

Intelligent Learning Environments

Research in Computing Science

Series Editorial Board

Comité Editorial de la Serie

Editors-in-Chief:

Editores en Jefe

Juan Humberto Sossa Azuela (Mexico)

Gerhard Ritter (USA)

Jean Serra (France)

Ulises Cortés (Spain)

Associate Editors:

Editores Asociados

Jesús Angulo (France)

Jihad El-Sana (Israel)

Jesús Figueroa (Mexico)

Alexander Gelbukh (Russia)

Ioannis Kakadiaris (USA)

Serguei Levachkine (Russia)

Petros Maragos (Greece)

Julian Padget (UK)

Mateo Valero (Spain)

Editorial Coordination:

Coordinación Editorial

Blanca Miranda Valencia

Research in Computing Science es una publicación trimestral, de circulación internacional, editada por el Centro de Investigación en Computación del IPN, para dar a conocer los avances de investigación científica y desarrollo tecnológico de la comunidad científica internacional. **Volumen 56**, diciembre 2012. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No. : 04-2005-121511550100-102, expedido por el Instituto Nacional de Derecho de Autor. *Certificado de Licitud de Título* No. 12897, *Certificado de licitud de Contenido* No. 10470, expedidos por la Comisión Calificadora de Publicaciones y Revistas Ilustradas. El contenido de los artículos es responsabilidad exclusiva de sus respectivos autores. Queda prohibida la reproducción total o parcial, por cualquier medio, sin el permiso expreso del editor, excepto para uso personal o de estudio haciendo cita explícita en la primera página de cada documento. Impreso en la Ciudad de México, en los Talleres Gráficos del IPN – Dirección de Publicaciones, Tres Guerras 27, Centro Histórico, México, D.F. Distribuida por el Centro de Investigación en Computación, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, México, D.F. Tel. 57 29 60 00, ext. 56571.

Editor Responsable: *Juan Humberto Sossa Azuela, RFC SOAJ560723*

Research in Computing Science is published by the Center for Computing Research of IPN. **Volume 56**, December 2012. Printing 500. The authors are responsible for the contents of their articles. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of Centre for Computing Research. Printed in Mexico City, in the IPN Graphic Workshop – Publication Office.

Volume 56

Volumen 56

Intelligent Learning Environments

Volume editors:

Editores del volumen

María Lucía Barrón Estrada
Ramón Zatarain Cabada
María Yasmín Hernández Pérez

Instituto Politécnico Nacional
Centro de Investigación en
Computación
México 2012



Instituto Politécnico Nacional
"La Técnica al Servicio de la Patria"



ISSN: 1870-4069

Copyright © Instituto Politécnico Nacional 2012

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional “Adolfo López Mateos”, Zacatenco
07738, México D.F., México

<http://www.ipn.mx>
<http://www.cic.ipn.mx>

The editors and the publisher of this journal have made their best effort in preparing this special issue, but make no warranty of any kind, expressed or implied, with regard to the information contained in this volume.

All rights reserved. No part of this publication may be reproduced, stored on a retrieval system or transmitted, in any form or by any means, including electronic, mechanical, photocopying, recording, or otherwise, without prior permission of the Instituto Politécnico Nacional, except for personal or classroom use provided that copies bear the full citation notice provided on the first page of each paper.

Indexed in LATINDEX and Periodica / Indexada en LATINDEX y Periódica

Printing: 500 / Tiraje: 500

Printed in Mexico / Impreso en México

Preface

(Prefacio)

The arrival of the computer to the world, more than fifty years ago, brought two dreams to life: to make computers as intelligent as humans can be, or more, and to use them for providing personalized support to human learning. A great variety of research has come out from these dreams, whose results have demonstrated computers can be very helpful in supporting human learning using Artificial Intelligent (AI) techniques, transforming information into knowledge, using it for tailoring many aspects of the educational process to the particular needs of each actor, and timely providing useful suggestions and recommendations.

The growing usage of computers in education we see today offers an excellent opportunity for exploring new ways of applying AI techniques to education. It also delivers huge amounts of information in need of intelligent management, and poses big challenges to the field on topics such as some enlisted above.

María Lucía Barrón Estrada
Ramón Zatarain Cabada
María Yasmín Hernández Pérez

Table of Contents

(Índice)

Page/Pág.

Intelligent Learning Environments

Design of an Intelligent Agent for Personalization of Moodle's Contents	11
<i>María Lucila Morales Rodríguez, José Apolinar Ramírez Saldivar, Julia Patricia Sánchez Solís, and Arturo Hernández Ramírez</i>	
Emotional Dialogue in a Virtual Tutor for Educational Software	19
<i>María Lucila Morales-Rodríguez, Juan J. González B., Rogelio Florencia-Juárez, and Julia Patricia Sánchez-Solís</i>	
Assessing and Advising on Lexical Richness in an Intelligent Tutoring System	29
<i>Jesús Miguel García Gorrostieta, Samuel González López, and Aurelio López López</i>	
Moveek: a semantic social network.....	37
<i>Pablo Camarillo Ramírez, Abraham Sánchez L., and David Núñez R.</i>	
Possibilistic Safe Beliefs vs. Possibilistic Stable Models	45
<i>Rubén Octavio Vélez Salazar and José Arrazola Ramírez</i>	
The Memory Map Model used for Personalization in Intelligent Learning Environments	53
<i>Carlos Ramírez and Benjamín Valdés</i>	
Design and Implementation of an Affective ITS	61
<i>María Lucía Barrón Estrada, Ramón Zatarain Cabada, Rosario Zatarain Cabada, and Arminda Barrón Estrada</i>	
On a LS-Adaptive Learning Objects Creation Methodology using LOM Metadata ..	69
<i>Aremy Olaya Virrueta Gordillo and Rodolfo Esteban Ibarra Orozco</i>	
A Model for the Representation of Competences Applied to Student's Knowledge Modeling	77
<i>Carlos Ramírez and Erik Sanchez</i>	

Regular Papers

An Analysis of Web Services Attributes for Discovery Support.....	89
<i>Héctor Jiménez Salazar, Christian Sánchez Sánchez, Carlos Rodríguez Lucatero, and Arturo Wulfrano Luna Ramírez</i>	
Analysis of the Quotation Corpus of the Russian Wiktionary	101
<i>Alexander Smirnov, Tatiana Levashova, Alexey Karpov, Irina Kipyatkova, Andrey Ronzhin, Andrew Krizhanovsky, and Nataly Krizhanovsky</i>	
Linguistic Support of a CAPT System for Teaching English Pronunciation to Mexican Spanish Speakers	113
<i>Olga Kolesnikova</i>	
Corpus morfológicamente representativo: preparación de datos y compilación para el español.....	131
<i>Liliana Chanona-Hernández y Alexander Gelbukh</i>	

Intelligent Learning Environments

Design of an Intelligent Agent for Personalization of Moodle Contents

María Lucila Morales-Rodríguez, José Apolinar Ramírez-Saldivar,
Julia Patricia Sánchez-Solís, and Arturo Hernández-Ramírez

División de Estudios de Posgrado e Investigación, Instituto Tecnológico de Ciudad Madero,
Ciudad Madero, Tamaulipas, Mexico

{lmoralesrdz, jpatricia.sanchez}@gmail.com, apolinar_r@yahoo.com,
ahr@prodigy.net.mx

Abstract. This paper presents the architecture of an Intelligent Learning Management System (ILMS) applied to Moodle. Specifically, we present the design and implementation of an agent that select the teaching strategy according to the student's learning style. The chosen teaching strategy is used to filter the learning objects that are displayed to the students.

