

**CLASSIFICATION OF LEARNING OBJECT'S WEB PAGES
UNDER EDUCATIONAL LEVELS**

By

Manuel J. Orán-Hernández

A thesis submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER ENGINEERING

**UNIVERSITY OF PUERTO RICO
MAYAGÜEZ CAMPUS**

December, 2014

Approved by:

Aidsa I. Santiago Román, Ph.D
Member, Graduate Committee

Date

José Fernando Vega Riveros, Ph.D
Member, Graduate Committee

Date

Amirhossein Chinaei, Ph.D
Member, Graduate Committee

Date

Nayda G. Santiago-Santiago, Ph.D
President, Graduate Committee

Date

Sonia M. Bartolomei Suárez, Ph.D
Representative of Graduate Studies

Date

Pedro I. Rivera Vega, Ph.D
Chairperson of the Department

Date

ABSTRACT

The internet has become the largest source of educational resources over the last decade. However, most of the educational resources are still unorganized and deficient in the application of educational models. To overcome this problem, different metadata formats, digital libraries and, web directories have been implemented. Currently, these applications require domain experts in Computing and Education to properly categorize the educational properties of an educational material manually. Recently, the *educational level* has raised as an important property of educational materials according to the 21st century pedagogical needs and interests of academicians. Nevertheless, most of the online educational materials still lack of this description.

In this thesis we addressed the task of automatically determining the educational level property of an educational material based on its web page on-page features. By experimenting on a data corpora of pre-labeled web pages of educational materials under the K-12 *educational levels*, we demonstrated that the determination of the Main Categories (Elementary School, Middle School, and High School) of the *educational levels* property can be automated by a computerized system using supervised learning techniques.

RESUMEN

El internet se ha convertido en el recurso más abundante de información educativa en la última década. Aún, la mayoría de los recursos educacionales en línea están desorganizados y carecen de la aplicación de modelos educativos. Para contrarrestar este problema, diferentes formatos de “metadata”, librerías digitales y directorios en línea han sido implementados. Actualmente estas aplicaciones requieren de expertos en las áreas de Computación y Educación para manualmente categorizar correctamente las propiedades de un material educativo. Recientemente, el nivel educativo ha surgido como una propiedad importante de los recursos educativos con respecto a las necesidades pedagógicas e intereses de los académicos del siglo 21st y la mayoría de los recursos educativos carecen de esta descripción.

En esta tesis nos enfocamos en la tarea de determinar automáticamente la propiedad del nivel educacional para un recurso educativo basado en los “on-page features” de una página web. Experimentando con un cuerpo de datos de páginas web de materiales educativos previamente etiquetados con la propiedad del nivel educativo de K-12, demostramos que la determinación de las categorías principales (Escuela Elemental, Escuela Intermedia y Escuela Superior) del los niveles educativos puede ser automatizada por un sistema computarizado utilizando técnicas de aprendizaje supervisado.

Copyright © 2014

by

Manuel J. Orán-Hernández

To God, my family, and the scientific community ...

TABLE OF CONTENTS

	<u>page</u>
ABSTRACT ENGLISH	ii
ABSTRACT SPANISH	iii
LIST OF TABLES	ix
LIST OF FIGURES	x
LIST OF ABBREVIATIONS	xi
1 INTRODUCTION	1
1.1 Motivation and Background	4
1.2 Problem Statement	6
1.3 Scope of the Work	7
1.4 Significance of the Study	8
1.5 Overview of the Methodology	9
1.6 Contributions	10
1.7 Organization of the Thesis	11
2 LITERATURE REVIEW	12
2.1 Knowledge Discovery in Databases	12
2.1.1 Educational Data Mining	15
2.2 Web Mining	16
2.2.1 Web Mining Content in Education	17
2.2.2 The Hyperlinked Web and The Semantic Web	18
2.2.2.1 Ontologies	20
2.3 Related Work	21
3 THEORETICAL FRAMEWORK	26
3.1 Educational Materials: Learning Objects	26
3.1.1 Learning Objects Properties	27
3.1.1.1 Learning Objects Metadata formats	27
3.2 Learning Object's Web Page classification	28
3.2.1 Classification Type	28
3.2.2 Feature Selection	31
3.2.3 Classification Algorithms	31
3.2.3.1 Naïve Bayes	33
3.2.3.2 Support Vector Machine (SVM)	34

3.2.3.3	Maximun Entropy (MaxEnt)	36
4	METHODOLOGY	39
4.1	General Description	39
4.2	Tools	39
4.2.1	Java	39
4.2.2	jsoup	40
4.2.3	Weka	40
4.2.4	LibLINEAR	40
4.3	Design Approach	41
4.4	Experiments	43
4.4.1	Determining Class Labels	43
4.4.1.1	Categories of Educational Levels	43
4.4.2	Data Gathering	44
4.4.3	Data Preprocessing	45
4.4.3.1	Web Pages Content Feature Selection	45
4.4.4	Data Transformation	47
4.4.4.1	Feature Extraction	47
4.4.4.2	Feature Weighting	48
4.4.4.3	Feature Selection	49
4.4.4.4	Data Mining or Pattern Extraction	50
4.4.4.5	Results Analysis	50
4.5	Experimental Questions	51
5	Experimental Results	55
5.1	Evaluation Metrics	55
5.2	Experimental Questions Results	58
5.3	Results Discussion	63
6	CONCLUSION AND FUTURE WORK	66
6.1	Conclusions	66
6.2	Future Work	67
	APPENDICES	70
A	Experiments Results per Classifier	71
A.1	Experiment 1: Results per Classifier	71
A.1.1	Multinomial Naive Bayes Results	71
A.1.1.1	Multinomial Naive Bayes 10-Fold Cross Validation Results	71
A.1.1.2	Multinomial Naive Bayes Split Percentage Results	72
A.1.2	Support Vector Machine Results	72
A.1.2.1	Support Vector Machine 10-Fold Cross Validation Results	72

	A.1.2.2 Support Vector Machine Split Percentage Results . .	73
	A.1.3 MaxEnt Results	74
	A.1.3.1 MaxEnt 10-Fold Cross Validation Results	74
	A.1.3.2 MaxEnt Split Percentage Results	74
B	Example	76
	B.1 Worked example	81

LIST OF TABLES

<u>Table</u>	<u>page</u>
5-1 Results Comparison	59
5-2 Support Vector Machine 10-Fold C-V in DC_{MC}	60
5-3 Support Vector Machine 10-Fold C-V in DC_{MC} Confusion Matrix . . .	60
5-4 Support Vector Machine 10-Fold C-V in DC_{SC}	61
5-5 Support Vector Machine 10-Fold C-V in DC_{SC} Confusion Matrix . . .	62
A-1 Multinomial Naive Bayes 10-Fold Cross Validation	71
A-2 Multinomial Naive Bayes 10-Fold Cross Validation Confusion Matrix .	71
A-3 Multinomial Naive Bayes Split Percentage	72
A-4 Multinomial Naive Bayes Split Percentage Confusion Matrix	72
A-5 Support Vector Machine 10-Fold Cross Validation	73
A-6 Support Vector Machine 10-Fold Cross Validation Confusion Matrix .	73
A-7 Support Vector Machine Split Percentage	73
A-8 Support Vector Machine Split Percentage Confusion Matrix	74
A-9 MaxEnt 10-Fold Cross Validation	74
A-10MaxEnt 10-Fold Cross Validation Confusion Matrix	74
A-11MaxEnt Split Percentage	75
A-12MaxEnt Split Percentage Confusion Matrix	75
B-1 Worked Example Data Corpus	81

LIST OF FIGURES

<u>Figure</u>	<u>page</u>
2-1 The Knowledge Discovery in Databases (KDD) technique [1]	13
2-2 Educational Data Mining Cycle [2]	15
2-3 The Hyperlink vs Semantic Web organizations [3]	19
2-4 Semantic Web Architecture [4]	20
3-1 Muticlass, single-label classification [5]	30
4-1 System Architecture	42
4-2 Educational Levels	54
B-1 Example LO's Web Page part 1 [6]	77
B-2 Example LO's Web Page part 2 [6]	78
B-3 Example LO's Web Page part 3 [6]	79

LIST OF ABBREVIATIONS

ASN	Achievement Standard Network
FN	False Negatives
FP	False Positives
HTML	HyperText Markup Language
JVM	Java Virtual Machine
KDD	Knowledge Discovery in Databases
LAR	Learning Application Readiness
LO	Learning Object
LOM	Learning Object Metadata
LTSC	Learning Technology Standards Committee
MaxEnt	Maximum Entropy (Multinomial Logistic Regression)
MNNB	Multinomial Naive Bayes
NSDL	National Science Digital Library
RDF	Resource Description Framework
SVM	Support Vector Machine
TF-IDF	Term Frequency ? Inverse Document Frequency
TN	True Negative
TP	True Positives
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator

CHAPTER 1

INTRODUCTION

The internet has become the largest and most used source of information over the last decades [7]. However, the internet content is disorganized and has led the era of information into what some experts has called the era of misinformation [8]. The rapid expansion that the internet has experienced, makes it cumbersome to organize the data in a structure that meets all the user needs [9]. To solve this problems, different web mining [10], web page classification [5], and metadata techniques [11] have been implemented for the ease of web's resources organization, interoperability, discoverability, dissemination, and description. These strategies aim to improve web directories, digital libraries, general search engines, focused crawlers of vertical search engines, question and answering systems, web content filtering, and web browsing services [5].

Currently we can find multiple research focusing on the description and organization of the internet's educational resources or Learning Objects (LO) since they are the most abundant and used resources over the internet. However, research has focused in describing LO's in their metadata leading to multiple formats with different properties, controlled vocabularies, low quality descriptions, limited automated metadata generation, and reuse [11, 12]. Also, to annotate a LO's metadata or organize and maintain them in digital libraries or web directories by their educational properties is a difficult task. These approaches demands the need of manual expert review of each LO being described. This process is time consuming and makes it

difficult to properly describe all the existing LOs in the web as well as those to be published in the future [5, 11].

For example, in order to determine a LO's web page property such as the educational level, a combination of computing and educational experts has to manually inspect the LO's web page, study a particular controlled vocabulary and finally decide on a value to be assigned to the property. A worked example is presented in Appendix B.

Afterwards, the LO's web page can be properly annotated in its metadata specification or can be organized in a digital library or web directory as well as being indexed by search engines under an educational level. This phenomena impacts the organization, discoverability, interoperability, and reuse of educational resources. Also, limits e-learning applications and the search engine's ability to index LOs under educational levels [5].

In this work, we analyzed the task of determining a LO's web page educational level property automatically. Our approach is presented as a web mining problem in which we applied web page and text classification techniques aiming to determine if the educational level of a LO's web page could be learned with a classification algorithm.

Throughout this document, educational levels are described in their K-12 Main Categories and Sub Categories based on the 21st century pedagogical needs and interests of academicians [11, 13] under the United States [14] educational system. The Main Categories are composed of 3 labels: Elementary School (K to Grade 5), Middle School (Grade 6 to 8), High School (Grade 9 to 12) and the Sub Categories

are composed of 13 labels from K to Grade 12 [11, 14]. We used these categories and labels to perform two independent flat, multiclass, single label, soft and hard classification based on a LO’s web page on-page text features.

To perform our classification experiments, we compiled a data corpora of LO’s web pages with their educational level already annotated by experts under the Learning Application Readiness (LAR) metadata format available in the National Science Digital Library (NSDL) [11]. This format presents the most complete controlled vocabulary for the educational level when compared to other LO’s metadata formats. From these resources we selected those annotated with the educational levels and focus on a sub set of its controlled vocabulary [11]. The final data corpora is composed of a data corpus for the Main Categories of Educational Levels with 1,500 examples for each of its 3 classes and a data corpus with 375 examples for each of 13 K-12 labels of the Sub Categories. We selected the first occurrences for each class label which were retrieved from the NSDL search engine result sets and kept only those that are available in web pages from which we can find on-page features in HTML format. Our final data corpus for the Main Categories was of 4,500 records and the final data corpus of the Sub Categories was of 4,875 records.

To evaluate this work, we performed percentage split and k-fold cross validation metrics with Multinomial Naïve Bayes (MNNB), Support Vector Machine (SVM), and Maximum Entropy (MaxEnt) [15] classifiers and determined their precision, recall, F-measure and accuracy in each experiment for the Main Categories and the Sub Categories. We sought to achieve acceptable classification results of $\approx 80\%$ for the precision, recall, and accuracy and $\approx 60\%$ for the F-measure [5, 15–19].

With our experiments we demonstrated that the Main Categories of Educational Levels is a LO's property that be automatically determined by an statistical classifier algorithm and that the Sub Categories achieved poor classification results. The main contribution of this thesis is the determination of the educational level property of LO's web pages via an statistical classifier. This can be used for automated metadata generation, organization, and maintenance of digital libraries or web directories, web content filtering and web browsing of LOs. It also improves the search engine's indexing process of LOs and can be used in e-learning systems to organize LO's web pages under educational levels.

1.1 Motivation and Background

The internet has become the main and largest source of educational resources also know as learning objects [7, 20]. Academicians rely on the internet as a start point to seek a LO more than any other educational platform [21]. The internet's LO usage in education has proven to improve the knowledge attained by a learner for an educational goal [22, 23]. The 21st century pedagogical needs and interests of academicians [11, 13] encompasses the use of the web's LO in educational affairs as an enhancement to a learning experience.

However, often, academicians have problems finding the right LO for an educational need [24]. This may lead to an incorrect web resource usage, making the user fall into possible knowledge misconceptions [25, 26] or difficult the intention of finding a particular LO in the web. These problems slow down the adoption of LOs in formal educational systems.

To improve the description, organization, discoverability, interoperability, and reuse of LOs in the web, different general and vertical search engines, metadata,

and digital libraries have been implemented. Nonetheless, current research for LOs is largely based on metadata which is limited to describing and organizing LOs based on standardized formats and controlled vocabularies for specific contexts [17]. Moreover, multiple metadata formats have been introduced leading to low quality educational descriptions due to the heterogeneity in the specification [20]. Also, they have to be manually generated by human experts by inspecting the LO and assigning values to a metadata field in a particular format. Indeed, this manual organization of a LO into predefined categories is also seen in digital libraries or web directories. This activity is time consuming and often results in poor quality descriptions of a LO. This is due to the vast quantity of data available over the web as well as the new data being uploaded to web everyday. These circumstances lead to the inability of crawling and indexing all the web's data by search engines, describing their metadata and organizing it into digital libraries manually by humans [5].

Recently, web mining and web page classification techniques are being employed in different domains with the goal to automate the process of extracting knowledge from a web page's content, sentiment or structure and employ statistical classifiers to assign the web page into predefined categories. Regarding LOs, these techniques are scarcely applied. This reveals a gap in the use of these techniques in the automated description and organization of LOs. Most works of LOs focus in the LO's metadata, digital libraries, and general or vertical search engines [11, 20]. We do not know of any other work that employed statistical classifiers to categorize LO's web pages according to their educational level based in educational metadata with controlled vocabularies for the 21st century pedagogical needs and interests of academicians [11, 13] under the Educational System of the United States [14]. The works closest to ours was developed by Thompson et al. [27] where they classified

web pages into the resource types of assignments, syllabus, exams, and tutorials inspired by educational metadata formats. Also, Hassan et al. [17] encompassed the peculiarities of the internet in order to determine the educativeness of a LO. Both of these works are based in limited human annotations, restricted data sets and based in classification categories not necessarily based in the 21st century pedagogical needs and interests of academicians [11, 13]. In this work we established a procedure to determine the Educational Level of a LO in an automated way which has been missing in previous works since they are based on outdated specifications of a LO's relevant educational properties.

