	(NULL)
4	2	2	1	(NULL)
5	3	1	1	(NULL)
6	4	1	1	(NULL)
7	5	4	1	exists (select 1 from crrf_learning_object_session los2 where los2.metadataId = ':INPUT_LO_ID' and los1.sessionId = los2.sessionId) and los1.metadataId <> ':INPUT_LO_ID'
8	6	5	1	e.id = ere.eventid and ere.relatedentityid = re.id and ere.role='widget' and (e.name='select' or (e.name = 'rate' and re.name in ('3','4','5'))) and exists (select cup2.user_id from crrf_user_profile cup1, crrf_profile_relation cpr1, crrf_profile_relation cpr2, crrf_user_profile cup2 where cup1.user_id = ':INPUT_USER_ID' and cup1.user_profile_id = cpr1.user_profile_id and cpr1.group_id = cpr2.group_id and cpr1.relation_type_id = cpr2.relation_type_id and cpr1.user_profile_id <> cpr2.user_profile_id and cpr2.user_profile_id = cup2.user_profile_id and re.entityId = cup2.user_id)

crrf_field:

field_id	field_name	field_table	field_column
1	Learning object title	:INPUT_LO	:INPUT_LO_TITLE
2	Learning object body	:INPUT_LO	:INPUT_LO_BODY
3	Learning object	:INPUT_LO	:INPUT_LO_KEYWORDS

	keywords		
4	learning object session associator	crrf_learning_object_session los1	los1.sessionId, los1.metadataId
5	users clusterers associator	event e, eventrelatedentity ere, relatedentity re	re.entityId, re.metadataId

Appendix 4: Class Diagram for Ranking and Recommendation Modules

Attributes of GeneralTechnique class:

Attribute Name	Attribute Description
trainingData	data to build train data mining technique.
upToDate	identify if data mining technique has been built or not. System will check this attribute when use the technique, and build it if not build yet. Usually system builds all data mining techniques after add training data to them, and before start use them.
attributes	data attributes in trainingData. There are different attributes for each technique. List of attributes for each technique explained below.
filter1 and filter2	filters to be used on trainingData before build and use the technique. Some filters work with data types different from types provided in trainingData, so there is a need to filter and transfer data to different form. For example, NaiveBayesMultinomialUpdateable algorithm works with string vector, and cannot work with text directly. So, user needs to define filter1 to change data from string to string vector. User can define two filters to be applied on trainingData. Generated data from these filters will be defined in filteredData attribute.
filteredData	system will filter trainingData using filter1 and filter2 if exist, then save result in filteredData attribute to be used to build technique.

Methods of GeneralTechnique class:

Method Name	Method Description
addAttribute	add new attribute for the technique to make it easy to configure new technique in sub-classes.
getAttribute	get attribute details from technique.
getAttributeValues	get list of values for nominal attributes.
setAttribute	replace existing attribute by new one.
updateAttributeValues	update list of values for nominal attributes.
makeInstance	create new instance to be added to trainingData or to be classified/clustered.
addData	add new instance to trainingData.
setClassIndex	set index of class attribute. Usually index is last attribute.
setupAfterCategories Added	create attributes in technique after add all of them.
filterInstance	filter instance to be classified/clustered.
buildIfNeeded	build technique before use it (if not already built).
getClassValueIndex	get index of a value in nominal attribute values.

Attributes of data mining techniques:

Technique Name	Attribute Name	Attribute Type	Attribute Description
LearningObjectSessionAssociator	_crrf_session_id_attr_	NOMINAL	HTTP session
	_crrf_learning_object_attr_	NOMINAL	Selected learning object ID
TextClassifier	_crrf_text_attr_	STRING	Course description and content
	_crrf_class_attr_	NOMINAL	Course ID
LearningObjectClassifier	_crrf_context_attr_	NOMINAL	Context class ID
	_crrf_language_attr_	NOMINAL	User's language ID
	_crrf_country_attr_	NOMINAL	User's country ID
	_crrf_ip_address_attr_	NUMERIC	IP address in decimal
	_crrf_learing_object_attr_	NOMINAL	Learning object ID
OneClassTextClassifier	_crrf_text_attr_	STRING	Text of learning interest or search objective
	_crrf_class_attr_	NOMINAL	One-class only. User ID
CourseClusterer	_crrf_institution_attr_	NOMINAL	Institution ID