Keywords. Intelligent Learning Management System, teaching strategy, learning style, personalization content.

1 Introduction

Today, it is clear that the use of information and communication technology in the process of teaching and learning has increased in schools and universities. This has led to the creation of Intelligent Tutoring Systems (ITS). An ITS is a software system that aims to guide students in their learning and/or training process in a personalized way. To achieve its objective, the ITS should model the teaching process, the expert's knowledge, the student understanding on that domain, and achieve communication between these elements. However, most of these systems are not designed to show the material to the students according to their preferences, nor are they designed to diagnose whether the student has developed the competences of the subject that will be taught. One exception is the architecture proposed by [1], which integrates the concept of learning styles, as well as competency-based education in the basic architecture of an ITS.

Also, there are other systems called Learning Management Systems (LMS), which are platforms where teachers can create and manage courses. However, according to [2], these systems don't allow the learner to get a personalized learning experience. Considering this problem, in this paper we apply the ITS architecture proposed by [1] to create an Intelligent Learning Management System (ILMS) applied to the LMS Moodle [3]. This proposal seeks to enhance the features of Moodle and provides a

personalized learning to the student. The design and implementation of an intelligent agent that chooses a teaching strategy according to the student's learning style is presented. This strategy will be used to select the content that will be displayed to the student.

This paper is structured as follows: section 2 describes the concepts related to learning styles. Section 3 presents the proposed architecture. Section 4 describes the design and implementation of the agent. Section 5 presents conclusions.

2 Learning Styles

According to [4], the idea that each person learns differently from others, allows us to find more suitable ways to facilitate their learning. However, we must be careful not to "label" the people, because the learning styles, although relatively stable, can change depending on the situation in which the person is.

In [4], the concept of "learning style" is defined as the fact that each person uses their own methods or strategies to learn. Although the strategies vary depending on what the person wants to learn, each person tends to develop certain preferences or global trends, trends that define their style of learning. Table 1 shows some learning styles that have been considered in the implementation of tutoring systems to adapt their teaching environments to users.

Table 1. Learning styles implemented in ITS.

Models of Learning styles	Works
Felder and Silverman model	Hernández [5], Caviedes [6], Cataldi [7]
Gardner's Multiple Intelligences	Cataldi [7]
VARK of Neil Fleming.	Araújo [8] and Peter [2]

Table 2. Example of Teaching-learning strategies for the VARK model.

Learning Style	Teaching-Learning Strategies
Visual	<ul style="list-style-type: none"> ▪ Pictures, Videos, and/or Posters.
Aural	<ul style="list-style-type: none"> ▪ Discuss topics with your teachers. ▪ Explain new ideas to other people. ▪ Use a tape recorder.
Read / Write	<ul style="list-style-type: none"> ▪ Dictionaries, Textbooks, Notes.
Kinesthetic	<ul style="list-style-type: none"> ▪ Field tours. ▪ Applications. ▪ Trial and error.

In this work the VARK Model [9] was the learning style used and implemented in Moodle. This model is a tool to find out the preferences of persons when they process information. This model takes its name from the initials of the four learning styles that it considers: Visual, Aural, Read/Write and Kinesthetic. Table 2 shows some of the Neil Fleming proposals for teaching-learning strategies for each learning style. The

learning style is detected by a process that applies a questionnaire to determine the student's dominant learning style, this process is described in [10].

3 Architecture

This section presents an Intelligent Learning Management System (ILMS) applied to the LMS Moodle based on the integration of the concept of learning styles and competency-based education in the basic architecture of an ITS (see Figure 1). The concept of learning style is implemented in the process called "selector agent of teaching-learning strategies" and the concept of competencies in the process called "diagnostic of competencies". Both processes were integrated in the tutor module.

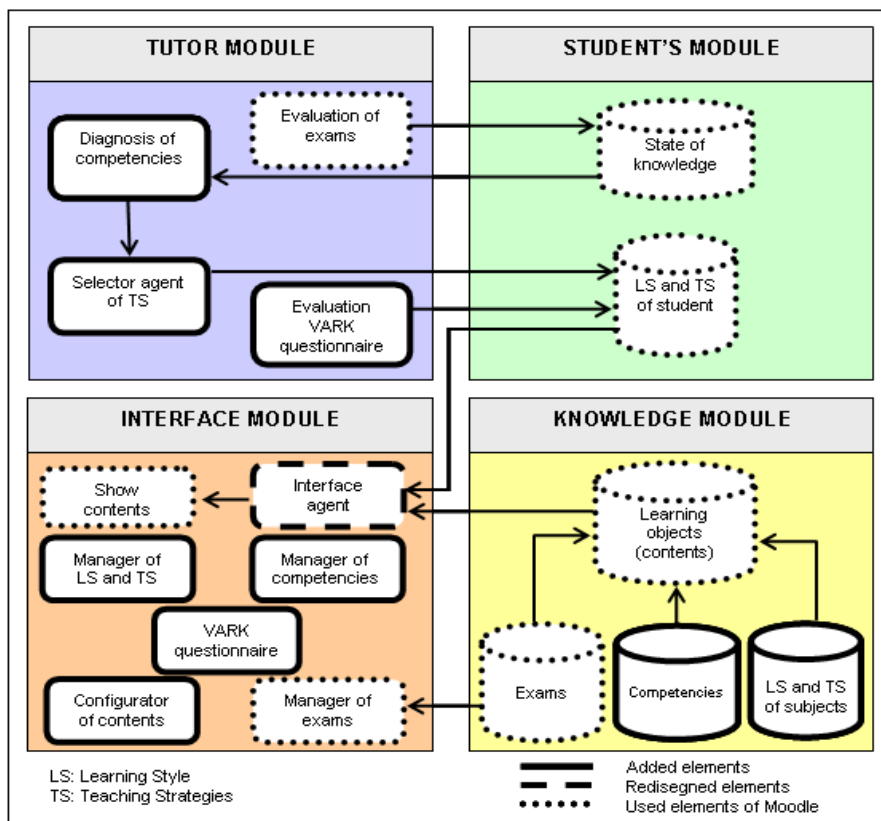


Fig. 1. Architecture of an Intelligent Learning Management System.

The modules of the architecture are related as follows: the *knowledge module* stores the contents to be taught, which are displayed via the *interface module*. The interface module displays the contents designed according to the learning style and the current teaching strategy for the student, this information is consulted in the

student module. The student module is updated by the tutor module, which is responsible for evaluating the performance of the student and diagnosing the competencies that students possess. If the student receives a diagnosis of unsatisfactory competence, the *tutor module* updates the teaching strategy and/or if necessary also the student's learning style, stored in student module, through the selector agent.