1.2 Problem Statement

Currently, LO's web pages are described and organized manually into metadata specifications, digital libraries and web directories by computer experts considering the opinion of experts in the educational field. Moreover, the educational level property of LO has recently risen as an important learning asset [11] for the description of a LO and has not been automated by any previous work regarding LOs in the web. The problem addressed in this work is that of reducing the process of manually describing and organizing LO's web pages by trying to automatically categorize the LO under its educational level. We established our task as a web mining and web page classification problem in which we employ text classification techniques. The aim is to determine if an statistical classifier can successfully categorize a LO's educational level based in on-page text features of its web page. Our approach serves as a basis for automatic generation and organization of metadata, digital libraries and web directories according the educational level categories for LOs.

1.3 Scope of the Work

This work employed web mining, web page classification and statistical text classifiers to categorize a LOs web pages under an educational level. We do not cover other properties of LO since they are partially covered in other works.

The notion of educational levels covered in this work is based on a sub set of the controlled vocabulary categories and labels of the LAR metadata format for this particular field. We chose the LAR metadata format vocabulary because it describes LOs according to the 21st century pedagogical needs and interests of academicians based on recent surveys [11, 13].

The *educational levels categories* in this work are established and referred through the rest of this document as *educational levels Main Categories and Sub Categories* as described in the LAR format and the Educational System in United States [14]. The Main Categories for K-12 are composed of the 3 labels: Elementary School (K to Grade 5), Middle School (Grade 6 to 8), High School (Grade 9 to 12) and the Sub Categories are composed of 13 labels from K to Grade 12. These were used as our class labels for our classification experiments. We used Multinomial NB, SVM and MaxEnt supervised learning algorithms since they have demonstrated to achieve acceptable results in similar text classification problems [5, 15, 17, 28].

Moreover, the classifiers are evaluated on a data corpus for the Main Categories with a total of 4,500 records and a data corpus for the Sub Categories with a total of 4,875 records of previously annotated LOs under their educational level based in the LAR controlled vocabulary. We retrieved web pages from the NSDL to build our data corpus only from those resources that were available in HTML. Other, formats such a PDF, WORD, JPEG, between others are not covered in this work since our focus is on categorizing a LO's web page contents under an educational level. As

features, we will use the on-page features of a web page such as HTML tags actual textual content. Other elements such as hyperlinks and the features of neighbors are not considered since we are not performing a structure or usage classification, neither a web site classification.

To evaluate our work, we performed percentage split and k-fold cross validation metrics [29]. For percentage split the data corpus was divided into the recommended ratios of 70% training data and 30% test data [18] and for k-fold cross validation we select the recommended $k = 10$ [15].

This work does not intend to automatically generate metadata, build a digital library, web directory or develop a search engine index for LO's described and organized under educational levels categories. The main focus of this work is to determine if an statistical classifier can acceptably categorize a LO's web page under an educational level so that it can be used to build or assist the previous mentioned applications.

1.4 Significance of the Study

Since the internet is mostly used for educational purposes [7] and LOs on the web are vast, disorganized and have to be manually described and organized under educational levels to reflect the 21st century pedagogical needs and interests of academicians [11, 13]. A methodology to determine a suitable statistical classifier for the categorization of a LO's web page under an educational level based on its on-page features is proposed. A system like this can eliminate the need for academicians to manually describe a LO and aid in the automatic generation of a LO's metadata for the educational level property. It can also be used to automatically build and maintain digital libraries or web directories based in open web resources.

Additionally, it can be used to produce ordered list of categories in which new web pages may be indexed by search engines. Moreover, researchers of LOs, web mining, web page classification, semantic web, data mining, text classification techniques, digital libraries, web directories, educational software, and e-learning systems as well as teachers and students seeking LOs in the web can benefit from this work.

1.5 Overview of the Methodology

In this work, we sought to automatically classify a LO's web page under and educational level. To achieve this, web mining, web page classification and text classification techniques were employed. To design our system we implemented the five Knowledge Discovery in Databases (KDD) steps: data acquisition, data preprocessing, data transformation, pattern extraction, and data analysis [1].

For data acquisition we retrieved a list of URLs of LO's web pages using the Java programming language pre-labeled within the educational levels. From this list, we retrieved a subset the URLs that were able to be parsed into its HTML content. We omitted sources such as .pdf, .doc, .jpeg, etc which require other type of classification analysis [5]. At the same time, the LO's web pages that we kept, were preprocessed as we selected a lower case version of text inside the HTML tags (on-page features). We selected all the available on-page text because not all web pages contains the same HTML tags due to the diversity of coding styles and programming languages syntaxes [5, 30]. With these pre-processed web pages, we built our data corpora composed of two data corpus.

We made a data corpus for the Main Categories of Educational Levels composed of 3 class labels: Elementary School, Middle School, and High School. The Elementary School class was composed of 250 instances for each sub class from K to Grade 5 for a total of 1,500 instances. The Middle School class was composed

of 500 instances for each sub class from Grade 6 to Grade 8 for a total of 1,500 instances. The High School class was composed of 375 instances for each sub class from Grade 9 to Grade 12 for a total of 1,500 instances. The final size of the this data corpus was 4,500 instances. The other data corpus was for the Sub Categories of Educational Levels composed of 13 class labels: one for each grade from K to Grade 12. Each class was composed of 375 instances for a total data corpus size of 4,875.

For each experiment, the data corpus was converted into the Attribute-Relation File Format (.arff) to which Weka machine learning tools can be applied [31]. Afterwards we transformed the data corpus into the vector space with the application of feature extraction, weighting, and selection. Once our corpus was reduced with the previous step, we proceeded to employ the pattern extraction algorithms. We run classification experiments with MNNB, SVM, and MaxEnt. To evaluate our results, we used the percentage split and k-fold cross validation metrics [15, 29]. For both of these strategies we analyzed the precision, recall, F-measure, and accuracy. We sought to obtain $\approx 80\%$ in precision, recall, accuracy, and $\approx 60\%$ in the F-measure [5, 15–19]. These values are based on the results observed in other text classification works [5, 15–19, 30].

1.6 Contributions

The efforts of this work makes the following contributions to science. We determined a methodology for applying educational models to web resources in an automated way. This was achieved by employing classification algorithms to LO’s web pages to determine its educational level. We found that the web pages text content can be used to successfully classify educational resources under the main categories of educational levels (Elementary School, Middle School, and High School). Also,

determined that classifiers can be used to generate metadata, build or maintain digital libraries for an educational resource web page under the main categories of the educational levels property. Additionally, we evidenced a lack of discriminatory effect of web page content for the sub categories of educational levels (K-12). Nevertheless, we applied the 21st century pedagogical needs and interest of academicians to the internet educational resources.

1.7 Organization of the Thesis

The document is organized as follows. The literature review in Chapter 2 introduces Knowledge Discovery in Databases, Data Mining, and their relations to the educational field and the internet. The theoretical background is presented in Chapter 3. This chapter describes the properties of educational materials or Learning Objects (LO) and how to apply classification algorithms to them. Chapter 4 discusses the methodology. We describe the tools that we are going to use, our system architecture, and we establish the our experiments goals. In Chapter 5 in which we discuss the evaluation metrics and present the results along with their discussion. Finally in Chapter 6 we conclude this work and recommend future work.

CHAPTER 2

LITERATURE REVIEW

In the following chapter, the literature review necessary to understand this thesis is discussed. Topics about Knowledge Discovery in Databases, data mining, educational data mining, web mining, web mining of educational data, the hyper-linked and semantic web, and their related systems are presented.

2.1 Knowledge Discovery in Databases

The purpose of the Knowledge Discovery in Databases (KDD) is to compile data from large data bank sources, organize them into useful groups of data to extract patterns and gain valuable knowledge from the analysis of the results [32]. It is common to find in literature that Data Mining and KDD are used interchangeably. However, it is important to note that data mining is a step in the KDD process. On one hand we have that KDD involves the process of data gathering, data preprocessing, data transformation, pattern extraction or data mining and the results analysis. On the other hand, data mining can be employed to extract patterns from a data bank without the need of a preprocessing or transformation step [1]. Figure 2.1 shows the steps, entities and actions involved in the KDD process in which data mining is encompassed.

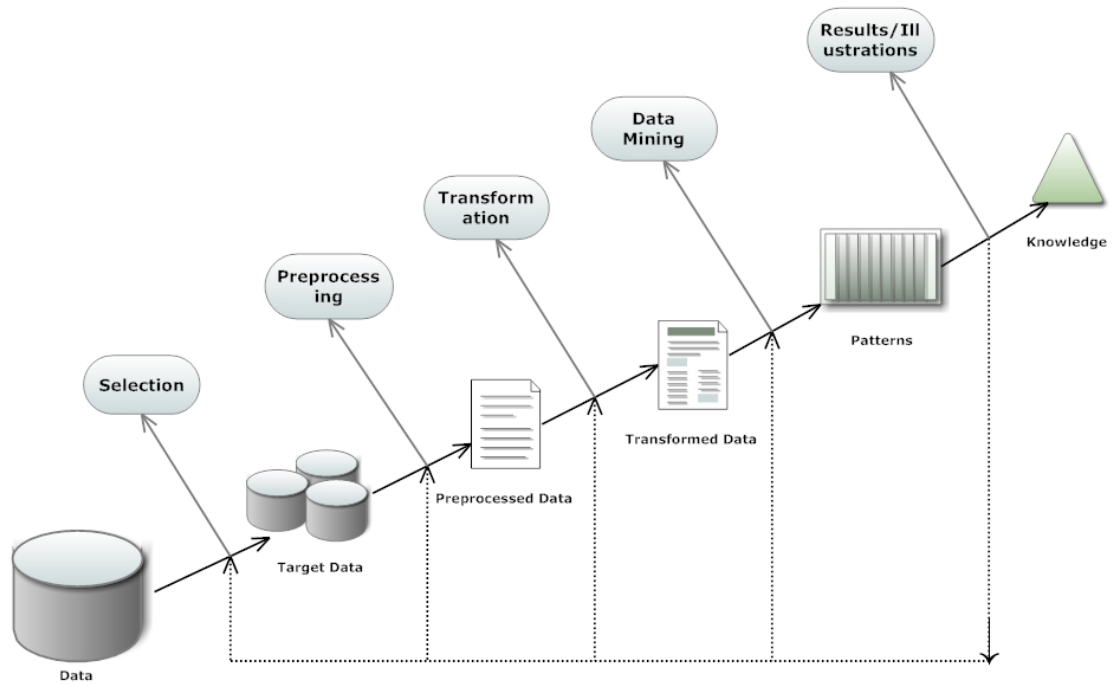


Figure 2–1: The Knowledge Discovery in Databases (KDD) technique [1]

The KDD cycle is decomposed into the following steps:

1. Data Gathering

The Data Gathering step is where the data to be analyzed is extracted in its original or raw form from its source.

2. Data Preprocessing

The Data Preprocessing step is where the raw data is parsed and cleaned from noisy data that can affect the focus of the experiment.

3. Data Transformation

The Data Transformation step is where the cleaned data is converted into data that can be read by the data mining algorithms. Commonly the data is converted into a feature space including techniques such as feature extraction, weighting and selection [33].

(a) Feature Extraction

Feature Extraction is the process of reducing the original data corpus into a feature space [33].

(b) Feature Weighting

Feature Weighting is a technique to gather information about the feature vectors such as its word occurrence in a document [33].

(c) Feature Selection

Feature Selection is a technique applied to find relevant features within the feature space before employing a data mining algorithm. [33].

4. Data Mining or Pattern Extraction

The Data Mining or Pattern Extraction step is where models are built (based in extracted patterns) using data mining algorithms. The type of algorithm to be used depends on the nature of the problem to solve. These algorithms can be based on classification, clustering, regression, etc.

5. Results Analysis

The results analysis gives knowledge about the performance of the model built by the data mining algorithm.

The KDD technique covers the whole process of discovering valuable knowledge from data. It is a non-trivial procedure which involves fine tuning of each step through trial and error. As Figure 2.1 shows, any step of the process can be revisited to be adjusted in order to obtain new results with different configurations. This can be repeated until satisfactory knowledge is obtained from the results.

In this thesis, we perform KDD on educational data over the internet. We gather educational web pages, preprocess and transform their data in order to perform

classification data mining algorithms and gain knowledge about their educational level.

internet

2.1.1 Educational Data Mining

Educational Data Mining (EDM) is an emergent research discipline that establishes its interest in crafting procedures for analyzing different types of data originated in educational environments. Those procedures are used to improve the usefulness of educational resources [1]. Figure 2.1.1 shows the cycle of EDM and the relationship between educational systems and the data mining technique in a social synergistic environment.

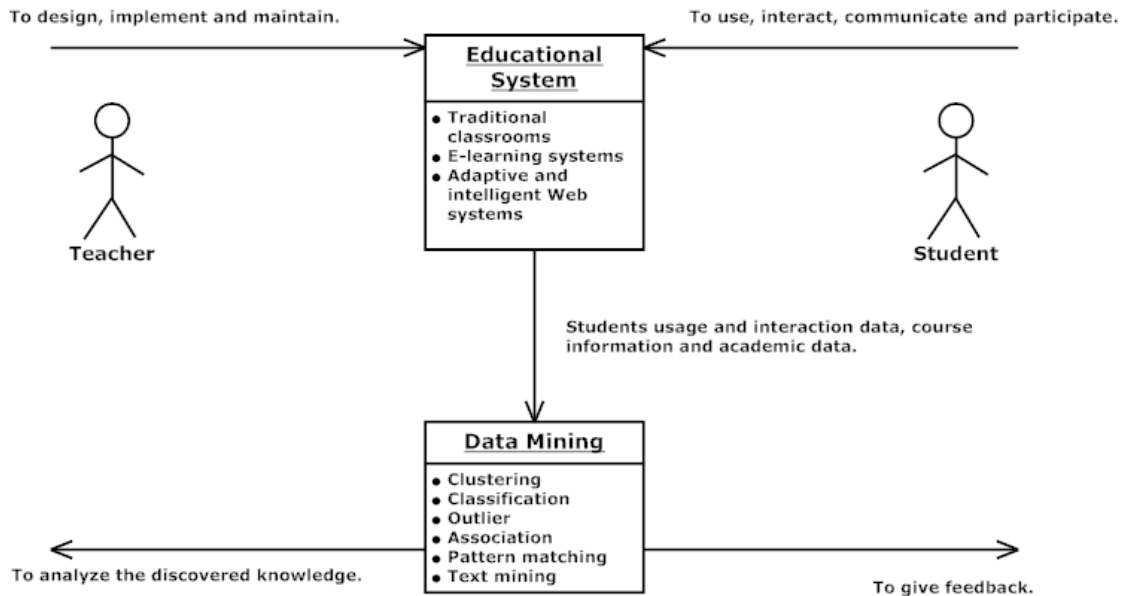


Figure 2-2: Educational Data Mining Cycle [2]

There are various EDM methods within the data mining models. These techniques are reduced to those relevant to educational sources. Sachin *et al.* [1], explain that the relevant methods to educational affairs are:

- prediction
- clustering
- relationship mining
- text mining
- outlier detection
- social network analysis

This work focuses on the text mining method of EDM based for web based platforms as it is an intention to classify educational web content under educational levels.