Technique Name	Attribute Name	Attribute Type	Attribute Description
	_crrf_course_attr_	NOMINAL	Course ID
	_crrf_user_attr_	NOMINAL	User ID
LearningClusterer	_crrf_event_name_attr_	NOMINAL	Event name in Moodle, such as view, add, delete, etc.
	_crrf_role_attr_	NOMINAL	Role name in Moodle such as course, forum, discussion, file, etc.
	_crrf_ip_address_attr_	NUMERIC	IP address in decimal
	_crrf_language_attr_	NOMINAL	User's language
	_crrf_country_attr_	NOMINAL	User's country
	_crrf_user_attr_	NOMINAL	User ID
SearchClusterer	_crrf_title_attr_	NOMINAL	Search keyword
	_crrf_learning_object_attr_	NOMINAL	Learning object ID
	_crrf_user_attr_	NOMINAL	User ID

Appendix 5: Software and Hardware Specifications

Software and hardware specifications used to execute data mining experiments:

Item	Description
Data Mining Library	Weka 3.7.7
Java Runtime Environment	Java SE 1.6.0 (build 1.6.0_11-b03)
Java Arguments	Java heap size 1500 MB (-Xmx1500m)
Operating System	Windows 7 Enterprise
Database	MySQL Server 5.1.30
Processor	Intel (R) Core (TM) 2 Duo CPU T9400 @ 2.53 GHz
Installed Memory	2.00 GB
System Type	32 bits Operating System

Appendix 6: CRRF Evaluation Guide

1. Introduction

Share and reuse of learning materials is one of the main goals of educational repositories. Finding appropriate reusable learning materials is still one of the highest challenges facing users of educational repositories. Users are not able to find many high quality learning materials relevant to them. Several tools were developed to improve searching learning objects, but most of these tools are content based oriented, and depends on metadata of resource content provided by indexers. In this thesis, I proposed Consolidated Ranking and Recommendation Framework (CRRF) that uses usage data to improve finding of relevant learning objects. Proposed work includes analysis of usage data to create dynamic user profiles automatically to improve Ranking and Recommendation of learning materials.

Ranking is the process of sort learning materials for user according to his contexts, learning interests, and search objectives. Ranking works when user performs search query on learning materials.

Recommendation is the process of suggest learning materials for user according to most frequent used materials, and according to learning materials recommended by his peer users.

2. Evaluation

The purpose of evaluation is to find the correct things in my approach, and also find wrong methods to fix them in future work. Evaluation will be applied for both ranking and recommendation methods.

Important: Please note that system will log all your transactions including used keywords, search results, ranked learning materials, manual evaluation, etc. I will analyse this information to evaluation the framework.

2.1. Login:

Follow below link, and use username and password provided to you to login. Please don't give any person this information because it is for your use only.

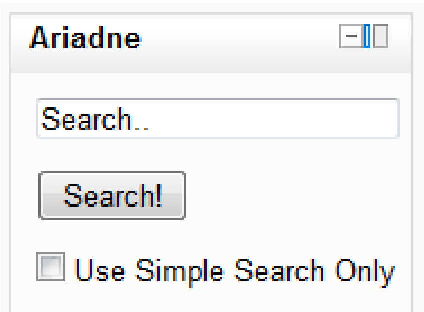
http://72.167.55.66

2.2. Your Courses:

You will be enrolled in some courses automatically, so please keep these courses in mind during perform your search. System will rank learning materials according to these courses. The learning materials which are relative to these courses, will get higher mark, and appear first in search result.