The execution of the selector agent depends on the result obtained by the process of diagnosis of competencies. That is, when a student does not meet the standard of competence established, the diagnostic process will execute selector agent. The purpose of the selector agent is to reinforce student learning, by selecting a teaching strategy that goes according to the current learning style of the student. When the teaching strategy is chosen, the interface agent uses this strategy to filter the learning objects developed for the current learning style and teaching strategy of the student that will be shown. This paper only describes the design and implementation of the selector agent.

4 Design and Implementation of the Selector Agent in Moodle

The selector agent is responsible for selecting the teaching strategy according to the current learning style of the student; it is used to filter the learning objects that will be displayed. When all teaching strategies related to the current learning style of the student have been used, the selector agent will be able to change their current style.

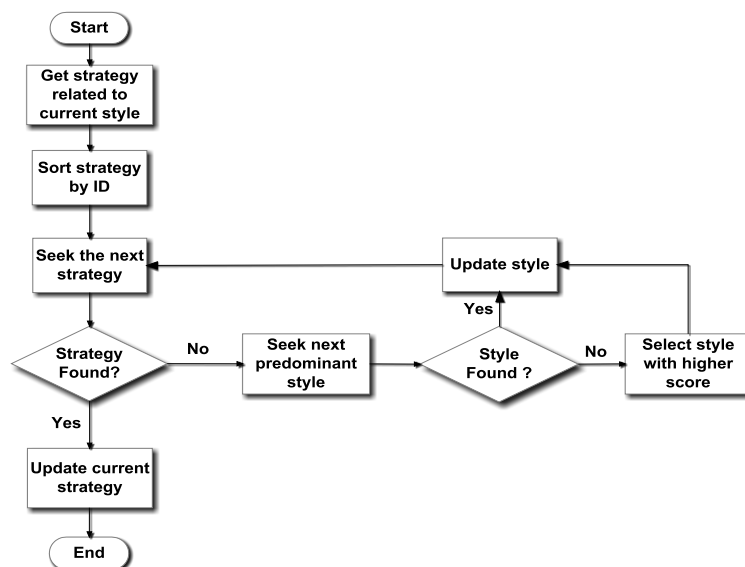


Fig. 2. Flowchart of the selector agent.

This change is performed according to the scores obtained by the student in the VARK questionnaire, choosing the next predominant learning style. When this action is performed on different occasions, and all learning styles have been used, then the selector agent will assign the style with the highest score as current learning style. After changing the current student's learning style, the agent updates the current teaching strategy (see Figure 2).

Below the implementation of the selection agent in Moodle is presented:

1. When the selector agent is executed, the first step is to perform a query to the database to get all teaching strategies that are related to the current learning style of the student. The current learning style of the student is loaded into memory in the object *\$USER->Estiloactual*. When the query is performing, the strategies are ordered according to their identifier (see Figure 3).

```
$rs_Estrategia = get_recordset_sql("SELECT Distinct msee.EstrategiaId
                                  FROM mdl_Scorms_ee msee, mdl_Estrategias me
                                  WHERE (msee.EstrategiaId = me.Id) and
                                  msee.CursoId = $Curso and
                                  msee.EstiloId = $USER->Estiloactual
                                  ORDER BY msee.EstrategiaId");
```

Fig. 3. Query to get teaching strategies.

2. Once the strategies associated with student's current style have been obtained, the strategy that has an ID greater than the ID of the current strategy is chosen. The current strategy ID is loaded into memory in the object *\$USER->EstrategiaActual*. When a strategy satisfies this condition, it is set as the current strategy (see Figure 4).

```
while ($row = rs_fetch_next_record($rs_Estrategia)) {
    if ($USER->EstrategiaActual < $row->EstrategiaId) {
        $estrategia_id = $row->EstrategiaId;
        $Encontrado = 1;
        break;
    }
}
rs_close($rs_Estrategia);
```

Fig. 4. Code to search the new teaching strategy.

3. When a strategy that satisfies the condition is not found, this means that all strategies have been used. So, the selector agent will be able to change the current learning style of the student, according to the scores obtained by the student in the

VARK questionnaire, choosing the learning style that has the following lower score compared with the current style score.

4. Subsequent to the selection of the new style, we proceed to select the ID of the first strategy associated with the new style, which must be associated with some materials (see Figure 5). This strategy will be set as the current.

```
$estrategia_id = get_field_sql("SELECT Distinct msee.EstrategiaId  
FROM mdl_Scorms_ee msee, mdl_Estrategias me  
WHERE (msee.EstrategiaId = me.Id) and  
msee.CursorId = $Curso and  
msee.EstiloId = $estilo_id  
ORDER BY msee.EstrategiaId");
```

Fig. 5. Code to select a teaching strategy related with the new learning style.

5. After the selector agent made the change of learning style and/or teaching strategy, this information is updated in three different places: 1) in the *EstiloActual* and *EstrategiaActual* fields contained in the *mdl_user_info_data* table, 2) in the *mdl_historical_agenteselector* table which stores the history of changes realized by the selector agent in the style and strategy and 3) in the *\$USER* object.

With these three updates, the selector agent function ends. Now, the next process is filtering the contents showed to the student through the interface agent, considering that contents are designed for the current style and strategy for the student.

5 Conclusions

This paper describes the modifications made to basic architecture of an ITS incorporating the concept of learning styles and competency-based education. This architecture is applied to the LMS Moodle in order to create an Intelligent Learning Management System (ILMS).

In particular, we describe the design and implementation of the processing performed by the selector agent to choose a teaching strategy according to the student's learning style. Supporting it with the idea expressed in [3], which states that students learn more effectively when they are taught according to their learning style.

References

1. Morales-Rodríguez, M. L., Ramírez-Saldivar, J. A., Hernández-Ramírez, A., Sánchez-Solís, J. P., Martínez-Flores, J. A.: Agente Selector de Estrategias de Enseñanza-Aprendizaje para la Educación Basada en Competencias. In: 17th International Congress on Computer Science Research (CIICC'11), pp. 53–63. Morelia, Mich. (2011)
2. Peter, S. E., Bacon, E., Dastbaz, M.: Learning styles, personalization and adaptable e-learning. In: Fourteenth International Conference on Software Process Improvement

3. Research, Education and Training, INSPIRE 2009, pp. 77-87, The British Computer Society, Swindon, UK (2009)
4. Dougiamas, M.: Moodle.org: open-source community-based tools for learning, <http://moodle.org/>.
5. Gómez Navas Chapa, L: Manual de estilos de aprendizaje: material autoinstruccional para docentes y orientadores educativos. Secretaría de Educación Pública (2004)
6. Hernández, Y., Rodríguez, G., Arroyo, F. G.: Integrating learning styles and affective behavior into an intelligent environment for learning. Workshop on Intelligent Learning Environments in MICAI 2010, November 8-13, Pachuca, Hgo. México. (2010)
7. Caviedes, P. D., Medina, G. V., García, P. O.: Diseño de un sistema tutor inteligente basado en estilos cognitivos. In: V Simposio Internacional de Bibliotecas Digitales. ISTECS, October 27-28, Albuquerque (2009)
8. Cataldi, Z., Lage, F.: Modelado del Estudiante en Sistemas Tutores Inteligentes, TE&ET Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología, 5, 29-38 (2010)
9. Gomes. A., Miranda, R., Vale. Z., Faria L.: A Web-Based Intelligent Tutoring System on Teaching and Learning Electrical Project. In: International Conference of Education, Research and Innovation, ICERI2010, pp. 5288–5294. IATED Press, Madrid (2010)
10. Fleming, N.: VARK - A Guide to Learning Styles, <http://www.vark-learn.com/english/index.asp>
11. Morales-Rodríguez, M. L., Ramírez-Saldivar, J. A., Hernández-Ramírez, A., Sánchez-Solís, J. P., Martínez-Flores, J. A.: Architecture for an Intelligent Tutoring System that considers learning styles. In: Workshop on Intelligent Learning Environments, November 28-29, Puebla, México (2011)