2.2 Web Mining

Web mining is an application of the data mining techniques over the internet. Web mining affairs are those of resource discovery, information extraction and pattern recognition generalization of the web's resources. Web mining tasks are described in a widely accepted taxonomy and its categories as explained by Klefodimos *et al.* are [10]:

1. Web Content Mining: Information and knowledge extraction of the Web's content. Text mining techniques are often applied in the classification of web pages and web documents to determine their subject, association patterns, etc.
2. Web Usage Mining: Knowledge extraction based in the user's activity over the internet. Data is extracted from web server logs, browser logs, cookies, and databases. The intention is to analyze the user's behavior and customize web services according to his/her particular interests.

3. Web Structure Mining: Knowledge extraction based on the association of the Web variety of objects such as web pages, multimedia, etc. These objects contain no unifying structure. They are related to one another via hyperlinks or social connections. Mining these connections can be done through Hyperlink Network Analysis (HNA) or Social Network Analysis (SNA). Web structure mining seeks to obtain knowledge from a web graph. It can be composed by the link connections between web pages or from other connections formed by the interaction of users in the social web. This technique is often used to determine the popularity or ranking of a web page. For example, the PageRank algorithm uses this technique.

2.2.1 Web Mining Content in Education

Similarly, the *Web Mining Taxonomy* can be applied to the Education field. Klefodimos *et al.* compiled in an overview [10] different works regarding Web Content Mining, Web Usage Mining and Web Structure Mining in Education. This work is related to Web Content Mining in Education as it is an intention to mine the content of Web pages.

Web Content Mining in Education research is interested in organizing, categorizing and retrieving valuable educational materials that can be used successfully in educational tasks [10]. This technique can be applied to:

- locating educational materials according to user needs in the open Web
- categorize educational materials under educational properties (e.g. topic area, resource type, etc)
- incorporating open Web resources in e-learning systems

In this work, we categorize LO's web pages under the educational level property using their web page text content to make the decision for its classification.

2.2.2 The Hyperlinked Web and The Semantic Web

The internet resources contains information about themselves as well as information about other related resources. They are related and organized under two approaches known as the Hyperlinked Web and the Semantic Web. Figure [2.2.2](#) shows the two web data organization approaches.

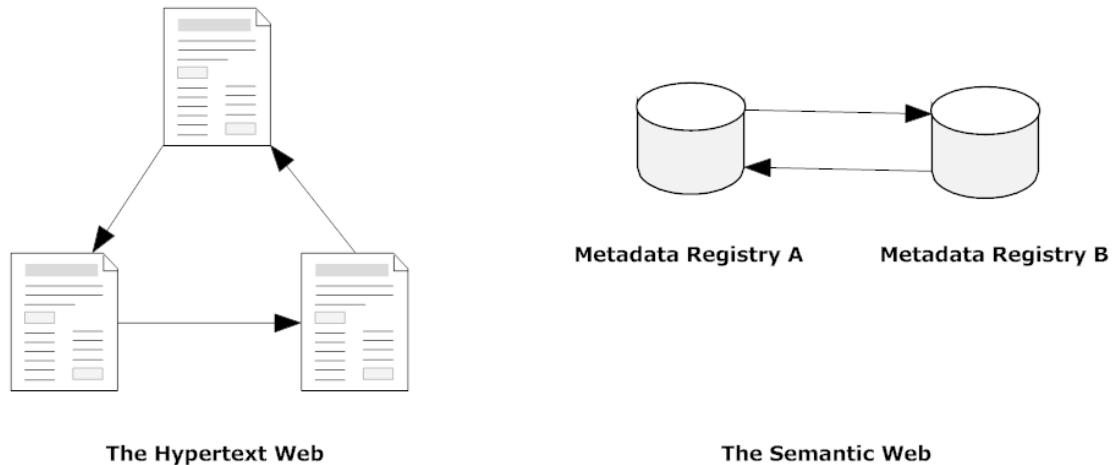


Figure 2–3: The Hyperlink vs Semantic Web organizations [3]

The Hyperlinked Web structure is composed of the raw source code found in web pages. This approach links documents in the web through the use of hyperlinks. For example a web page may contain links to other related web pages or it may populate its content by using resources in other URLs [3].

The Semantic Web is a way to implement artificial intelligence in the web for machine readable documents. It diverges from normal hyperlinks published in HTML since it seeks to treat description models of data as if it were one database [3]. This means that the storage of data in web documents can be analyzed intelligently by web browsers. The Semantic Web can be seen as an alternative to the relational/transactional database approach. It makes data attributes public to web machines for a faster querying of records [4]. The core of the Semantic Web mining starts with a definition of a formal ontology [4]. The ontology vocabulary is made of proofs and logic rules. They can be expressed in syntax specifications such as Resource Description Framework RDF or Ontology Web Language (OWL) Afterwards, they can be implemented in scripting programming languages such as

eXtensible Markup Language (XML) according to the World Wide Web Consortium (W3C) standards [34]. The Semantic Web Architecture can be seen in Figure 2.2.2.

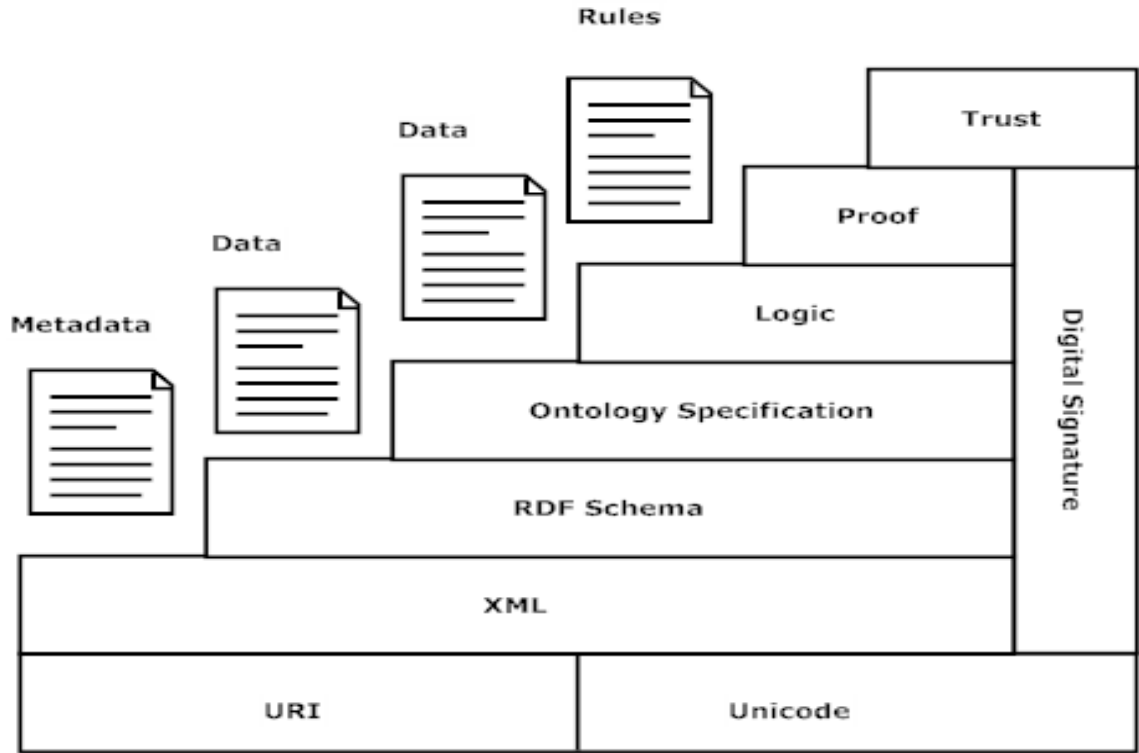


Figure 2–4: Semantic Web Architecture [4]

In this work, both web organizations are used synergistically as explained by Qi *et. al.* in [5]. In one hand we use the Semantic Web to extract pre-labeled properties of web pages of educational materials under controlled vocabularies. On the other hand, we exploit the Hyperlinked Web to perform our pattern extraction step by using the web page content.

2.2.2.1 Ontologies

An ontology is a logical description of an abstraction in a domain. It can be seen as crafted entity, made of a particular vocabulary used to describe a characteristic and a set of assumptions based in the meaning of the words [35]. Ontologies

intend to deliver explicit formal semantics for a conceptualization, normalized with a shared semantic agreement for the meaning of the terms in an expression. Finally it forms a well-defined vocabulary with logically demonstrated relations between entities [36]. The process of defining ontology in computer terms involves four steps [4]:

1. Abstracting the ontology to classes.
2. Establishing a hierarchy for those classes.
3. Specifying variables and their supported values.
4. Assigning values for variables for instances

Our use of ontologies permits us to infer meta-description of the targeted educational resources. This work uses educational ontologies. In particular, those regarding to educational levels. Their description provides with pre-labeled data under an educational property as basis for our classification task.

2.3 Related Work

The disorganization and exponential growth that the internet currently experiences, has raised interest in the Artificial Intelligence (AI) field as an intention to improve its Web services [7]. Particularly, we can find multiple works regarding the organization and description of LOs over the Web. It can be distinguished between works that focus on the retrieval of LO in general and vertical search engines, meta-data description of LOs and digital collections, and, the applications of LOs on e-learning platforms which is out of the scope of this work.

Recent work regarding LOs meta-data and digital libraries is described by the following authors:

Hodgins *et al.* [12] explains the IEEE Learning Object Metadata (LOM) specification as a medium for web educational resource sharing and interoperability. They introduced a meta-data standard for educational materials such it could completely describe all the Web's educational resources.

Sutton *et al.* introduced the Achievement Standards Network (ASN) [37] which describes educational standards in Resource Description Framework (RDF) with unique Uniform Resource Identifiers (URI's). Their meta-description promotes the semantic correlation of different educational standards by decomposing them into learning goals promoting their interoperability.

Ginger *et al.* [11] established the Learning Application Readiness (LAR) for educational resources meta data format. It includes meta tags specification about the subject, education level, resource type, audience, educational standards, and other elements. The educational standards tags specifies educational outcome text or preferably an ASN code related to its educational standard URI. This format is very similar to the LOM but its based on newer theory and human surveys [11, 12].

Moreover, these descriptions have been used for automated meta-data generation as shown by:

Pasanato *et al.* [38] developed a tool for semi automatic and automatic LOM meta data generation based on Wiki pages and user feedback. They successfully generated some of the LOM fields according to a human survey, but still unable provide reliable meta data for multiple fields such as Resource Type, Difficulty, and others.

Edvardsen *et al.* [39] analyzed document code from common file types to generate LOM for a reduced set of its specification. They studied automatic meta-data generation for Latex, Word and Power Point documents in which they were successful in determining the Title and the Language attributes of the LOM format.

Nevertheless, LO descriptions also play an important role in general and vertical (vertical refers to domain specific) search engines as shown by:

Curlango *et al.* [20] developed a search assistant to improve a teacher's search result of LOs. They used Google as the underlying search tool and developed an scenario for teachers to improve the presentation of their desired query for a LO. Their work suggest that LOs retrieval can be improved by filtering the Web content according to the teachers needs.

Shao *et al.* [40] developed a vertical search engine for educational documents in formats such as PDF, Word, PPT, HTML, etc. Their design was based in the Apache Lucene index. To improve the Lucene weight of terms based ranking, they considered recommended sources and factors of clicks to improve the relevant retrieval of documents.

Shah *et al.* [41] introduced a hybrid search engine (composed of a meta search engine and a topical search engine) with a re-ranking module for e-learning tutorials. They use Google, Yahoo, and Bing to search along with a topic searcher then further classify or re-rank its results based on topics. For re-ranking and classification they consider the author's profile, their experience, and highest degree.

Alatrash *et al.* [42] designed a ranking algorithm for biomedical literature using relevance feedback based on fuzzy logic and the Unified Medical Language System (UMLS). They rank the search results according to the UMLS ontology, providing high level mapping words under a medical domain. They found that ontological ranking gives more relevancy than low level meaning or statistical ranking and can improve the retrieval of medical resources over the Web.

As can be seen in previous works, the LOs research is oriented towards their description, organization, discoverability, dissemination, and reuse. However, we can see that most works focus on limited data sets or rely on human surveys for the validity of their results. To overcome this problem in different domains, Web mining and Web page classification has been successfully used as in works such as those by:

Miltsakaki *et al.* [15] performed web text classification in real time and analyzed the reading difficulty of the text. They achieved satisfactory results for the classification of the resources subject category such as arts, education, sports, and others. Also they calculated the reading difficulty of the text in the page using three different readability formulas. Their experiments were based on a pre-labeled data corpus of educational resources.

Chen *et al.* [43] classified Web pages according to their genre. They categorized web pages with the labels of: homepage, information search page, and, information and resource page. To achieve this they used on-page features for the classification and improved the precision and recall over previous similar works.

Hassan *et al.* [17] classified the Web's LO according to their educativeness based on a data set of human annotations by a mimic of an hypothetical student. They

found that the educativeness of document is property that can be used to improve the organization and retrieval of LOs over the Web.

Khade *et al.* [30] performed Web page classification based on the attractiveness of the page with supervised learning. In their experiments, they extracted on-page features to determine if a Web page was attractive to users or not. They achieved satisfactory classification results to improve the retrieval of Web pages.

Herzog *et al.* [44] applied a mathematical representation of Web page objects such as calculating the distance of buttons, input fields, and others for web automation. They used this approach for feature selection in a classification experiment which demonstrated satisfactory results in the identification of objects in web pages.

Our efforts are an intention to improve the description, organization, and retrieval of LOs presented in web pages. The educational data is still at a very early stage to be properly organized [11]. The successful effort to automate the categorization of the Web LOs under different educational properties such as the educativeness[17], suggests a similar approach for the educational level property. We do not know of any other work that has covered this task. Our approach can be used for automated metadata generation, the creation, organization and maintenance of digital libraries or Web directories. This replaces the process of manually determining the educational level of a LO's web page. The main focus of this work is to determine a suitable classification algorithm for such a task.

CHAPTER 3

THEORETICAL FRAMEWORK

In this thesis we perform KDD regarding educational materials over the internet. For the experiments of this work, its important to understand how educational materials are described in the internet focusing on their web pages. With their description in their meta-data and actual web page text content, we develop a process to categorize them under an educational level.

3.1 Educational Materials: Learning Objects

A learning object can be described as any resource that can be used for educational purposes. However, LO definitions vary through different works. Curlango *et al.* [20] differentiated between authors that describe a LO as any digital learning activity, while others also consider non-digital learning activities. Also, some say that they must have an educational purpose and others disagree with this property [20]. He also describes how this incongruence in the definition of a LO has lead to the disorganization, multiple metadata formats and poor discoverability of LOs.

In this work, we base our understanding of a LO as justified in the definition made by the Learning Standards Committee (LTSC) in their specification of the IEEE for LOs [12]. We accept this viewpoint since it is the most commonly accepted definition by researchers [20] which is stated as:

“any entity, digital or non-digital that may be used for learning, education or training”.