2.3. Rank Learning Materials:

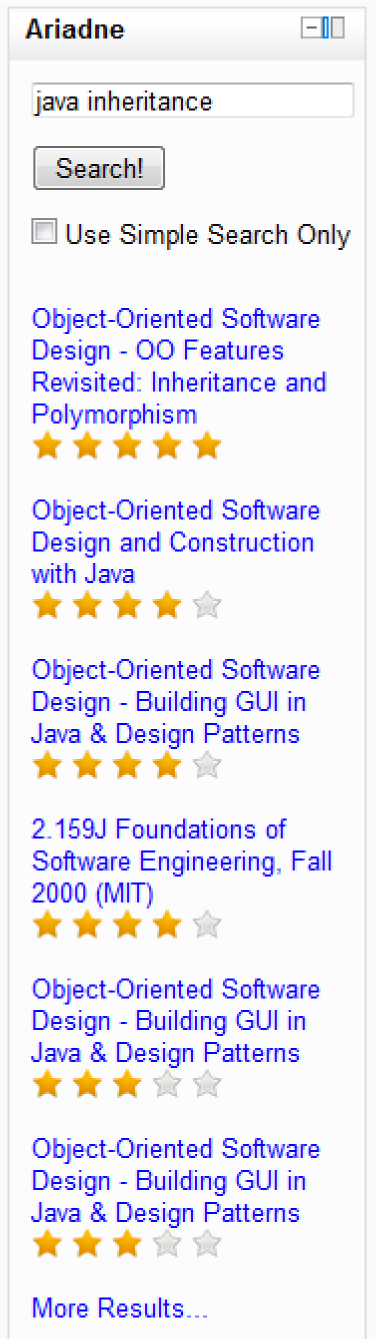
After login, on the left down of your screen, you can find Ariadne search widget to let you search on learning materials. This widget has been developed by eummena.org project, and I added to it intelligent Ranking and Recommendation methods.



Click on search text field and enter your search keyword, then click Search button.

System will get result, Rank it according to your courses, learning interests (how you use E-Class), and search objectives (previous keywords used in search).

Search result will look like below figure. System will show first 6 results, and you can click on (More Results).



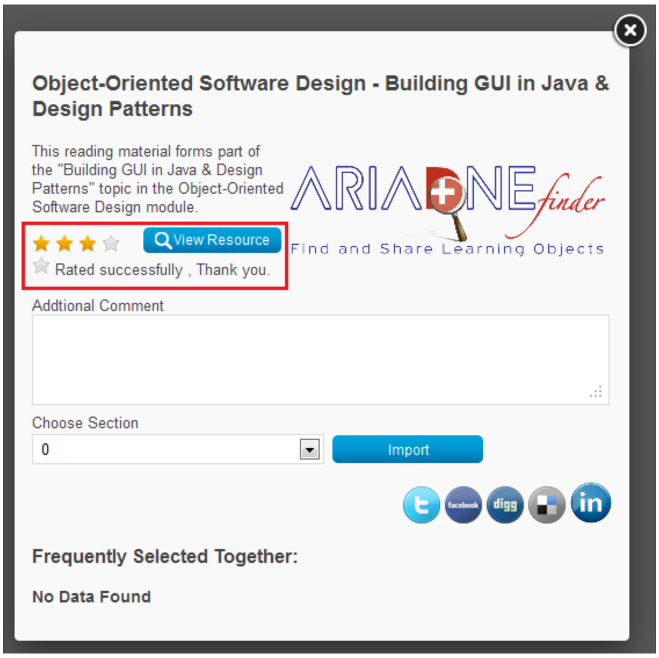
You can check box (Use Simple Search Only) to cancel the Ranking process, and retrieve search result without any process from my system. This can help you to compare the effect of my system on search result.

You can click on any learning material to show more details.

2.4. Evaluate Ranking Result:

System will log all your transactions, but you need to tell me your opinion about ranked learning materials. You can do this by evaluate the rank for each learning material manually.

You can evaluate ranking for learning material manually by click on one of five stars as shown in below figure. Remember that system ranked learning materials according to courses you are enrolled in them. Also you can click on (View Resource), and system will understand that you liked the learning material and want to download it. Some download resource links are broken due to inconsistency in repository that I use, and this bug has nothing to do with my work.