Emotional Dialogue in a Virtual Tutor for Educational Software

María Lucila Morales-Rodríguez, Juan J. González B., Rogelio Florencia-Juárez,
and Julia Patricia Sánchez-Solís

División de Estudios de Posgrado e Investigación, Instituto Tecnológico de Ciudad Madero,
Ciudad Madero, Tamaulipas, México

{lmoralesrdz, jpatricia.sanchez}@gmail.com,
jjgonzalezbarbosa@hotmail.com, rogelio.florencia@live.com.mx

Abstract. In this paper, an educational software used to enhance the skills of children in the area of math is presented. This software has a tutor who guides the child in different activities. The tutor exhibits personality traits and emotions in its dialogue to create a sense of immersion in the student and catch his attention towards the game. We adapted an Emotional Extension of the Artificial Intelligence Markup Language, structure here named as EE-AIML.

Keywords. Educational software, intelligent conversational agent, AIML, personality, emotions.

1 Introduction

Based on the latest assessment of education in Mexico [1], an increase in the level of achievement of students between 3rd and 6th grade was found in the subject of Mathematics. However, the results show that 63% of the elementary school students obtained an insufficient/elemental level of proficiency and only 37% of students obtained a Good/Excellent level of proficiency. At the secondary school level, these values become more critical, reaching percentages of 84.2% and 15.8% respectively. We thought that the use of fun educational games that reinforce learning could be a new way to increase the interest on learning math.

This paper presents educational software to reinforce the learning of mathematics, which incorporates an Emotional Embodied Conversational Agent (E-ECA) with the role of tutor. The virtual tutor's aim is to encourage, repress, correct and help the student during the different activities. The tutor has a cognitive module, implemented using a Bayesian network, for modeling a student's performance. Based on the performance of the student, the tutor sets the difficulty level of the arithmetic problems that the student must solve. The tutor's aim is to create a sense of immersion in the student and thus, catch his attention on the game, incorporating personality traits and emotions in the tutor dialogues. For this, we adapted in the dialogue module an architecture for incorporating personality traits and emotions in written dialogues in natural language [2]. This architecture is an Emotional Extension made to the

AIML language (Artificial Intelligence Markup Language), which we here refer to as EE-AIML (Emotional Extension –AIML).

This paper is structured as follows: Section two presents related works and some definitions. Section three describes the overview of the EE-AIML architecture. Section four describes characteristics related to the implementation of immersion. Section five describes the adaptation of the EE-AIML architecture into the tutor dialog module, and finally, section six presents the conclusions and future work.

2 Educational Software and Emotional Interaction

In the field of math-oriented educational software, there are works such as MatheMax [3], MatheMax Pro [4] and Matris [5], which were designed for assisting students in basic math operations, starting with exercises like counting, simple addition, subtraction, multiplication and division as well as mixed modes for different skill levels. The feedback of these systems takes the form of a congratulatory graphic and sound. There are some educational software that integrate a character with the role of presenters, some examples are the works Mathematics with Pipo [6] and Mathematics with Mario 2 [7], which contains several different math games and exercises with different learning objectives ranging from counting, drawing with numbers, simple operations (logical sequences) to complex operations such as sorting, measuring, weighing, handling coins, etc..

Usually, the works that integrate virtual characters provide a slight feedback about student performance and can become static and repetitive, because they are not adapted to the context of the situation. We think that the way to include a more realistic interaction is to have dialogues or phrases consistent to the context, for example, correcting the student when a problem is really detected or congratulating him when he has performed well. Another problem, which we have identified, is that the traditional educational software does not integrate the affective aspect in the interaction with the student.

In this paper these problems are addressed by adapting the EE-AIML architecture to the dialog module of the virtual tutor. The EE-AIML architecture uses the AIML language as dialogue manager. This architecture incorporates personality traits and emotions in the dialogues of a virtual character. AIML language is an XML specification, which is a scripting language that defines a database of question-answer useful for programming chat robots [8].

Personality is an inherent people characteristic that largely influences the thoughts, feelings and human behavior. It is also the feature that makes a person different of another [9, 10]. Over time, several theories about personality that follow a psychological approach have been developed. [10] classifies them into: a) psychoanalytic theory, b) theories of traits, c) behavioral theories, d) biological theories, e) social learning theories and f) social cognitive theories.

The dialogues, besides exhibiting personality, also convey emotions. Some of the emotional theories that have influenced computer science researchers are Appraisal theories, Dimensional theories, Anatomical theories, and Rational theories. Among

the existing emotional theories, the Appraisal theory has had a major impact on the design of virtual agents. This theory is the most used in the implementation of computational models [11].

A work that integrates personalities and emotional models with the AIML Standard is the Huang et al work [14], in which they propose a framework called GECA, to facilitate communication between the different modules, such as sensor inputs from the human users, inference engine, emotion model, personality model, dialogue manager, face and body animation, etc. To control the agent's behavior, conversational markers were incorporated into standard AIML tags to specify parameters for nonverbal inputs and outputs.

Others works have made extensions to AIML language to incorporate emotional characteristics. Tee Conie [15], Sumedha Kshirsagar [16], and Baldassarri [17] have incorporated new tags in the AIML language in order to control aspects such as facial expression and response selection basing these processes on an emotional state.

In our research, the emotional aspect of the virtual character is controlled by the EE-AIML architecture, which is based on the behavioral model of Morales-Rodríguez [12]. This behavioral model is a combination of appraisal and dimensional theories of emotions, integrated through the model of personality called Five Factor Model [13].

3 Emotional Dialogue Manager in a Virtual Tutor

The virtual tutor proposed is an Emotional Embodied Conversational Agent (E-ECA). It is composed of two interconnected modules, a cognitive-emotional module and a dialogue manager.

The cognitive-emotional module performs cognitive evaluations based on the context of the conversation and updates the emotional state according to the E-ECA personality. The dialogue manager is an Emotional Extension of AIML (EE-AIML), which has the function of selecting the response phrases of virtual tutor.

The dialogue manager contains a knowledge base that comprises the E-ECA dialogues, which are categorized in *conversational contexts*. This categorization is used to identify the conversational topic of the user inputs. The structure of the standard tags of AIML are <category>, <pattern> and <template>. The AIML interpreter seeks the user input that matches the terms defined by the <pattern> tag, thus, the output speech act (delimited by <template> tag) is consistent with the input [14] (see Fig. 1).