Examples of a LO includes: multimedia content, instruction and assessment materials, games, homework, tests, presentations, text content, and so on. It is important to note that this work is based on that definition, however, the scope of this work is restricted to digital entities since our experiments are focused in mining text content of web pages.

3.1.1 Learning Objects Properties

The properties of a LO are those that describe its intrinsic components. These descriptions intends to model learning objects according to their educational and computational attributes. They are described with educational properties such as educational level, difficulty level, learning objective, etc. Also a LO has computational properties such as its format, size, source, etc. Other properties are a LO’s educativeness, relevance, content category, resource type, expertise, and others [17]. Most of these properties are included in metadata formats to improve their description, organization, interoperability, and dissemination over the Web of LOs [11, 12].

3.1.1.1 Learning Objects Metadata formats

Metadata specifications for LOs are based on their properties. There exist multiple formats that can be used to describe a LO. Their difference is in the theory that supports the properties included in the specification. Some metadata formats consider a broader number of properties than others. Also, their minimum required fields to properly describe a LO differ. Examples of metadata for LOs are Learning Object Metadata (LOM) and Learning Application Readiness (LAR).

Both of these formats serve to describe a LO properly. However, this work is based on the LAR specification since it was recently created to have a better description of the 21st century educational needs [11]. For our experiments we constructed a data corpus of LO's web pages previously annotated with the LAR format. These resources are available through the NSDL Search Engine with open access.

The focus of this work is on the educational properties of a LO, particularly its educational level as described in the LAR specification. This property identifies the grade level being addressed by a LO. This is one of the most important properties that academicians seek when searching for LOs in the Web [13, 23].

3.2 Learning Object's Web Page classification

Web page content classification is essential for automated metadata generation, organizing and maintaining web directories or digital libraries, and improving search results. The effort of this work is to apply this strategy to automatically determine a LO's web page educational level being addressed by its content (Web content mining). This technique requires to extract textual features of a web page to make inference about its content. Its common to use statistical text classifiers to categorize the mined text. Before applying this technique, a classification type has to be established as well as determining which textual features will be used as an input to the classification algorithm [5].

3.2.1 Classification Type

A classification can be either binary or multiclass depending on the quantity of the category labels. When the instances are categorized under two different classes, we have a binary classification. In a multiclass classification we have more than two different classes to which instances can be assigned. Both of these strategies can

be divided into single-label and multi-label classifications. In the single-label case an instance belong to one and only one label. The multi-label approach can assign more than one label to a single instance. Also, they can be divided into soft or hard classifications. In soft classifications a instance is assigned to class based on a likelihood (e.x. a probability distribution) and in hard classification an instance is assigned or not to a class based on binary decision. Finally we can distinguish between the following types of classification as described by Qi *et al.* [5]:

- Binary classification: A classification in which we have two classes and an instance belongs to one of them.
- Multiclass, single-label, hard classification: A classification in which we have more than two classes and each instance can be assigned to one and only one class label based on a binary decision.
- Multiclass, single-label, soft classification: A classification in which we have more than two classes and each instance can be assigned to one and only one class label based on a likelihood.
- Multiclass, multi-label, hard classification: A classification in which we have more than two classes and each instance can be assigned to one or more class labels based on a binary decision.
- Multiclass, multi-label, soft classification: A classification in which we have more than two classes and each instance can be assigned to one or more class labels based on a likelihood.

Moreover, depending on the organization of category labels, we can perform a flat or hierarchical classification. The flat classification considers parallel categories and the hierarchical classification organizes instances in a hierarchy of categories [5]. In this work we perform a flat Multiclass, single-label classification as illustrated in Figure 3.2.1. We decomposed the hierarchical nature of the Educational Levels into a single flat classification per level (in our case 2 levels: Main Categories and Sub Categories as explained in Chapter 4).

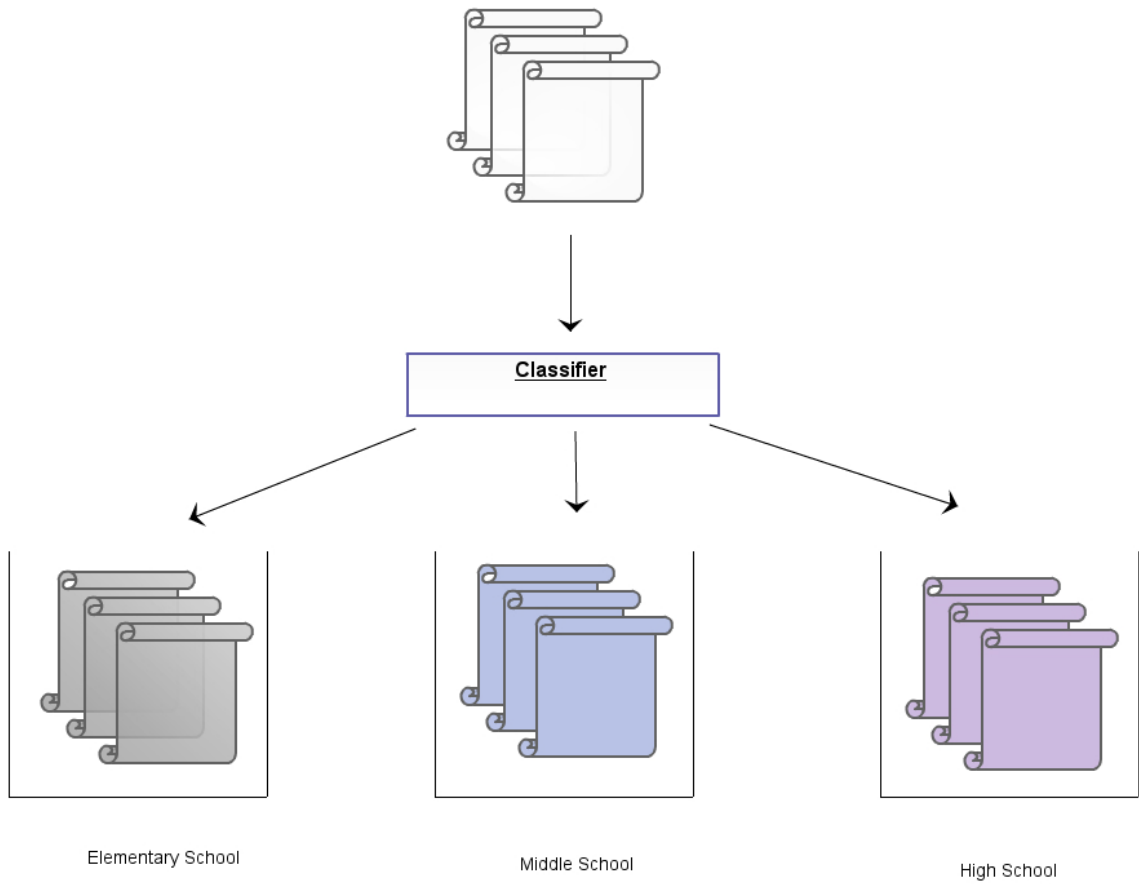


Figure 3–1: Muticlass, single-label classification [5]

This approach suits the need of our data since we are classifying according to the LAR standard for the educational levels.

3.2.2 Feature Selection

In classification experiments, the features are what is fed into the classifiers as a mean to perform its categorization. It is based on the features analysis that a classifier makes its decision to assign an instance to a particular category label. Regarding Web pages, we can find the following features that can be used in order to perform a classification [5]:

- On-page features: Actual visible text content of a Web page, page source code, text inside HTML tags.
- Features of the neighbors: Hyperlinked pages consideration. It includes a summarized version of the On-page features of the neighboring pages.
- Artificial Links: Considers Web pages in a ranked list for a particular query and can summarize their On-page features and Features of the neighbors.

In this work, our feature selection is based on the on-page features of a Web page. Due to the nature of our categories, it is not necessary to consider the features of the neighbors as we do not intend to classify a whole website. Consequently, Artificial Links is out of the scope of this work. Indeed, features of the neighbors and Artificial links are computationally expensive and troublesome for those who don't have industrial access to a search engine [5].

3.2.3 Classification Algorithms

The classification algorithms technique is based on deriving a mathematical model on the basis of a training set in which the categories of the instances are

known. Once the classifier is trained, a test set is given to it to determine the performance of the classification model on new data instances [45].

Classifiers are commonly applied in text categorization problems [18]. Sebastini *et. al* [46] describes the text classification function as seen in Equation 3.1. The notation below is as appears in [28].

$$\Phi : D \times C \rightarrow \{T, F\} \quad (3.1)$$

where

$D = \{d_1, d_2, d_3, \dots, d_{|D|}\}$ is the Domain of documents and,

$C = \{c_1, c_2, c_3, \dots, c_{|C|}\}$ is the set of pre-defined classes

-If an T value is assigned to an instance of the form $\langle d_i, c_j \rangle$ this indicates that the decision to assign document d_i to the class c_j is taken.

-If an F value is assigned to an instance of the form $\langle d_i, c_j \rangle$ this indicates that the decision to assign document d_i the class c_j is not taken.

The goal is to derive an unknown target function that can model the categorization of instances into classes. The output function is known as the classifier.

Despite the availability of numerous classification algorithms, none of them is said to be better than other for solving all the problems as explained by the *No*

Free Lunch Theorem [47]. But, they can be factored out for particular problems in which they have demonstrated to have better performance over others. Naïve Bayes, SVM, C4.5 neural networks, Logistic Regression, Decision Tree and MaxEnt classification algorithms are commonly used classifiers for text classification [5, 15, 17, 18, 28]. Between those, NB, SVM, and MaxEnt have reported the best performances [5, 15, 17, 28]. This suggests to use those three algorithms for future text classification tasks. This work will compare NB, SVM and MaxEnt to determine which one is the best performing classifier for our text categorization problem. These three classifiers are introduced in the next sections.

3.2.3.1 Naïve Bayes

The Naïve Bayes classifier is a simple and effective probabilistic method that is widely employed in text categorization methods. It uses conditional probabilities to estimate the category of a given document. The most important characteristic of the Bayes classifiers is that it assumes that all the attributes are independent in a given class context. This assumption makes Bayes a good choice for text classification tasks since the nature of a document is composed is composed of many different word and the classification decision will be based on single word and not in word phrases. To explain the mathematics of the Bayesian classification as expressed by Dan *et. al.* in [28], we introduce a document d_i that belongs to a class $C = \{c_1, c_2, c_3, \dots, C_{|c|}\}$. The probability of a class being mapped to a document is calculated by the Bayes equation as shown in Equation 3.2 as expressed in [28].

$$P(c_j|d_i) = \frac{P(c_j)P(d_i|c_j)}{P(d_i)} \quad (3.2)$$

The main goal in the Bayes classification is to calculate $P(C_j)$ which can be estimated from the frequency of instances in the training data and $P(d_i|c_j)$ which can be calculated from distributions such as the maximum likelihood model (MLM), Multinomial model (MN), Poisson model (PM), etc. The MN is the commonly used for the classification of multiple classes [28].

In this work we work with multiple classes, therefore we experiment with a multiclass classification and to calculate $P(d_i|c_j)$ we use the MN distribution. Based on this distribution, a document can be seen an ordered sequence of word events from a vocabulary V . It is assumed that the probability of word event in a document is independent of the context, position and length of the document. A document is taken from a multinomial distribution of independent trials of words corresponding to the document length. If we let N_{it} be a natural number symbolizing the number of times a word w_t occurs in a document, we can express the probability of a document given its class as in Equation 3.3 expressed in [28].

$$P(d_i|c_j) = P(|d_i|)|d_i|! \prod_{t=1}^{|V|} \frac{P(w_t|c_j)^{N_{it}}}{N_{it}!} \quad (3.3)$$

3.2.3.2 Support Vector Machine (SVM)

The SVM algorithm in its simplest form, separates two sets of data on the basis of training examples for both sets. It constructs a “decision surface” over the two sets in a hyperplane [18]. The idea is to maximize the separation between the two sets in the hyperplane by learning from training examples. Afterwards, new data can be categorized by calculating to which set it maps on the hyperplane [48]. SVMs rely in kernel equations which can transform data in one domain to another

domain in which the data become linearly separable. These equations may be linear, quadratic, etc [28]. Sassano in [49] described SVMs in theoretical form as follows:

Let a training data be represented in the form of:

$$(x_i, y_i), \dots, (x_l, y_l) \in R^n, y_i \in \{+1, -1\}$$

Then

$$g(x) = \text{sgn}(f(x))$$

is the decision function with:

$$f(x) = \sum_{i=1}^l y_i \lambda_i K(x_i, x) + b$$

subject to,

$$\forall_i : 0 \leq \lambda_i < C \text{ and } \sum_{i=1}^l y_i \lambda_i = 0$$

where the vectors x_i with $\lambda_i \neq 0$ are called support vector and C is the cost making a wrong decision and K is the Kernel function which in its linear case it can be expressed as:

$$K(x_i, x) = x_i \cdot x$$

Now we can re-write $f(x)$ as

$$f(x) = w \cdot x + b$$

where $w = \sum_{i=1}^l y_i \lambda_i x_i$

SVM training consists of finding λ_i and b by solving the maximization problem expressed with a Lagrange function as Equation 3.4.

$$\widetilde{L(\lambda)} = \sum_{i=1}^l \lambda_i - \frac{1}{2} \sum_{i=1}^l \lambda_i \lambda_j y_i y_j K(x_i, x_j) \quad (3.4)$$

restricted to

$$\forall_i : 0 \leq \lambda_i < C \text{ and } \sum_{i=1}^l y_i \lambda_i = 0$$

The previous equation determines the optimal decision boundary (optimal hyperplane) between two classes. However, in this work we will use multiclass SVMs (composed of 2 or more simple SVMs) as we have 3 main categories and 13 sub categories of educational levels to be separated in different hyperplanes [18]. In our case we use the one vs the rest strategy and the model is built by constructing one SVM for each class as stated in [50].

3.2.3.3 Maximun Entropy (MaxEnt)

The Maximun Entropy (MaxEnt) also know as Multinomial Logistic Regression is a multiclass version of the Logistic Regression (used for binary categories)

algorithm. Its categorization is based on determining if features are related to a category label by calculating the probability scores of the features [15]. The multinomial version of the Logistic Regression is used in this work since we run a multiclass classification. Yu *et. al.* [51] described the MaxEnt theoretical framework as follows:

Let x be a document, y a class, $w \in R^n$ the weight vector. The function $f(x, y) \in R^n$ describe the features extracted from the document x and the class y . Then,

$$P_w(y|x) = \frac{\exp(w^T f(x, y))}{\sum_{y'} \exp(w^T f(x, y'))}$$

If we have training samples with a count of N in the form of $\{(x, y)\}$ with $\{x_i\}$ grouped to l unique documents x_i , we can compute the empirical probability distribution $\tilde{P}(x_i, y) = \frac{N_{x_i, y}}{N}$. The number of times (x_i, y) appears in the training data is denoted by $N_{x_i, y}$.