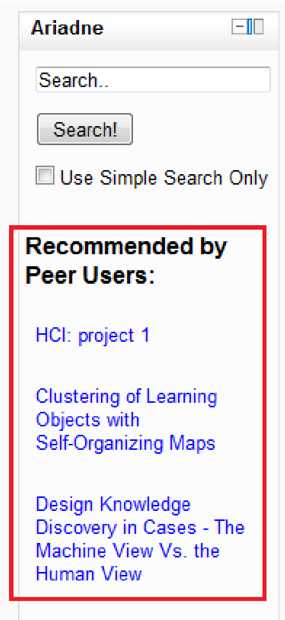
2.5. Recommendation of Learning Materials:

There are two types of Recommendation will be provided by system:

2.5.1. Recommendation by Peer Users:

System will find relations between you and other users such as courses, learning interests, and search objectives, then find most learning materials selected and recommended by other users.

This Recommendation will work automatically without do any search.



2.5.2. Frequently Selected Together:

When you open any learning material details, system will find other learning materials that usually have been selected together. This function similar to (Frequently Bought Together) provided by Amazon. System will consider selection for any learning material when you click on (View Resource).

2.6. Evaluation Period:

System gathers usage data and analyses it every one hour. So, your interaction with the system will be applied on Ranking and Recommendation after one hour (maximum). Also my Ranking and Recommendation methods depend on actions by other users.

So, please use the system for 5 days. You can use it for short time only every day.

2.7. Evaluation Methods:

I will analyse system logs to evaluate the framework. Also, after you complete your usage, please login to below link, and fill the survey. It is very short, and will take less than 5 minutes.

Use below link to login to Survey:

https://docs.google.com/forms/d/1sbEZGL5L8UcDKt4UlVJG0HmC66L437Pz1KIxX3Iixdg/viewform

Appendix 7: CRRF Evaluation Survey

Consolidated Ranking and Recommendation Framework

This survey is to evaluate Consolidated Ranking and Recommendation Framework for Learning Objects.

The survey has 12 questions only, it is expected to be completed in less than 5 minutes.

The research is being conducted by master student Baha' Harasheh in Al-Quds University, with cooperation from Eummena, and supervision of Dr. Jad Najjar and Dr. Rashid Jayousi.

Continue »

Powered by Google Drive

This content is neither created nor endorsed by Google.

Report Abuse - Terms of Service - Additional Terms

Consolidated Ranking and Recommendation Framework

* Required

Survey Authentication

Username *

Please insert username sent to you by email (used to login to Consolidated Ranking and Recommendation Framework).

Survey Code *

Please insert survey code sent to you by email.

« Back Continue »

Powered by Google Drive

This content is neither created nor endorsed by Google.

Report Abuse - Terms of Service - Additional Terms

Consolidated Ranking and Recommendation Framework

* Required

System Usability Scale

1. I think that I would like to use this system frequently. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

2. I found the system unnecessarily complex. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

3. I thought the system was easy to use. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

4. I think that I would need the support of a technical person to be able to use this system. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

5. I found the various functions in this system were well integrated. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

6. I thought there was too much inconsistency in this system. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

7. I would imagine that most people would learn to use this system very quickly. *

	1	2	3	4	5

8. I found the system very cumbersome to use. *

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

9. I felt very confident using the system. *

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

10. I needed to learn a lot of things before I could get going with this system. *

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

« Back Continue »