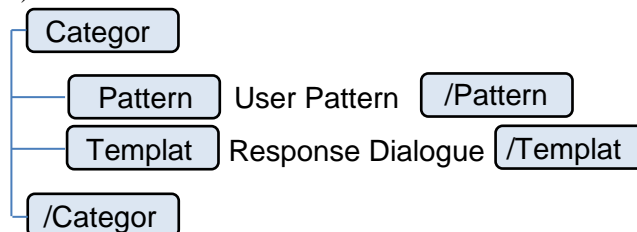


Fig. 1. Tags structure in AIML language.

The new extended structure of AIML allows the implementation of different personalities for the E-ECA. Each personality has a defined set of emotions and different ranges of emotional intensity. Different speech acts are associated to each range of emotional intensity, which respond to user input pattern. The new structure is presented in Figure 2.

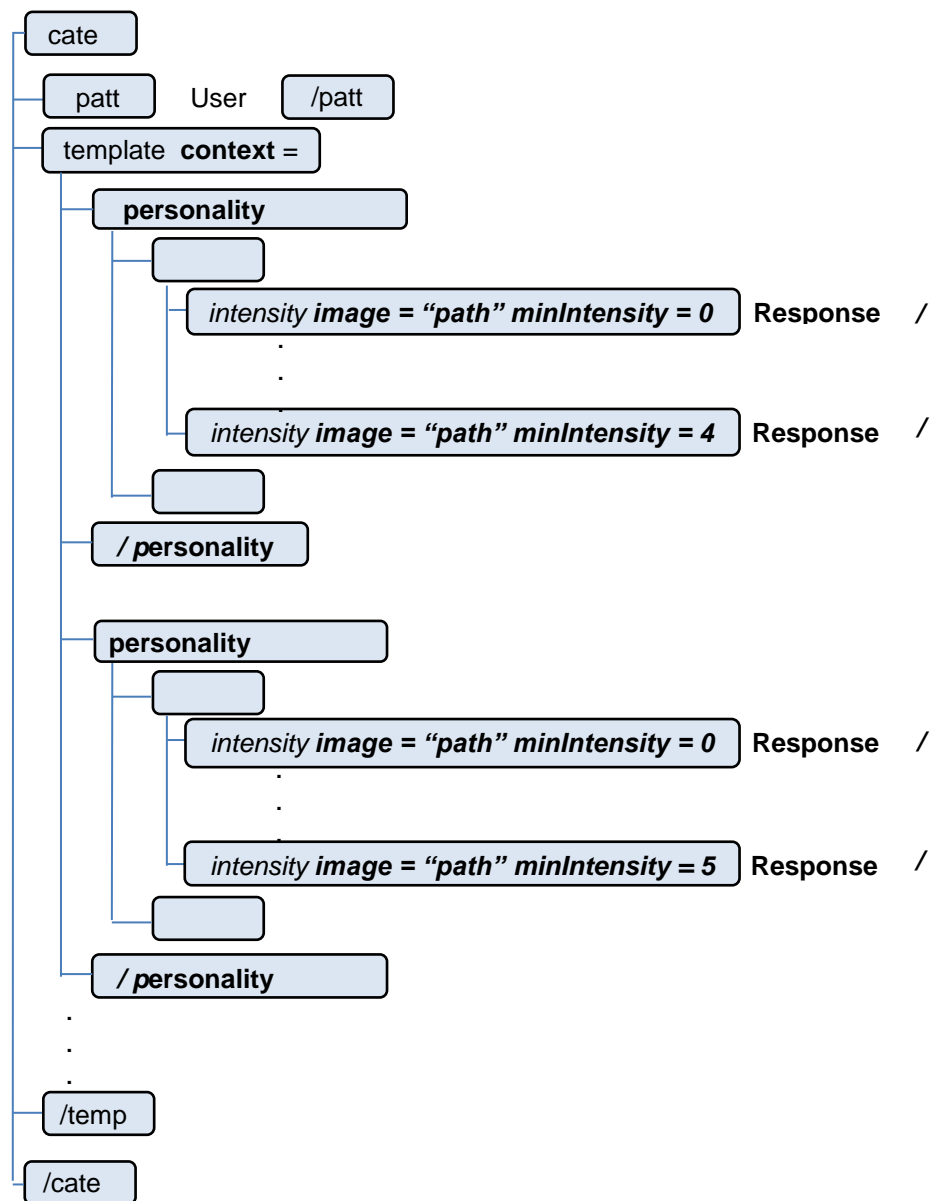


Fig. 2. Structure of the EE-AIML architecture

4 Implementation of Immersion

In order to increase the student's interest in the game, get their attention and reinforce their knowledge, the software tries to develop a sense of immersion in the student, allowing him to choose the character with whom he felt most identified.

The software has two characters, a male and a female teacher with different personalities expressed by their interaction dialogues. The images of the tutors are presented in Figure 3.

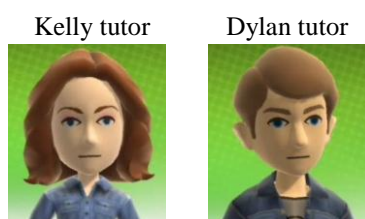


Fig. 3. Images of the virtual tutors

The exercises are presented dynamically, with the values of the responses varying randomly and appearing in random positions on the screen. If after 20 seconds, the student has not chosen a correct answer, the answer options change their position on the screen. This is in order to induce the student to respond during the shortest possible time. An example of the problem and answer options is presented in Figure 4:



Fig. 4. Example of exercise interface and their answers options

The natural language dialogues used are characterized based on the personality and emotions defined for each tutor using the EE-AIML architecture. This allows different dialogues and the virtual tutors can exhibit a different personality and emotional behavior, endowing the interaction of greater dynamism and naturalness.

5 Integration of the EE-AIML Architecture to the Dialog Module of the Tutor

In this software the student interacts using the mouse to select their answer for problems shown by the tutor, there is no keyboard input available.

As described in [2], the EE-AIML architecture is designed to interact with users through dialogues written in natural language, which are formalized in a knowledge base of the AIML language. Therefore, in order to adapt EE-AIML architecture to the dialog module of the tutor, we identified phrases that represent possible actions raised during the interaction by the student. For example, if a student chooses the correct answer to an addition problem, the pattern for knowledge base of AIML can be defined as "the sum is correct". This is necessary to retrieve from the knowledge base, the context and feedback (response phrases) appropriate to the action selected during interaction. The phrases recovered from the knowledge base are filtered according to the personality and emotional state of the tutor.