Now, we can express the MaxEnt (ME) classifier as a regularized negative log-likelihood as in Equation 3.5

$$\begin{aligned} \min_w P^{ME}(w) &= - \sum_{i=1}^l \sum_y \tilde{P}(x_i, y) \log P_w(y|x_i) + \frac{1}{2\sigma^2} w^T w \\ &= \sum_{i=1}^l \tilde{P}(x_i) \log \left(\sum_y \exp(w^T f(x_i, y)) \right) - w^T \tilde{f} + \frac{1}{2\sigma^2} w^T w \end{aligned} \tag{3.5}$$

where σ is the misclassification cost, the marginal probability of x_i is $\tilde{P}(x_i) = \sum y \tilde{P}(x_i, y)$, then assuming $y_i \in Y = \{1, 2, 3, \dots, |Y|\}$, the expected vector of $f(x_i, y)$ is

$$\tilde{f} = \sum_{i=1}^l \sum_y \tilde{P}(x_i, y) f(x_i, y)$$

This work is based on the previous cases as we need to handle the multinomial case of a logistic regression. More details in the mathematics of the multinomial logistic regression can be seen in [\[51\]](#).

CHAPTER 4

METHODOLOGY

4.1 General Description

This chapter explains the methodology used in this work to solve the stated problem. We discuss the tools, steps, and techniques along with how they were integrated and used.

4.2 Tools

In this work, we used four different programming tools: Java, Jsoup, Weka and LibLINEAR. A description of these follows.

4.2.1 Java

The algorithms developed and used in this work were coded in Java. This programming language was chosen because other tools used in this work such as Weka and Jsoup are written in Java. This feature facilitates the application and extension of the functionalities of the other tools used in our work. Moreover, Java is portable across different operating systems and its code can be executed on any machine with Java Virtual Machine (JVM) installed.

Java has a rich API from which we use multiple classes to complement our work. The most important for this work are the classes included in the native packages of `java.net` and `java.io`. Within the `java.net` package we make use of the classes `URL` and `URLConnection`. They both work in conjunction and facilitate the web services necessary to connect to a Web Page and extract its HTML. With these classes we

collected our list of Web Page’s URLs by making a connection to the NSDL Search Engine and traversing the result sets. To access the HTML text of a connection, we used the `BufferedReader` and the `InputStreamReader` classes of the `java.io` package. Also, to write the extracted HTML text to a `.txt` file, we used the `PrintWriter` class which is also included in the `java.io` package. Finally, other feature that we used from Java was its implementation of threads from which we took advantage to retrieve multiple URLs in parallel processes to achieve a faster data gathering step.

4.2.2 jsoup

Jsoup is an open source HTML parser written in Java with the ability to extract and clean an HTML source code into a simple text document. In this work we included the libraries of this parser into our Java code to extract the plain text from the HTML tags of our retrieved Web Pages URLs.

4.2.3 Weka

Weka is a Java based open source tool focused in machine learning tasks. It provides powerful algorithms for all steps in the data mining process. In this work we used this tool to preprocess our data with the `stringToWordVector` and `attributeSelection` filters. Also, we ran our classification experiments using their implementations of `MNNB` and `MaxEnt` algorithms [31].

4.2.4 LibLINEAR

LibLINEAR is a Support Vector Machine and logistic regression library which contains all of their implementations in the primal and dual form. It supports multiclass classification using one vs the rest and Cramer and Springer strategies. This library has a version written in Java that can be embedded in Weka as a single plug-in. We used this library to perform our experiments of the SVM using the one

vs the rest multiclass strategy [52].

4.3 Design Approach

The design of our system is explained as a KDD technique that can be decomposed into five main steps which are data gathering, data preprocessing, data transformation, pattern extraction, and results analysis. It is important to note that the data mining process encompasses the possibility of going back to any previous step. Going back to a step is done in order to adjust any undesired behavior found in a more advanced step. An example would be going back to the data transformation step because the model built in the pattern extraction step did not achieved acceptable results. The nature of the KDD technique is non trivial and involves trial and error tuning. Through the development of this work some KDD steps were revisited to achieve the best results possible within our goals. The data preprocessing step was performed various times in which we experimented with all the text in all tags, only using the text in the <title>,<h1>,<h2>,<h3> tags and only using the text in <title>,<h1>,<h2>,<h3>,<p> tags. For each of these iterations we performed various configurations of data transformation and pattern extraction. The best results were obtained when using all the text. Therefore the methodology and results explained in this thesis are focused in that case. To visualize the KDD steps applied in this work, we illustrate our system architecture in Figure 4.3.

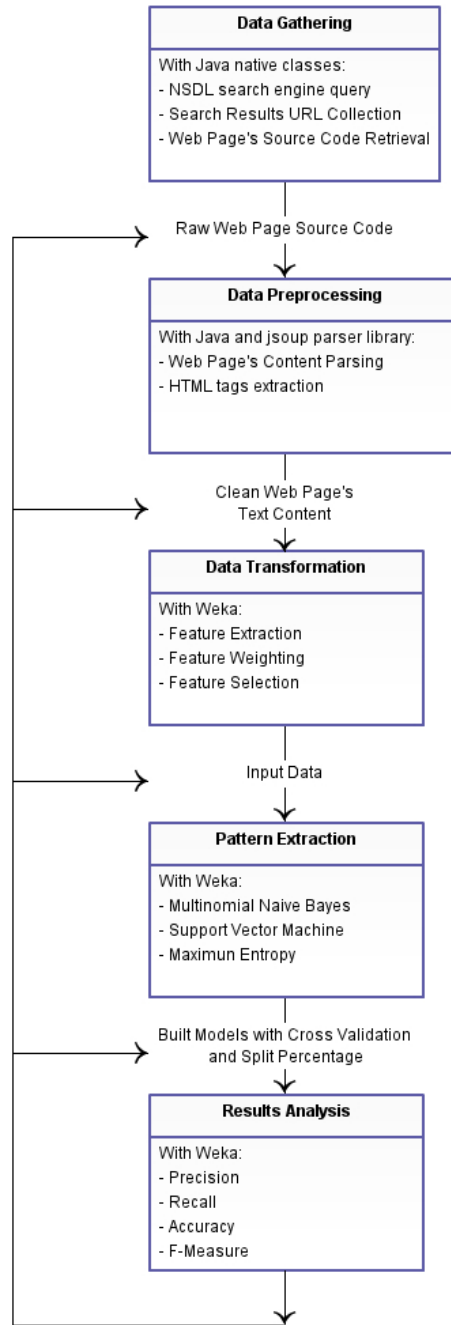


Figure 4-1: System Architecture

Each step of the system architecture is explained in the following section.

4.4 Experiments

Before starting to discuss the data gathering step, we justify the selection of the data retrieved in this work. For our experiments, we had to decide the category labels (dependent variables) on which our pattern extraction step was going to be based and choose our features (independent variables) to perform the prediction (evaluating the dependent variables with the independent variables) for each instance in our data corpora.

4.4.1 Determining Class Labels

To perform a classification, we need decide a name or label for each class. Into these classes is that we classify documents. Each document belongs to a particular class with a label. In the following section we define the class labels for our experiments.

4.4.1.1 Categories of Educational Levels

Our classification is based in the hierarchical structure of the Categories of Educational Levels for K-12. We define them as specified in the LAR format and based in the Education System of the Unites States [14]. We made two different groups of classes for each level of the hierarchy of the educational levels. Our organization was made into the Main Categories of Educational Levels and into the Sub-Categories of Educational Levels as seen in Figure 4.3.

We made this distinction of classes to treat the hierarchical structure of the Educational Levels into two single flat classifications steps in a top down fashion.

4.4.2 Data Gathering

To build the data corpora for our experiments, we retrieved a list of 4875 web page's URLs from the NSDL Search Engine services from a total of 62,483 available records with pre-labeled data with the main and sub categories of the educational levels. To achieve this step, we used Java to extract the URLs from the result sets of the NSDL Search Engine.

The Main Categories and Sub Categories data corpora was built as follows:

1. Data Corpus 1 (DC_{MC})

DC_{MC} is the data corpus of the Main Categories of Educational Levels composed 3 classes named Elementary School, Middle School and High School. The Elementary School class is composed of 250 instances for each sub class from K to Grade 5 for a total of 1,500 instances. The Middle School class is composed of 500 instances for each sub class from Grade 6 to Grade 8 for a total of 1,500 instances. The last class, High School is composed of 375 instances for each sub class from Grade 9 to Grade 12 for a total of 1,500 instances. The final size of the this data corpus is 4,500 instances to perform classification experiments.

2. Data Corpus 2 (DC_{SC})

DC_{SC} is the data corpus of the Sub Categories of Educational Levels composed of 13 classes, one for each grade from K-Grade 12. Each sub class was composed of 375 instances for a total data corpus size of 4,875.

Experiments were performed on the previous explained data corpora. We used two data corpus because each one contained LO's web pages pre-labeled differently. DC_{MC} was composed of 3 folders and DC_{SC} of 13 folders. Each folder represented a

class and contained the instances for it. Each instance was in the .txt format. This division was necessary so that the files could be loaded into Weka as an Attribute-Relation File Format (.arff) as explained below in the Preprocessing step.

4.4.3 Data Preprocessing

To start to clean the data, we used the library Jsoup to retrieve the HTML text content of all the tags found in web page source code. The text content of the Web Pages was saved into two different data corpus. One data corpus for the Main Categories and other for the Sub Categories of educational levels. It is important to note that a web page's source code is noisy and does not contain the same HTML tags. It may vary because of different coding styles and the web pages needs. Some web pages may need a particular set of HTML tags that are not needed in others [5]. Therefore, we experimented with the text of various different collection of tags and used those that attained better classification results. The collection of tags used are discussed below.

4.4.3.1 Web Pages Content Feature Selection

The features used in this work were based in the text available within the HTML tags of a Web Page that is seen as actual text when a web page is rendered by a web browser. We performed our experiments with the following collection of tags:

- Selecting commonly found tags in web pages such as: <title> which defines the title of a document, <h1> which defines the first heading, <h2> which defines the second heading and, <h3> which defines the third heading.
- Selecting the previous with the addition of a tag with descriptive text: <title>, <h1>, <h2>, <h3> and, <p> which defines a paragraph.

- Selecting all the tags that could be found in a web page.

Our selection is based on previous Web page classification works in which it is recommended to use a small and significant proportion of a Web page text. Moreover, the chosen tags are those which are basic to any HTML document since other tags are optional and some HTML documents may omit them [5].

However, based on our informal results, we omitted the reduced tag collection recommended in [5] since we achieved the best results by considering all the text in all the tags available for a particular web page. We followed this approach since not all web pages contains the same tags. This is due to the different coding styles and formats supported in the web [5]. If we would have limited our selection to a handful list of tags, some web pages in the data corpora would not contain them. Thus, impairing the classification results with those that have them.

Afterwards, once we had (DC_{MC}) and (DC_{SC}) , we loaded our data corpus into Weka by using the Simple Command Line Interface (SimpleCLI) and running the TextDirectoryLoader function to load all our .txt files into a single Attribute-Relation File Format (.arff) that can be interpreted by the Weka machine learning functions. Before building our model, we proceeded to reduce the feature space in order to improve the classification performance. In this step we performed feature extraction, weighting and selection by applying the Weka filters: StringToWord-Vector and AttributeSelection. These steps are decomposed in the following data mining step.

4.4.4 Data Transformation

The data transformation step (commonly explained as part of the data pre-processing step in other machine learning literature) turns a data corpus into a representative input for the classifiers. Data is transformed into what is called a feature space composed of feature vectors. In this work, web pages text is transformed into feature vectors to generate the feature space. These feature vectors are feed into the classifiers as their input. This transformation is necessary since the classifiers in this work do not operate in pure string words since they are represented with mathematical functions.

With $DC_M C$ or $DC_M S$ of the data corpora converted into .arff files, we were ready to start the transformation steps which includes feature extraction, weighting, and selection. We started by applying the stringToWordVector filter which perform feature extraction by converting string features into features with information about the word occurrence available in the strings of the text. Note that we performed different feature extraction steps as needed for different experimental questions.

4.4.4.1 Feature Extraction

Feature extraction is the process of reducing the feature space from the original set of data [33]. In this work we applied the StringtoWordVector filter of Weka which is used to express a document as a reduced vector space model. With this filter we also employed feature weighting since the output to this filter is a conversion of the string features into feature vectors with information about its word occurrence in a document [31]. To derive our feature space we employed the following techniques (using the stringToWordVector filter of Weka) with parameter recommendations found in [5, 28, 53]:

- Term Frequency-Inverse Document Frequency (TF-IDF) Transform: To determine how important a word is to a single file within the corpus.
- doNotOperateOnPerClassBasis: To restrict the maximum number of words and the minimum term frequency to be based on all the documents within the corpus and not in a per-class fashion.
- lowerCaseTokens: To convert to lower case all the words.
- outputWordCount: To have the real number for the times a word appears in a document.
- normalizeDocLength: To have an integer for the times a word appears in a document.
- stemmer: To derive words from a common stem.
- stopword removal: To remove particular words in which we used the default list of English stop lists
- tokenizer: To extract phrases of words we used N-Gram tokenizer considering one to three words in a phrase
- wordsToKeep: To set how many words we were going to keep per class.

After all the previous functions are set, we apply this StringToWordVector filter to our original data corpus to produce a reduced version of it in a vector representation form in order to make it a machine operable corpus.

4.4.4.2 Feature Weighting

Feature weighting is a technique to gather information about the feature vectors such as its word occurrence in a document [31]. A commonly used technique is the term frequency-inverse document frequency (TF-IDF) Transform [5, 18]. Therefore, we use it in this work. This equation is shown in Equation 4.4.4.2 and expressed in [18].

$$\text{TF-IDF}(t_k, d_j) = (t_k, d_j) \log \frac{|T_r|}{T_r(t_k)} \quad (4.1)$$

where (t_k, d_j) is the number of times t_k appears in d_j and $T_r(t_k)$ is the number of documents in which the term t_k appears or the document frequency of the term t_k the number of documents in which

that defines a term as t_K and $T_r(T_k)$ as the number of documents in which a term t_K is present.

4.4.4.3 Feature Selection

Feature selection is a technique applied to find relevant features within the feature space before employing a learning algorithm task [33]. Commonly employed functions in this technique is to calculate the Information Gain (IG) function of each feature and rank them with a certain threshold. Features with values below the threshold are discarded and not considered in the final feature space for the pattern extraction step. We use these techniques in this work to reduce the feature space as recommended in [53]. The IG equation is described in [31, 33] can be seen in Equation 4.2.

$$\text{InfoGain}(\text{Class}, \text{Feature}) = H(\text{Class}) - H(\text{Class}|\text{Feature}) \quad (4.2)$$

and,

$$H(\text{Class}) = - \sum_i P(\text{class}_i) \log_b P(\text{class}_i) \quad (4.3)$$

$$H(Class|Feature) = - \sum_{i,j} p(class_i, feature_j) \log \frac{p(feature_j)}{p(class_i, feature_j)} \quad (4.4)$$

In this work we applied the AttributeSelection filter of Weka as a technique to improve the classification performance. This technique is composed of an evaluator function and a search function as explained below:

- evaluator- InfoGainAttributeEval: To evaluate how much information a feature gives about a class. This commonly know as the Information Gain (IG) function [33].
- search- Ranker: To organize the features in a list based on their information gain. In this work we used this technique with a threshold of 0. This means that we only considered the attributes with information gain values above 0.

This final transformation step leaves our data corpus ready to build models.