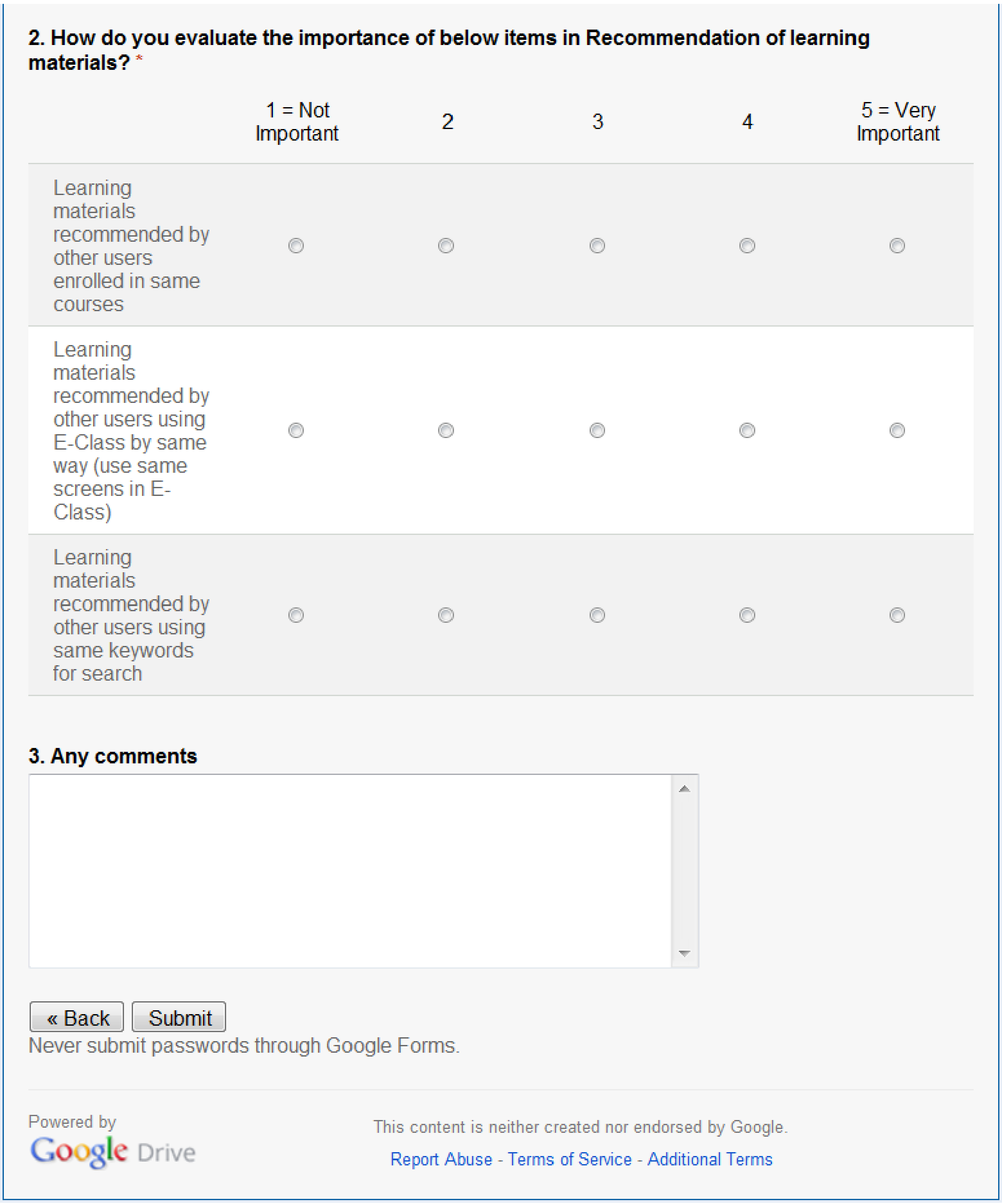
Consolidated Ranking and Recommendation Framework

* Required

Ranking and Recommendation

1. How do you evaluate the importance of below items in Ranking of learning materials? *

	1 = Not Important	2	3	4	5 = Very Important
Courses you are enrolled in	◯	◯	◯	◯	◯
Your interactions with course materials (lectures, forums, assignments, etc)	◯	◯	◯	◯	◯
Your previous search keywords	◯	◯	◯	◯	◯


2. How do you evaluate the importance of below items in Recommendation of learning materials? *
1 = Not Important
2
3
4
5 = Very Important
Learning materials recommended by other users enrolled in same courses
Learning materials recommended by other users using E-Class by same way (use same screens in E-Class)
Learning materials recommended by other users using same keywords for search
3. Any comments
« Back
Submit
Never submit passwords through Google Forms.
Powered by
Google Drive
This content is neither created nor endorsed by Google.
Report Abuse - Terms of Service - Additional Terms