The following code shows an example of a knowledge base for modeling the input pattern "THE SUM IS CORRECT" to the personalities of the tutors called Dylan and Kelly, in particular, are presented the responses associated with the emotion of joy.

```
<category>
<pattern> THE SUM IS CORRECT </pattern>
<template context = "PROBLEM_OK">
  <personality value = "Kelly">
    <joy>
      <intensity image = "Kelly_Joy1.jpg" intensityMin = "0"
        intensityMax = "2.5">
        <Random>
          <li>It was OK</li>
          <li>Your answer was correct</li>
        </Random>
      </intensity>
      .
      .
      .
      <intensity image = "Kelly_Joy4.jpg" intensityMin = "7.5"
        intensityMax = "9">
        <Random>
          <li>Excellent, Congratulation, continues well !!!</li>
          <li> Very well, nobody does it better than you!!</li>
        </Random>
      </intensity>
    </joy>
  </personality>
  <personality value = "Dylan">
    <joy>
```



```

<intensity image = "Dylan_Joy1.jpg" intensityMin = "0"
      intensityMax = "2.5">
  <Random>
    <li>Good!</li>
    <li>Excellent</li>
  </Random>
</intensity>
.
.
.
<intensity image = "Dylan_Joy4.jpg" intensityMin = "7.5"
      intensityMax = "9">
  <Random>
    <li>Excellent, congrats you did great!!!</li>
    <li>Cool, nobody does it better than you!!</li>
  </Random>
</intensity>
</joy>
.
.
.
</personality>
</template>
</category>

```

Figure 5 presents examples of possible phrases and images that can be selected according to the intensity of joy that the tutor Kelly could experience.





Intensity	Image	Phrases
$0 \leq \text{Intensity} < 2.5$		<i>It was OK</i> <i>Your answer was correct</i>
$2.5 \leq \text{Intensity} < 5$		<i>You did very well</i> <i>Right, keep it up!</i>
$5 \leq \text{Intensity} < 7.5$		<i>Excellent, continue well and learn a lot!</i> <i>Right, continue well and go for more!</i>
$7.5 \leq \text{Intensity} < 10$		<i>Excellent, congratulations continue well!</i> <i>Very Well, nobody does it better than you!!</i>

Fig. 5. Phrases and images that Kelly can select according to the intensity of the Joy emotion

6 Conclusions and Future Work

In this paper we present the integration of emotional dialogue in a virtual tutor for educational software that reinforces the learning of students in the area of math.

The virtual tutor expresses personality traits and emotion through their dialogue aiming to develop a sense of immersion in the student in order to get his attention and thus enhance their knowledge through the different exercises.

The EE-AIML architecture [2] was adapted as a tutor dialogue module to give the student feedback on his performance.

The EE-AIML architecture is an AIML core extension that integrates personalities and emotional tags incorporating the behavioral model of Morales-Rodríguez [12], which is a combination of Appraisal and Dimensional theories of emotions integrated through the Five Factor Model [13] of personality.

As future work we are interested in conducting an analysis to determine factors or elements that appeal to the students for improving the user interface, and to add activities associated with the student's learning style. Related to the virtual tutor, some of future works will endow the tutor with the ability to handle written dialog entries to generate conversations and to improve the non-verbal expression that it currently has.

References

1. Resultados de Prueba Enlace 2011 Básica y Media Superior, http://www.enlace.sep.gob.mx/content/gr/docs/2011/ENLACE2011_versionFinalSEP.pdf
2. Morales-Rodríguez, M. L., González Barbosa, J. J., Florencia Juárez, R., Fraire Huacuja, H. J. y J. A. Martínez Flores: Emotional Conversational Agents in Clinical Psychology and Psychiatry, *Advances in Artificial Intelligence*, LNCS 6437, pp. 458-466 (2010)
3. Behling, J. : MatheMax - practice counting and arithmetics (ABC-Ware, Shareware and Freeware for Kids), <http://www.abc-ware.com/mathe.htm>.
4. edpr1011 - MatheMax Pro, <http://edpr1011.wikispaces.com/MatheMax+Pro>.
5. MaTris - practice arithmetics with Tetris game (ABC-Ware, Shareware and Freeware for Kids), <http://www.abc-ware.com/matris.htm>.
6. Matemáticas con PIPO, <http://www.pipoclub.com/espanol/pipo4/home.htm>.
7. Zona de Alumnos de la Junta de Castilla y León (zonaalumnos), <http://www.educa.jcyl.es/educacyl/cm/zonaalumnos>.
8. Wallace, R. S.: *The elements of AIML Style*. ALICE A. I. Foundation, (2003)
9. Vinayagamoorthy, V., Gillies, M., Steed, A., Tanguy, E., Pan, X., Loscos, C., Slater, M.: Building expression into virtual characters. In: *Eurographics Conference State of the Art Reports*, September 4-8, Vienna (2006)
10. He, X.: *An Affective Personality for an Embodied Conversational Agent*. M. Sc. Thesis, Curtin University of Technology, Department of Computer Engineering (2007)
11. Marsella, S., Gratch, J., Petta, P.: *Computational Models of Emotion. Blueprint for Affective Computing, a sourcebook and manual*. In: Scherer, K.R., Bänziger, T., Roesch, E. (eds.), pp 21-41 Oxford University Press (2010)
12. Morales-Rodríguez, M. L.: *Modèle d'interaction sociale pour des agents conversationnels animés. Application à la rééducation de patients cérébro-lésés*, PhD Thesis, Toulouse, Université Paul Sabatier, 108 p. (2007)

13. McCrae, R. R., John, O. P.: An Introduction to the Five-Factor Model and Its Applications, *Journal of Personality*, 60, pp. 175–215 (1992)
14. Huang, H.-H., Cerekovic, A., Tarasenko, K., Levacic, V., Zoric, G., Pandzic, I. S., Nakano, Y., Nishida, T.: Integrating embodied conversational agent components with a generic framework, *Multiagent Grid Syst.*, vol. 4, no. 4, pp. 371–386 (2008).
15. Connie, T., Sing, G. O., Michael, G.K.O., Huat, K.L.: A Computational Approach to Emotion Recognition in Intelligent Agent. In: *The Asian Technology Conference in Mathematics (ATCM 2002)*, Melaka (2002)
16. Kshirsagar, S. Magnenat-Thalmann, N.: A Multilayer Personality Model. In: *SMARTGRAPH'02 Proceedings of the 2nd International Symposium on Smart Graphics*, pp. 107–115, ACM, New York (2002)
17. Baldassarri, S., Cerezo, E., Anaya, D.: Interacción emocional con actores virtuales a través de lenguaje natural. In: *VIII Congreso Internacional de Interacción Persona-Ordenador*, pp. 343–352, Zaragoza (2007)

Assessing and Advising on Lexical Richness in an Intelligent Tutoring System

Jesús Miguel García Gorrostieta¹, Samuel González López²,
and Aurelio López-López²

¹ Universidad de la Sierra, Moctezuma, Sonora,
Mexico

² Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla,
Mexico

jmgarcia@unisierra.edu.mx, {sgonzalez, allopez}@inaoep.mx

Abstract. Guiding students on writing is a hard and time consuming chore for advisors, since it requires several iterations before achieving an acceptable level. Normally, when professors advise students close to graduation, most questions are about the structure of the thesis project. Issues such as the correct wording or abuse of certain terms within a title, problem statement, objectives and justification become the main tasks of the instructor. In this paper, we present a web-based intelligent tutoring system (ITS) to provide student advice in structuring research projects. We propose a student model based on a network to follow the progress of each student in the development of the project and personalized feedback on each assessment. This tutor includes a module for assessing the lexical richness, which is done in terms of lexical density, lexical variety, and sophistication. We also establish the methodology for future testing with undergraduate students.