4.4.4.4 Data Mining or Pattern Extraction

With the data preprocessed and transformed, we were ready to start building our classifiers for pattern extraction. We used MNNB, SVM and MaxEnt classifiers with percentage split and k-fold cross validation techniques. The validation techniques are explained in Chapter 5

4.4.4.5 Results Analysis

The results analysis gives knowledge about the performance of the classifiers when predicting a particular class label. We sought to obtain values of $\approx 80\%$ in precision, recall, accuracy and $\approx 60\%$ in the F-measure which show that a classifier

is acceptably predicting class labels as seen in other classification works [5, 15–19]. Our results are discussed in Chapter 5

4.5 Experimental Questions

By applying the previous techniques we seek answers to:

1. Which classifier (between MNNB, SVM, and MaxEnt) achieves the best results in predicting the Main Categories of educational levels?

To answer this question we run an experiment in the data corpus (DC_{MC}) using NB, SVM and MaxEnt in which we applied Feature extraction and Weighting using the stringToWordVector of Weka with the following settings:

- Term Frequency-Inverse Document Frequency (TF-IDF) Transform: TRUE
- doNotOperateOnPerClassBasis: TRUE
- lowerCaseTokens: Note, this was already performed in the pre-processing step.
- outputWordCount: True
- normalizeDocLenght: True
- stemmer: NULL
- stopword removal: True
- tokenizer: N-Gram Tokenizer (1 to 3)
- wordsToKeep: 1,000 words.

Also, we applied feature selection using the AttributeSelection filter of Weka as follows:

- evaluator- InfoGainAttributeEval: applies to (IG) function
- search- Ranker: With a threshold of 0

2. Can the on-page features of a LO's Web page be used to classify it under a label of the Main Categories of Educational levels?

To answer this question we use the results of the best classifier of the first question.

3. Which labels of the Main Categories can be successfully predicted?

This question is answered by observing the results of question 2 along with its confusion matrix.

4. Can the on-page features of a LO's Web page be used to classify it under a label of the Sub Categories of Educational levels?

To answer this question we run a similar experiment to the one in question 1 in the data corpus (DC_{MS}). Moreover, we will only run the experiment using the best performing classifier of question 1. To answer this question we run an experiment in the data corpus (DC_{MC}) using NB, SVM and MaxEnt in which we applied Feature extraction and Weighting using the stringToWordVector of Weka with the following settings:

- Term Frequency-Inverse Document Frequency (TF-IDF) Transform: TRUE
- doNotOperateOnPerClassBasis: TRUE
- lowerCaseTokens: Note, this was already performed in the preprocessing step.
- outputWordCount: True
- normalizeDocLenght: True
- stemmer: NULL
- stopword removal: True
- tokenizer: N-Gram Tokenizer (1 to 3)
- wordsToKeep: 5,000 words.

Also, we applied feature selection using the AttributeSelection filter of Weka as follows:

- evaluator- InfoGainAttributeEval: applies to (IG) function
- search- Ranker: With a threshold of 0

5. As stated in the Introduction of this thesis: *in order to determine a LO's Web page educational level, a combination of computing and educational experts has to manually inspect the LO's Web page, study the controlled vocabulary for this LO's property and finally decide to which category of educational levels it will be assigned.*

Can this process be automated?

To answer this question we have to analyze the results of all the previous questions.

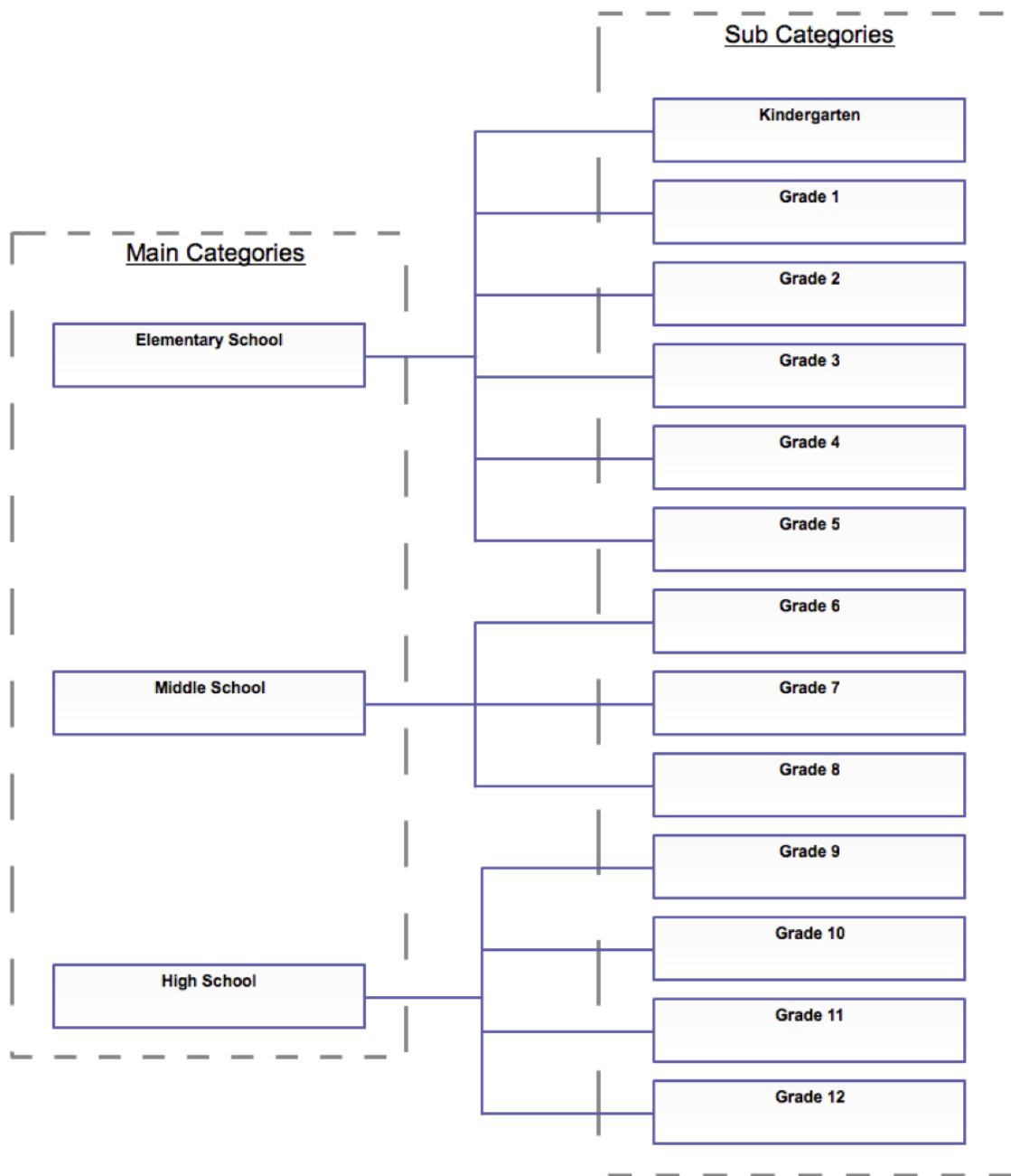


Figure 4-2: Educational Levels

CHAPTER 5

EXPERIMENTAL RESULTS

The following section, explains the evaluation metrics and discusses the results for each experimental question.

5.1 Evaluation Metrics

Most works only use one evaluation procedure and multiple classifiers but, its been suggested to use both multiple classifiers and multiple evaluation techniques in order to gain more information from our results [29]. Our classifiers were evaluated based in the percentage split and k-fold cross validation techniques.

In percentage split, the data corpus is divided into $x\%$ training data and $y\%$ test data. The recommended ratios are 70% for training data and 30% test data, although this can vary depending on the nature of the data corpus. Its common to employ different percentage splits depending on the nature of the problem. In this work we will use the recommended values [18, 29].

In k-fold cross validation, the data corpus is divided into k partitions. In an iterative process, a partition is selected as a training set and the rest serve as the test set. This is repeated for every partition and the average is calculated to determine the final result. The recommended k value is 10, although this can vary depending on the nature of the data corpus. In this work we used the recommended values through the experiments [15, 29].

With both approaches, we analyzed the precision, recall, accuracy and, F-Measure of the MNNB, SVM, MaxEnt classifiers in each of the educational levels Main Categories and the Sub Categories. These values are be averaged for all labels in each category label (in this case 3 labels in the Main Categories and 13 labels for the Sub Categories) [31]. To explain these metrics, we first introduce the variables of a confusion matrix as in [18]:

- True Positives (TP) is the number of documents correctly assigned to a category.
- False Positives (FP) is the number of documents incorrectly assigned to a category.
- True Negatives (TN) is the number of documents correctly un-assigned to a category.
- False Negatives (FN) is the number of documents incorrectly un-assigned to a category.

The Precision is defined as the probability that if a random document is classified, it is correct. This can be seen as the classifiers “degree of soundness” [18]. The precision indicates that the algorithm is correctly classifying random documents in the data set. Equation 5.1 shows its mathematical expression.

$$P = \frac{TP}{TP + FP} \quad (5.1)$$

The Recall is the probability that if a random document is to be classified under a category, the decision is taken. This is known as the “degree of completeness” of the classifier [18]. The recall indicates if classifier can acceptably classify new

documents. Equation 5.2 demonstrates how it is calculated.

$$R = \frac{TP}{TP + FN} \quad (5.2)$$

The accuracy determines the overall performance of a classifier[31]. Equation 5.3 shows how to calculate it.

$$A = \frac{TP + FN}{TP + FP + TN + FN} \quad (5.3)$$

The F-measure is a calculation of the harmonic mean between precision and recall. It relates the arithmetic mean and the geometric mean of the precision and the recall. This measure indicates the quality of the classifier [18]. This equation shows a trade-off between the precision and recall [43]. It is expressed as in Equation 5.4.

$$F = \frac{2RP}{R + P} \quad (5.4)$$

5.2 Experimental Questions Results

After we analyze all the previous equations for each classifier, we proceed to make inferences about our experimental questions. To have a insight of which values for these metrics would achieve acceptable results, we observed similar works [5, 15–19] in which accepted values ranged from $\approx 60\%$ in F-measure and $\approx 80\%$ in accuracy, precision and recall.

The following answers only present the results obtained for cross validation from which we obtained the best results. A complete list of the results can be seen in Appendix A.

To answer the first question:

1. Which classifier (between MNNB, SVM and MaxEnt) achieves the best results in predicting a Main Category of the Educational Levels?

The overall performance of each classifier is:

SVM overall Accuracy= (91.1111%)

MaxEnt overall Accuracy= (85.6889 %)

MNNB overall Accuracy= (71.5111%)

Also, we analyze the individual results of the three classifiers as shown in Table 5-1.

Table 5-1: Results Comparison

Classes	MNNB			SVM			MaxEnt		
	P	R	F	P	R	F	P	R	F
Elementary School	0.781	0.633	0.699	0.940	0.888	0.913	0.895	0.789	0.838
Middle School	0.606	0.703	0.651	0.867	0.902	0.884	0.827	0.851	0.839
High School	0.786	0.810	0.798	0.930	0.943	0.936	0.855	0.931	0.891
Wiegthed Avg	0.724	0.715	0.716	0.912	0.911	0.911	0.859	0.857	0.856

As we can see from the results, the SVM classifier achieved the best performance, followed by MaxEnt then MNNB. Although, they all performed relatively well, we can see SVM hard classification scheme is better suited for this task.

To answer the second question:

2. Can the on-page features of a LO's Web page be used to classify it under a label of the Main Categories of Educational levels?

We proceed to observe the individual results of the best performing classifier and analyze its results per class and its confusion matrix. The individual results are shown in Table 5-2 and in Table 5-3.

Correctly Classified Instances or Overall Accuracy: 4100 (91.1111%)

Incorrectly Classified Instances: 400 (8.8889%)

Total Number of Instances: 4500

Table 5–2: Support Vector Machine 10-Fold C-V in DC_{MC}

Class	Precision	Recall	F-Measure
Elementary School	0.940	0.888	0.913
Middle School	0.867	0.902	0.884
High School	0.930	0.943	0.936
Weighted Avg.	0.912	0.911	0.911

Table 5–3: Support Vector Machine 10-Fold C-V in DC_{MC} Confusion Matrix

Class	Elementary School	Middle School	High School
Elementary School	1332	142	26
Middle School	66	1353	81
High School	19	66	1415

Our results demonstrate that the on-page features of LO’s web pages can be used to predict the Main Categories of Educational Levels.

To answer the third question:

3. Which labels of the Main Categories can be successfully predicted?

Our results in question 2 demonstrated that all the labels can be successfully predicted. It is important to note that the High School label achieved the best results among the other two labels. This indicates that the LO’s Web pages for High School contain more discriminative on-page features than the other two classes.

To answer the fourth question:

4. Can the on-page features of a LO’s Web page be used to classify it under a label of the Sub Categories of Educational levels?

We proceed to observe the individual results of the best performing classifier and analyze its results per class and its confusion matrix. The individual results are shown in Table 5-4 and in Table 5-5.

The overall performance of the model is:

Correctly Classified Instances: (1324) = 27.159%

Incorrectly Classified Instances: (3551) = 72.841%

Total Number of Instances: 4875

Table 5-4: Support Vector Machine 10-Fold C-V in DC_{SC}

Class	Precision	Recall	F-Measure
Kindergarten	0.566	0.549	0.558
Grade 1	0.488	0.480	0.484
Grade 2	0.346	0.328	0.337
Grade 3	0.326	0.301	0.313
Grade 4	0.184	0.200	0.192
Grade 5	0.310	0.272	0.290
Grade 6	0.418	0.432	0.425
Grade 7	0.068	0.339	0.354
Grade 8	0.083	0.427	0.381
Grade 9	0.077	0.133	0.125
Grade 10	0.079	0.035	0.035
Grade 11	0.079	0.011	0.011
Grade 12	0.024	0.024	0.024
Weighted Avg.	0.272	0.272	0.271

Table 5-5: Support Vector Machine 10-Fold C-V in DC_{SC} Confusion Matrix

Class	Kindergarten	Grade 1	Grade 2	Grade 3	Grade 4	Grade 5	Grade 6	Grade 7	Grade 8	Grade 9	Grade 10	Grade 11	Grade 12
Kindergarten	206	65	66	4	9	5	5	2	13	0	0	0	0
Grade 1	54	180	93	13	6	6	4	9	8	1	0	0	1
Grade 2	76	82	123	37	17	8	6	8	13	1	1	1	2
Grade 3	8	25	37	113	112	51	11	1	13	2	0	0	2
Grade 4	9	5	16	106	75	124	6	8	20	4	1	0	1
Grade 5	4	6	11	50	149	102	9	10	19	8	4	0	3
Grade 6	3	3	3	5	7	11	162	74	74	14	7	6	6
Grade 7	3	1	1	2	14	12	89	127	106	13	1	3	3
Grade 8	1	2	4	9	13	5	70	89	160	6	2	6	8
Grade 9	0	0	1	4	1	2	14	8	26	50	104	79	86
Grade 10	0	0	0	2	2	1	3	2	4	133	13	116	99
Grade 11	0	0	0	1	2	1	4	1	5	95	118	4	144
Grade 12	0	0	0	1	0	1	5	4	4	97	109	145	9

Based on the poor results, the Sub Categories of Educational Level could not be acceptably learned based in a LO's Web page on-page features.

To answer the fifth question (main research question):

5. As stated in the Introduction of this thesis: *in order to determine a LO's Web page educational level, a combination of computing and educational experts has to manually inspect the LO's Web page, study the controlled vocabulary for this LO's property and finally decide to which category of educational levels it will be assigned.* Can this process be automated?

By comparing the good results obtained in the classification results of the Main Categories and the poor results obtained for the Sub Categories, we see that the complete hierarchy decomposed into two hard classifications could not be acceptably learned. We determined that by using the on-page features of a LO's Web page, the Main Categories of Educational Levels can be acceptably learned and the Sub Categories could not.

5.3 Results Discussion

Based on our results from each experiment, we determined that the Main Categories of Educational Levels can be acceptably learned. The best performing classifier was the SVM, followed by the MaxEnt and then MNNB on the data corpus of the Main Categories. This model achieved the best results compared to MNNB and MaxEnt due to its mathematical representation of vectors where the other two are based in probabilistic measures. Given the data corpora and the preprocessing and transformation techniques employed in this work, the decision to assign a document to class is better employed in a vectorial and hard classification instead to

determining a probability of assign document to a class in a soft classification.

For the SVM, we achieved weighted average values for the Precision of 91.2% indicating that the classifier is acceptably assigning random Web Pages LO's to their class, 91.1% in the Recall which means that the classifiers is acceptably assigning random Web Pages LO's of a class to its actual class, 91.1% in the F-measure to indicates a ratio between the Precision and Recall which tells us the accuracy of a particular classification is acceptable and a high Accuracy of 91.1% which tells us that the model is acceptably classifying LO's Web Pages. The class label with the highest results was High School with a Precision of 93.0%, Recall with 94.3% and a F-measure of 93.6%. This is related to the content of a LO's Web Page which indicate that the High School LO's Web Pages contains more discriminative features that the other two class labels.

Also, with experiments performed for the Sub Categories of Educational Levels, we obtained poor results in the Precision, Recall, F-Measure and overall Accuracy of 27.159% for the SVM classifiers. This results indicated that in the data corpus and pre-processing techniques used in this experiment, these categories could not be acceptably classified. These results are highly related to having multiple classes, in this case 13 classes, and, a relatively low examples for each class. In the confusion matrix can be noted by the diagonal, that most errors were misclassification in neighboring grades based on our class distribution.

Finally, we have demonstrated that Main Categories of Educational Levels can be classified with high precision, recall, F-Measure and Accuracy and that Sub Categories of Educational Levels could not, based in a LO's Web page on-page features

and the experimental procedure used in this work.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusions

In this thesis, we sought to automatically determine the Educational Level of a LO's Web Page. We constructed a data corpora composed of a data corpus with the Main Categories of Educational Levels and other with the Sub Categories of Educational Levels. By running a multiclass classification under the three labels of the Main Categories, namely Elementary School, Middle School and High School, we determined that the Main Categories of educational levels can be successfully predicted based on its LO's Web page on-page features. There is not much statistically significant differences between these labels as the results vary slightly. However, we found that the High School label achieved the best results over the other two labels. This demonstrates that the High School web pages contains more discriminative features than the other two labels.

Moreover, in the data corpus for the Main Categories, we compared three classification algorithms in which all achieved satisfactory results. However, the best performance was achieved by SVM, followed by MaxEnt, and MNNB. These results shows that these classification algorithms can be used for automatic metadata generation and the organization of digital libraries and web directories. This approach eliminates the process of manually determining a LO's Web page under the Main Categories of Educational Levels.

Also, with experiments performed on the data corpus of the Sub Categories of Educational Levels, we determined that these labels could not be acceptably learned based on its LO's Web page on-page features. From our results in the confusion matrix for the Sub Categories we noted that most of the misclassifications occurred in the neighbor values of the diagonal. We also noted that the misclassification occurred the most within each group of the Main Categories. For example, for K to Grade 5 (Elementary School), the misclassification occurred between those grades. The same behavior was observed for Grades 6 to 8 (Middle School) and for Grades 9 to 12 (High School). This shows that neighboring grades contains similar educational activities. These LO's web pages are similar in their educational text content. This phenomena can be attributed to the different topics covered by schools in different grades. What is covered in first grade in one school may be covered in second grade in another school and so on.

Finally we conclude that the upper level of the hierarchy or Main Categories of Educational Levels can be successfully classified using its LO's Web page on-page features and that more experiments need to be carried out for the Sub Categories of Educational Levels. Our results demonstrate that LO's web pages contain discriminative features when treated as groups (Elementary School, Middle School and, High School) but not enough discriminative features when treated individually (K-12).

6.2 Future Work

From our experiments, we can point out various trends for future work. To improve the results of such classification experiments, we suggest to increase the data corpus size by including more examples for each class. Also, results may improve by considering other features of the web page such as different HTML tags collection and the features of neighboring web pages by deep linking. It can also

be experimented with web usage and structure mining. Both of these approaches may improve the results when used alone or in conjunction with web content mining.

Additionally, in our experiments we do not analyze the semantic of the LO's web page text content. By considering the skill level of the LO presented in a web page the classification for the Sub Categories may be improved. We suggest experimenting with the Bloom's Taxonomy of Cognitive Levels which can be a better indicator for resource being of a particular grade. Natural Language Processing (NLP) techniques may be used in this approach.

Indeed, another approach that can be done is to divide the classification of the lower level of the hierarchy (Sub Categories) by splitting the task into simpler classifications for each of the Main Categories. A hierarchical classification experiment can be performed in which the the first classification is done at the upper level of the hierarchy and from those results, classify the lower level of the hierarchy. This would be to classify each Main Category independently and then use another classifier to categorize its particular Sub Categories.

Another suggestion is to consider different class labels configurations of the educational levels to train the classifiers with grades that are not neighbors. An example could be to build a binary classifier with the classes being Kindergarten and Grade 12. This would be also done with the subsequent grades. With this approach the neighboring misclassifications that occurred in the Sub Categories may be eliminated.

Another trend would be to consider other properties of Learning Objects such as their type of resource (content, assessment or pedagogy) or if it is an assignment

or a test. These properties are annotated in the LAR metadata description. We suggest to follow a similar approach to the one in this work but with other educational properties.

We also suggest to experiment with other type of formats such as multimedia materials. This type of classification would be more challenging due to their need of image, audio, or video processing. Also, it would interesting to classify the educational properties of the speech of an educator in real time. To achieve this, speech processing is needed to convert the speech to text and then classify it.

All these suggestions would improve the quality, organization, discoverability and re-use of educational materials over the internet. It also facilitates and improves the use of these resources in formal education to enrich the knowledge that a learner may acquire through the course of life.

APPENDICES

APPENDIX A

EXPERIMENTS RESULTS PER CLASSIFIER

A.1 Experiment 1: Results per Classifier

A.1.1 Multinomial Naive Bayes Results

A.1.1.1 Multinomial Naive Bayes 10-Fold Cross Validation Results

- Correctly Classified Instances or Overall Accuracy: 3218 (71.5111%)
- Incorrectly Classified Instances: 1282 (28.4889%)
- Total Number of Instances: 4500
- Total number of features: 975

The independent probability of each class

1. Elementary School 0.3333333333333333
2. Middle School 0.3333333333333333
3. High School 0.3333333333333333

Table A–1: Multinomial Naive Bayes 10-Fold Cross Validation

Class	Precision	Recall	F-Measure
Elementary School	0.781	0.633	0.699
Middle School	0.606	0.703	0.651
High School	0.786	0.810	0.798
Weighted Avg.	0.724	0.715	0.716

Table A–2: Multinomial Naive Bayes 10-Fold Cross Validation Confusion Matrix

Class	Elementary School	Middle School	High School
Elementary School	949	434	117
Middle School	232	1054	214
High School	34	251	1215

A.1.1.2 Multinomial Naive Bayes Split Percentage Results

- Correctly Classified Instances or Overall Accuracy: 959 (71.037%)
 - Correctly Classified Instances: 391 (28.963%)
 - Total Number of Instances: 4500
 - Total number of features: 975
1. Elementary School 0.3333333333333333
 2. Middle School 0.3333333333333333
 3. High School 0.3333333333333333

Table A–3: Multinomial Naive Bayes Split Percentage

Class	Precision	Recall	F-Measure
Elementary School	0.860	0.634	0.730
Middle School	0.608	0.669	0.637
High School	0.705	0.835	0.765
Weighted Avg.	0.728	0.710	0.711

Table A–4: Multinomial Naive Bayes Split Percentage Confusion Matrix

Class	Elementary School	Middle School	High School
Elementary School	300	122	51
Middle School	45	295	101
High School	4	68	364

A.1.2 Support Vector Machine Results

A.1.2.1 Support Vector Machine 10-Fold Cross Validation Results

- Correctly Classified Instances or Overall Accuracy: 4100 (91.1111%)
- Correctly Classified Instances: 400 (8.8889%)
- Total Number of Instances: 4500
- Total number of features: 975

Table A–5: Support Vector Machine 10-Fold Cross Validation

Class	Precision	Recall	F-Measure
Elementary School	0.940	0.888	0.913
Middle School	0.867	0.902	0.884
High School	0.930	0.943	0.936
Weighted Avg.	0.912	0.911	0.911

Table A–6: Support Vector Machine 10-Fold Cross Validation Confusion Matrix

Class	Elementary School	Middle School	High School
Elementary School	1332	142	26
Middle School	66	1353	81
High School	19	66	1415

A.1.2.2 Support Vector Machine Split Percentage Results

70% Train Set, 30% Test Set

- Correctly Classified Instances or Overall Accuracy: 1197 (88.6667%)
- Correctly Classified Instances: 153 (11.3333%)
- Total Number of Instances: 4500
- Total number of features: 975

Table A–7: Support Vector Machine Split Percentage

Class	Precision	Recall	F-Measure
Elementary School	0.921	0.863	0.891
Middle School	0.827	0.875	0.850
High School	0.916	0.924	0.920
Weighted Avg.	0.889	0.887	0.887

Table A–8: Support Vector Machine Split Percentage Confusion Matrix

Class	Elementary School	Middle School	High School
Elementary School	408	56	9
Middle School	27	386	28
High School	8	25	403

A.1.3 MaxEnt Results

A.1.3.1 MaxEnt 10-Fold Cross Validation Results

- Correctly Classified Instances or Overall Accuracy: 3856 (85.6889 %)
- Correctly Classified Instances: 644 (14.3111%)
- Total Number of Instances: 4500
- Toal Number of Features: 975

Table A–9: MaxEnt 10-Fold Cross Validation

Class	Precision	Recall	F-Measure
Elementary School	0.895	0.789	0.789
Middle School	0.827	0.851	0.851
High School	0.855	0.931	0.931
Weighted Avg.	0.859	0.857	0.857

Table A–10: MaxEnt 10-Fold Cross Validation Confusion Matrix

Class	Elementary School	Middle School	High School
Elementary School	1183	201	116
Middle School	102	1277	121
High School	37	67	1396

A.1.3.2 MaxEnt Split Percentage Results

70% Train Set, 30% Test Set

- Correctly Classified Instances or Overall Accuracy: 252 (73.4694%)
- Correctly Classified Instances: 91 (26.5306%)
- Total Number of Instances: 4500
- Toal Number of Features: 975

Table A–11: MaxEnt Split Percentage

Class	Precision	Recall	F-Measure
Elementary School	0.868	0.736	0.796
Middle School	0.788	0.807	0.797
High School	0.801	0.913	0.853
Weighted Avg.	0.820	0.816	0.815

Table A–12: MaxEnt Split Percentage Confusion Matrix


Class	Elementary School	High School	Middle School
Elementary School	348	72	53
High School	39	356	46
Middle School	14	24	398

APPENDIX B

EXAMPLE

As an example, we explain the manual process of determining the educational level of the LO's web page by a teacher and discuss our methodology to automate it.

If a teacher wants to determine the educational level of the following LO's web page <http://www.learner.org/interactives/dailymath/cooking.html> as seen in Figure [B-1](#), [B-2](#), [B-3](#).



Monthly Update sign up

[in](#)
[f](#)
[t](#)

Teacher resources and professional development across the curriculum

[About Us](#)
[Video Series](#)
[Professional Development](#)
[Course & Video Licensing](#)
[Lesson Plans](#)
[Interactives](#)
[News & Blog](#)

Interactives

Choose One

MATH IN DAILY LIFE

How do numbers affect everyday decisions?

+54305463365463
+292345678901234

Cooking by Numbers

- Introduction
- Playing to Win
- Savings and Credit
- Population Growth
- Home Decorating
- Cooking by Numbers
- The Universal Language
- Related Resources

Not all people are chefs, but we are all eaters. Most of us need to learn how to follow a recipe at some point. To create dishes with good flavor, consistency, and texture, the various ingredients must have a kind of relationship to one another. For instance, to make cookies that both look and taste like cookies, you need to make sure you use the right amount of each ingredient. Add too much flour and your cookies will be solid as rocks. Add too much salt and they'll taste terrible.

Ratios: Relationships between quantities

That ingredients have relationships to each other in a recipe is an important concept in cooking. It's also an important math concept. In math, this relationship between 2 quantities is called a ratio. If a recipe calls for 1 egg and 2 cups of flour, the relationship of eggs to cups of flour is 1 to 2. In mathematical language, that relationship can be written in two ways:

1/2 or 1:2

Both of these express the ratio of eggs to cups of flour: 1 to 2. If you mistakenly alter that ratio, the results may not be edible.

Working with proportion

All recipes are written to serve a certain number of people or yield a certain amount of food. You might come across a cookie recipe that makes 2 dozen cookies, for example. What if you only want 1 dozen cookies? What if you want 4 dozen cookies? Understanding how to increase or decrease the yield without spoiling the ratio of ingredients is a valuable skill for any cook.

Let's say you have a mouth-watering cookie recipe:

Figure B-1: Example LO's Web Page part 1 [6]

1 cup flour
 1/2 tsp. baking soda
 1/2 tsp. salt
 1/2 cup butter
 1/3 cup brown sugar
 1/3 cup sugar
 1 egg
 1/2 tsp. vanilla
 1 cup chocolate chips

This recipe will yield 3 dozen cookies. If you want to make 9 dozen cookies, you'll have to increase the amount of each ingredient listed in the recipe. You'll also need to make sure that the relationship between the ingredients stays the same. To do this, you'll need to understand proportion. A proportion exists when you have 2 equal ratios, such as 2:4 and 4:8. Two unequal ratios, such as 3:16 and 1:3, don't result in a proportion. The ratios must be equal.

Going back to the cookie recipe, how will you calculate how much more of each ingredient you'll need if you want to make 9 dozen cookies instead of 3 dozen? How many cups of flour will you need? How many eggs? You'll need to set up a proportion to make sure you get the ratios right.