Keywords: E-learning, natural language processing, intelligent tutoring system, lexical richness.

1 Introduction

Guiding and instructing a student on research or thesis writing is a hard and time consuming chore for advisors, since requires several iterations before achieving an acceptable level. There is a need to alleviate the burden of this task, possibly by technologies such as tutoring systems.

An intelligent tutoring system (ITS) is a system that provides personalized instruction or feedback to students without much involvement of instructors. Recent advances in ITS include the use of natural language technologies to analyze student writing and provide feedback as presented in the article by McNamara [1]. Writing Pal (WPal) is an ITS that offers strategy instruction, practice, and feedback for developing writers. There are also intelligent virtual agents able to answer questions for the student related to an academic subject [2]. A dialogue-based ITS called Guru was proposed in [3], which has an animated tutor agent engaging the student in a collaborative conversation that references a hypermedia workspace, displaying and animating images significant to the conversation. Another dialogue-based ITS Auto

Tutor uses dialogues as the main learning activity [4]. All these ITS use Natural Language to interact with the student similarly to the ITS we present in this paper.

Normally when advising students close to graduation, most questions are about the structure of the thesis or research project. Issues such as the correct wording or abuse of certain terms within a title, problem statement, objectives and justification become the main task of the instructor. In this paper, we present a web-based intelligent tutoring system (ITS) to provide student advice in structuring research projects. We propose a student model based on a network to follow the progress of each student in the development of the project and personalized feedback on each assessment. This tutor includes a module for assessing the lexical richness, which is done in terms of lexical density, lexical variety, and sophistication.

There are a variety of methods to evaluate the use of vocabulary (lexicon) in text. One of them is to measure the sophistication of some papers using text word lists. In [5], they used a list of 3000 easy words. For Spanish, some studies use the list provided by the SRA (Spanish Royal Academy) of 1000, 5000 and 15000 most frequent words. Other works have used Yule's K to measure the richness in texts [6], where this kind of measures focus on the word repetitions and this is considered a measure of lexical variety.

We also establish the methodology for future testing with undergraduate students. We cannot ignore the importance of language, especially in writing, when considering the formation of higher education students; one of these stages of formation is related to the generation and application of knowledge through research, which are usually placed in the last semesters of their programs of study.

Each institution adopts various mechanisms that allow students to enter in the field of research, either through business internship, professional practice or in the various forms of professional qualification that presents the possibility of doing a research. However, the process of drafting the research projects is usually not an easy task for students. Therefore, our proposed system intends to assist the work of the instructor and to facilitate and guide students through this process.

The paper is organized as follows. Section 2 describes the model underlying the tutoring systems, while section 3 details the implementation with examples of draft evaluations. We conclude in section 4, discussing further works.

2 The Model

The intelligent tutor in the Domain Module presents material concerning the different elements of the project, such as the problem statement, title, objectives and justification. For each element, a test is applied to validate the reading of materials and practical exercises are applied using the richness Lexical Analyzer to achieve a high level of density, diversity and sophistication in the student text productions. The results of the test and lexical analysis are sent to the Student Progress Module to update the knowledge state of the student in a network. Figure 1 shows the intelligent tutor model.

The Student Progress Module (SPM) records the student's progress in the network which is depicted in Figure 2, when the student completes the test, the value of the

test node element is updated and the SPM calculates the student's progress for the parent node using the weights assigned to each question in the test [7].

Similarly as when performing the exercises with the lexical analyzer, the corresponding node in the network is updated and the SPM estimates the student's progress for the parent node using the weights assigned to the lexical density, variety and sophistication in the Lexical Analyzer.

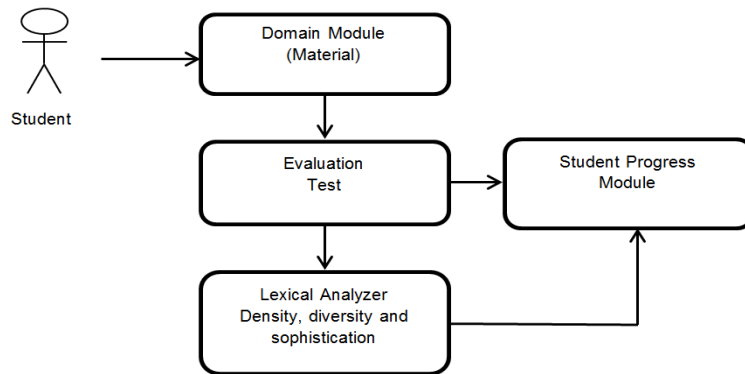


Fig. 1. Model of Intelligent Tutoring System.

Figure 2 illustrates the weights assigned to each node according to the experience of the teacher. For instance, in the Test node of the Statement, a weight of 50% of the parent node problem statement is assigned, which includes 5 questions to verify that the student has read the material. Once the student has correctly answered questions, a 50% of advance in the concept is assigned, as shown in Figure 3. This will enable the student to use the lexical analyzer to perform three exercises which have a combined weight of 50% of the parent node, which is distributed as follows: 20% to lexical density, 20% to lexical diversity, and finally 10% for lexical sophistication.

So, by completing the exercise of lexical density with a high grade, the student would have advanced 70% in the concept, as shown in Figure 6. Also when the student gets a high grade on the exercise of lexical diversity, he will have completed 90% of the problem statement concept, leaving only the exercise of lexical sophistication to complete the 100% of the concept.

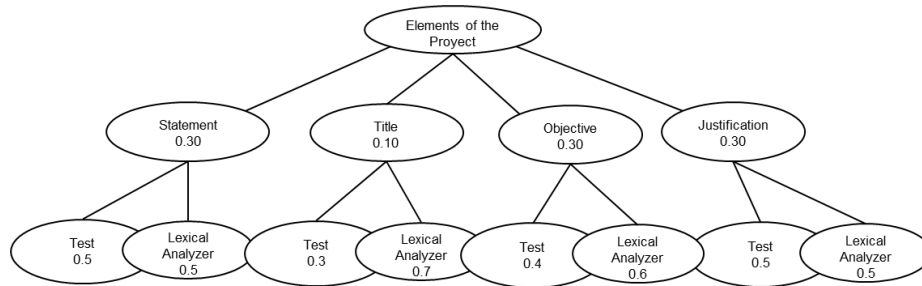


Fig. 2. Network used in the Student Model.

Lexical analysis focuses on the evaluation of three measures: lexical density, lexical variety and sophistication, which together assess lexical richness. The first measure, lexical variety, seeks to measure student ability to write their ideas with a diverse vocabulary. This feature is computed by dividing the unique lexical types (Tlex) by the total of lexical types (Nlex). Tlex refers to the unique terms of content, while Nlex represents total terms of content, both ignoring empty words [8].

The lexical density aims to reflect the proportion of content words in the complete text. This measure is calculated by dividing the unique lexical types or content words (Tlex) by the total words of evaluated text (N), i.e. the number of words before removing stop words.