Start by figuring out how much flour you will need if you want to make 9 dozen cookies. When you're done, you can calculate the other ingredients. You'll set up the proportion like this:

$$\begin{array}{rcl} \text{1 cup} & & \text{3} \\ \text{flour} & & \text{dozen} \\ \hline \text{X cups} & \times & \text{9} \\ \text{flour} & & \text{dozen} \end{array}$$

You would read this proportion as "1 cup of flour is to 3 dozen as X cups of flour is to 9 dozen." To figure out what X is (or how many cups of flour you'll need in the new recipe), you'll multiply the numbers like this:

$$\begin{array}{l} \text{X times 3 = 1 times 9} \\ \text{3X = 9} \end{array}$$

Now all you have to do is find out the value of X. To do that, divide both sides of the equation by 3. The result is $X = 3$. To extend the recipe to make 9 dozen cookies, you will need 3 cups of flour. What if you had to make 12 dozen cookies? Four dozen? Seven-and-a-half dozen? You'd set up the proportion just as you did above, regardless of how much you wanted to increase the recipe.

What if your recipe has metric measurements? Find out more about the metric system in "Meters and Liters: Converting to the Metric System of

Figure B-2: Example LO's Web Page part 2 [6]

Math in Daily Life -- Cooking by Numbers

<http://www.learner.org/interactives/dailymath/cooking.html>

Measurements."

"Math in Daily Life" is inspired by programs from For All Practical Purposes.

[Home](#) | [Catalog](#) | [About Us](#) | [Search](#) | [Contact Us](#) | [Site Map](#) |

[Tweet](#) (30)



© Annenberg Foundation 2014. All rights reserved. [Legal Policy](#)

Figure B-3: Example LO's Web Page part 3 [6]

After inspecting the web page text content, the teacher has to determine to which educational level this LO belongs. A teacher can assign this LO to Middle School based on the Learning Application Readiness (LAR) controlled vocabulary for the educational level property.

Assume that we have a decision model (classifier) built based on previous examples for Elementary School, Middle School, and High School. Given such a classifier we can extract the text content of new examples (in this case the LO's web page seen in Figure [B-1](#), [B-2](#), [B-3](#)) and give it as an input to the classifier so that it automatically assign this LO's web page to the Middle School category. This strategy eliminates the need for a human to inspect and determine the LO's web page educational level manually. To illustrate the classification step, we provide a worked example in the following section.

B.1 Worked example

In this example we use Multinomial Naive Bayes as our classifier. The goal of this example is to illustrate how a classifier is built and then used to assign an unknown document to a class. Assume the data corpus in Table B-1.

Table B-1: Worked Example Data Corpus

	Doc	Words	Class
Training	1	Count Grade 1 Understand	Elementary School
Training	2	Draw Grade 3	Elementary School
Training	3	Ratio Grade 6 Ratio	Middle School
Training	4	Ratio Grade 7 Proportion	Middle School
Training	5	Multiply Grade 11 High School	High School
Training	6	Divide Grade 9 Grade 10	High School
Test	7	Ratio Proportion Proportion	?

In this example we have 6 training documents (these are 6 LO's web pages) with pre-labeled classes from which we will build our classifier. The Test example seen in Figure B-1, B-2, B-3 is the document that we want to categorize under an Educational Level.

Training Step:

Prior probabilities:

$$P(c) = \frac{N_c}{N}$$

where c is class N_c is the number of documents for a class and N is the total number of documents,

$$P(\text{ElementarySchool})=2/6$$

$$P(\text{MiddleSchool})=2/6$$

$$P(HighSchool)=2/6$$

Conditional Probabilities:

$$P(w|c)= \frac{count(w, c) + 1}{count(c) + |V|}$$

where w is a word and $|V|$ is the vocabulary cardinality (number of unique words in the data corpus). In this case the vocabulary is: Count, Grade 1, Understand, Draw, Grade 3, Ratio, Grade 6, Grade 7, Proportion, Multiply, Grade 11, High School, Divide, Grade 9, and Grade 10. $|V| = 15$ then,

$$P(Ratio|ElementarySchool)=(0+1) / (5+15)= 1/20$$

$$P(Proportion|ElementarySchool)= (0+1) / (5+15)= 1/20$$

$$P(Ratio|MiddleSchool)= (3+1) / (6+15)= 4/21$$

$$P(Proportion|MiddleSchool)= (1+1) / (6+15)= 2/21$$

$$P(Ratio|HighSchool)=(0+1) / (6+15)= 1/21$$

$$P(Proportion|HighSchool)=(0+1) / (6+15)= 1/21$$

Test Step:

To evaluate to which class Doc 7 (the Test example) belongs:

$$P(ElementarySchool|Doc7) \propto \frac{2}{6} * \frac{1}{20} * \frac{1}{20} * \frac{1}{20} \approx 0.00004$$

$$P(MiddleSchool|Doc7) \propto \frac{2}{6} * \frac{4}{21} * \frac{2}{21} * \frac{2}{21} \approx 0.00028$$

$$P(HighSchool|Doc7) \propto \frac{2}{6} * \frac{1}{21} * \frac{1}{21} * \frac{1}{21} \approx 0.00003$$

By comparing the previous probabilities, we can see that Doc 7 would be assigned to the Middle School class since it achieved the highest probability between $P(ElementarySchool|Doc7)$, $P(MiddleSchool|Doc7)$ and $P(HighSchool|Doc7)$.

REFERENCES

- [1] R Barahate Sachin and M Shelake Vijay. A survey and future vision of data mining in educational field. In *2012 Second International Conference on Advanced Computing & Communication Technologies (ACCT)*, pages 96–100. IEEE, 2012.
- [2] Shengjian Liu. Educational web mining applications in intelligent web-education systems. In *2011 International Conference on Information Technology, Computer Engineering and Management Sciences (ICM)*, volume 4, pages 254–257, 2011.
- [3] Chuan Zhang and Ruoman Zhao. The construction of teaching resource library based on semantic web. In *2011 3rd International Workshop on Intelligent Systems and Applications (ISA)*, pages 1–4. IEEE, 2011.
- [4] Amar Nayak, Jitendra Agarwal, Vinod Kumar Yadav, and Shadab Pasha. Enterprise architecture for semantic web mining in education. In *2009 ICCEE'09. Second International Conference on Computer and Electrical Engineering*, volume 2, pages 23–26. IEEE, 2009.
- [5] Xiaoguang Qi and Brian D Davison. Web page classification: Features and algorithms. *ACM Computing Surveys (CSUR)*, 41(2):12, 2009.
- [6] Annenberg Foundation. Math in daily life, cooking by numbers. <http://www.learner.org/interactives/dailymath/cooking.html>.
- [7] Maninder Kaur, Nitin Bhatia, and Sawtantar Singh. Web search engines evaluation based on features and end-user experience. *International Journal of Enterprise Computing and Business Systems*, 1(2), 2011.

- [8] Miriam J Metzger. Making sense of credibility on the web: Models for evaluating online information and recommendations for future research. *Journal of the American Society for Information Science and Technology*, 58(13):2078–2091, 2007.
- [9] Li Guang-ming and Geng Wen-juan. Research and design of meta-search engine oriented specialty. In *2010 International Symposium on Intelligence Information Processing and Trusted Computing (IPTC)*, pages 204–207. IEEE, 2010.
- [10] Alexandros Kleftodimos and Georgios Evangelidis. An overview of web mining in education. In *Proceedings of the 17th Panhellenic Conference on Informatics*, pages 106–113. ACM, 2013.
- [11] Kathryn Ginger and Letha Goger. Evaluating the national science digital library for learning application readiness. *Proceedings of the American Society for Information Science and Technology*, 48(1):1–4, 2011.
- [12] IEEE Learning Technology Standards Committee et al. Draft standard for learning object metadata. *Accessed July*, 14:2002, 2002.
- [13] Christine L Borgman. *Fostering learning in the networked world: The cyber-learning opportunity and challenge*. DIANE Publishing, 2011.
- [14] U.S. Network for Education Information: U.S. Department of Education. Structure of u.s. education. Retrieved November 2013.
- [15] Eleni Miltsakaki and Audrey Troutt. Real-time web text classification and analysis of reading difficulty. In *Proceedings of the Third Workshop on Innovative Use of NLP for Building Educational Applications*, pages 89–97. Association for Computational Linguistics, 2008.
- [16] Syahidah Sufi Haris and Nazlia Omar. A rule-based approach in blooms taxonomy question classification through natural language processing.
- [17] Samer Hassan and Rada Mihalcea. Learning to identify educational materials. *ACM Transactions on Speech and Language Processing (TSLP)*, 8(2):2, 2011.

- [18] Anwar Ali Yahya and Addin Osman. Automatic classification of questions into bloom's cognitive levels using support vector machines. In *The International Arab Conference on Information Technology, Naif Arab University for Security Science (NAUSS), Riyadh, Saudi Arabia*, pages 1–6, 2011.
- [19] Norazah Yusof and Chai Jing Hui. Determination of bloom's cognitive level of question items using artificial neural network. In *2010 10th International Conference on Intelligent Systems Design and Applications (ISDA)*, pages 866–870. IEEE, 2010.
- [20] Cecilia Curlango-Rosas, Gregorio A Ponce, and Gabriel A Lopez-Morteo. A specialized search assistant for learning objects. *ACM Transactions on the Web (TWEB)*, 5(4):21, 2011.
- [21] Eleanor Wombwell and Dan Smith. Student and teacher views of the internet. *ACM Inroads*, 2(1):38–41, 2011.
- [22] James W Pellegrino, Naomi Chudowsky, Robert Glaser, et al. *Knowing what students know: The science and design of educational assessment*. National Academies Press, 2001.
- [23] Carlin Llorente Shelley Pasnik. Digital learning objects potential to support early learning. Education Development Center / SRI International, 2011.
- [24] Shaochun Xu, Xuhui Chen, and Dapeng Liu. Classifying software visualization tools using the bloom's taxonomy of cognitive domain. In *2009 CCECE'09 Canadian Conference on Electrical and Computer Engineering*, pages 13–18. IEEE, 2009.
- [25] Michelene TH Chi. Commonsense conceptions of emergent processes: Why some misconceptions are robust. *The journal of the learning sciences*, 14(2):161–199, 2005.
- [26] Burcin Acar Sesen and Elif Ince. Internet as a source of misconception: radiation and radioactivity. *TOJET*, 9(4), 2010.

- [27] Cynthia Thompson, Joseph Smarr, Huy Nguyen, and Christopher D Manning. Finding educational resources on the web: Exploiting automatic extraction of metadata. In *International Workshop & Tutorial on Adaptive Text Extraction and Mining held in conjunction with the 14th European Conference on Machine Learning and the 7th European Conference on Principles and Practice of*, page 79, 2003.
- [28] Li Dan, Liu Lihua, and Zhang Zhaoxin. Research of text categorization on weka. In *Intelligent System Design and Engineering Applications (ISDEA), 2013 Third International Conference on*, pages 1129–1131, 2013.
- [29] Q Al-Radaideh. The impact of classification evaluation methods on rough sets based classifiers. *International Arab Conference for Information Technology (ACIT 2008)*, 2008.
- [30] Ganesh Khade, Sudhakar Kumar, and Samit Bhattacharya. Classification of web pages on attractiveness: A supervised learning approach. In *2012 4th International Conference on Intelligent Human Computer Interaction (IHCI)*, pages 1–5. IEEE, 2012.
- [31] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.
- [32] Suhem Parack, Zain Zahid, and Fatima Merchant. Application of data mining in educational databases for predicting academic trends and patterns. In *2012 IEEE International Conference on Technology Enhanced Education (ICTEE)*, pages 1–4. IEEE, 2012.
- [33] Pádraig Cunningham. Dimension reduction. In *Machine learning techniques for multimedia*, pages 91–112. Springer, 2008.
- [34] World Wide Web Consortium. World wide web consortium. <http://www.w3.org/>, 1, 2013.

- [35] Peter Mika. Ontologies are us: A unified model of social networks and semantics. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(1):5–15, 2007.
- [36] Juan Ye, Lorcan Coyle, Simon Dobson, and Paddy Nixon. Ontology-based models in pervasive computing systems. *The Knowledge Engineering Review*, 22(4):315–347, 2007.
- [37] Stuart Allen Sutton and Diny Golder. Achievement standards network (asn): an application profile for mapping k-12 educational resources to achievement standards. In *International Conference on Dublin Core and Metadata Applications*, pages 69–79, 2008.
- [38] Luciano TE Pansanato and Renata PM Fortes. Strategies for automatic lom metadata generating in a web-based cscl tool. In *Proceedings of the 11th Brazilian Symposium on Multimedia and the web*, pages 1–8. ACM, 2005.
- [39] Lars Fredrik Høimyr Edvardsen, Ingeborg Torvik Sølvsberg, Trond Aalberg, and Hallvard Trætteberg. Automatically generating high quality metadata by analyzing the document code of common file types. In *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries*, pages 29–38. ACM, 2009.
- [40] Lei Shao, Jianwei Li, and Xuerong Gou. Research and design of a vertical search engine for educational resources. In *2011 International Conference on Advanced Intelligence and Awareness Internet (AIAI 2011)*, pages 159–163. IET, 2011.
- [41] Axita Shah, Sonal Jain, Rushabh Chheda, and Avni Mashru. Model for re-ranking agent on hybrid search engine for e-learning. In *2012 IEEE Fourth International Conference on Technology for Education (T4E)*, pages 247–248. IEEE, 2012.
- [42] Massuod Alatrash, Hao Ying, Peter Dews, Ming Dong, and R Michael Massanari. Ranking biomedical literature search result based on relevance feedback using fuzzy logic and unified medical language system. In *Fuzzy Information*

- Processing Society (NAFIPS), 2012 Annual Meeting of the North American*, pages 1–6. IEEE, 2012.
- [43] Guangyu Chen and Ben Choi. Web page genre classification. In *Proceedings of the 2008 ACM symposium on Applied computing*, pages 2353–2357. ACM, 2008.
 - [44] Christoph Herzog, Iraklis Kordomatis, Wolfgang Holzinger, Ruslan R Fayzrakhmanov, and Bernhard Krüpl-Sypien. Feature-based object identification for web automation. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pages 742–749. ACM, 2013.
 - [45] Mindy K Ross, Ko-Wei Lin, Karen Truong, Abhishek Kumar, and Mike Conway. Text categorization of heart, lung, and blood studies in the database of genotypes and phenotypes (dbgap) utilizing n-grams and metadata features. *Biomedical informatics insights*, 6:35, 2013.
 - [46] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.
 - [47] David H Wolpert. The supervised learning no-free-lunch theorems. In *Soft Computing and Industry*, pages 25–42. Springer, 2002.
 - [48] Md Mursalin, Motaher Hossain, Md Kislu Noman, et al. Performance analysis among different classifier including naive bayes, support vector machine and c4.5 for automatic weeds classification. *Global Journal of Computer Science and Technology*, 13(3), 2013.
 - [49] Manabu Sassano. Virtual examples for text classification with support vector machines. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 208–215. Association for Computational Linguistics, 2003.
 - [50] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425,

2002.

- [51] Hsiang-Fu Yu, Fang-Lan Huang, and Chih-Jen Lin. Dual coordinate descent methods for logistic regression and maximum entropy models. *Machine Learning*, 85(1-2):41–75, 2011.
- [52] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.
- [53] M Indra Devi, R Rajaram, and K Selvakuberan. Generating best features for web page classification. *Webology*, 5(1):Article–52, 2008.

CLASSIFICATION OF LEARNING OBJECT'S WEB PAGES UNDER EDUCATIONAL LEVELS

Manuel J. Orán-Hernández

Department of Electrical and Computer Engineering

Chair: Nayda G. Santiago-Santiago

Degree: Master of Science

Graduation Date: December 2014