The third measure is sophistication, which attempts to reveal the knowledge of technical concepts and is the proportion of "sophisticated" words employed. This measure is computed as the percentage of words out of a list of 1000 common words, provided by the SRA. All the measures take values between 0 and 1, where 1 indicates a high lexical value, and values close to zero mean a low value of the lexicon of the evaluated section.

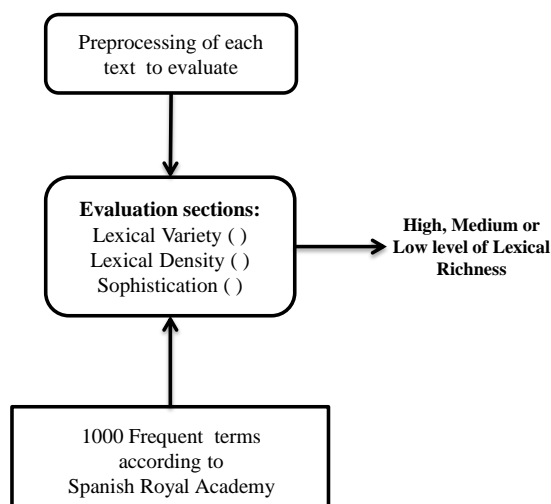


Fig. 3. Model of Lexical Analyzer.

The preprocessing of the text was filtering and removing empty words from a list provided by the module of NLTK-Snowball. Stop words include prepositions, conjunctions, articles, and pronouns. After this step, only content words remained, which allowed the calculation of the three measures. Finally, the results produced by the Lexical Analyzer are sent to the Student Progress Module, so the intelligent tutor manages the results achieved by the student.

A scale ranging in High, Medium and Low in lexical richness has been established based on our previous work [9], where we analyzed research proposals and theses of graduate and undergraduate students.

3 The Intelligent Tutoring System

The intelligent tutoring system is developed in PHP for easy access via web and the network structure is stored in a MySQL database, the lexical analyzer is developed in Python because of the easy access to processing tools of natural language. The analyzer uses the open source tool FreeLing¹ for stemming words and then analyzes the density, diversity and sophistication in the text.

Figure 4 shows the graphical interface of the tutoring system in which we observe the button to the main menu to access the elements of the project (in Spanish *Elementos del proyecto*) inside we find links to access the problem statement, title, objectives and justification. For each element, there are three sections: material, test and practical evaluation. In this figure, we can also notice the progress section (in Spanish *Avance*) in the left side, reporting the progress in the concept. As we can see, to enter the practical evaluation, the student must first successfully complete the test receiving a 50% advance in the concept and 15% in the complete project.

Fig. 4. Lexical Analyzer for Density (in Spanish).

The section of practical evaluation is also depicted in figure 4, where the student writes his problem statement to be analyzed for lexical density. First, the analyzer performs a tokenization of words, then a classification based on the 1000 most common words of Spanish are done to identify stop words and the rest as content words. Density analysis measures the balance between content words and stop words,

¹ This software is available at <http://nlp.lsi.upc.edu/>.

if the text has too many stop words it will have a very low density, if the text has just a few stop words compared to content words it will have a high density.

As we can see in Figure 5 the feedback of the lexical density analysis and the level assigned to the problem statement proposed by the student is "Low Density" (in Spanish *Densidad Baja*) due to the large number of stop words relative to content words, the system sends a message to the student with a feedback according to the level assigned.

The message displayed is "we suggest reviewing of the text, there are few content words, try to reduce the terms outlined in red" (in Spanish *Se sugiere revisar el texto, ya que existen pocas palabras de contenido, procura reducir los términos subrayados en rojo*) in the paragraph, we observe stop words underlined to indicate to the student that it is necessary try to reduce them, and a progress bar is presented to indicate the progress of his writing graphically, in this case a 50.98% of advance.

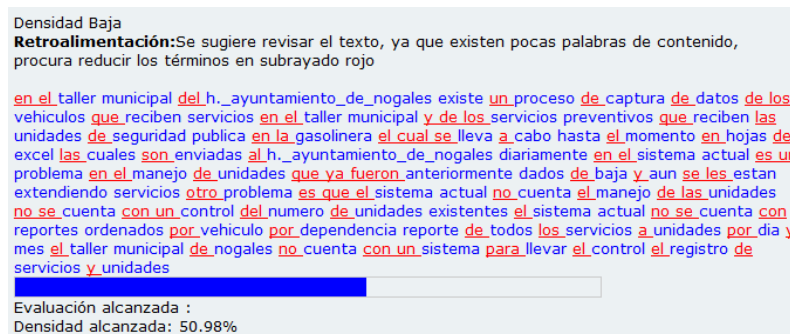


Fig. 5. Detailed feedback of Lexical Analyzer for Density (in Spanish).

After correcting the paragraph, the analyzer indicates a high level (in Spanish *Densidad Alta*) and activates the access link to the analysis of lexical diversity (in Spanish *Análisis de diversidad léxica*), as shown in Figure 6, the feedback indicates that the statement problem is in balance between stop and content words, with a 66.67% of lexical density.

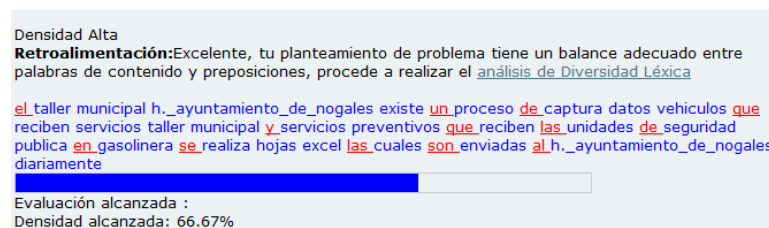


Fig. 6. Detailed feedback from Lexical Analyzer, for High Density in text (in Spanish).

Figure 7 shows the lexical analyzer for diversity which are content words that are repeated several times such as "services" (in Spanish *Servicios*) and "units" (in Spanish *unidades*). This case has a medium level of diversity with a feedback to the student "There are still repetitive words of content, modify your text, avoid using the

same word several times, try using synonyms for such word" (in Spanish *Aún existe repetición de palabras de contenido, modifica tú texto evitando usar varias veces la misma palabra, procura usar sinónimos de dicha palabra*) with a 62.16% of progress in diversity, that is graphically illustrated by the progress bar at the bottom of the figure.



Fig. 7. Lexical Analyzer for Diversity (in Spanish).

Upon completion of the exercise of lexical diversity, the student can access the exercise of sophistication which measures the degree to which the student uses uncommon words, hopefully specialized to the domain of computer science.

Once completed the three lexical analyses, the student can move on to the next item of the project and the teacher can review a more refined statement of the problem.

4 Conclusion and Future Works

The use of intelligent tutoring system for research project drafts aims to support teachers in reviewing research projects providing material to the student, by tracking their progress and lexically analyzing the drafting of their writings.

In future work, we intend to use the ITS with college students who start with their research and observe the performance for future changes in the system. The experiment will use a control group and an experimental group to watch the progress in the two groups regarding to the recommendations of the tutor in their lexical richness. This pilot implementation will seek to measure if the student has been concerned with reaching only a medium level of lexical analysis, or if he is looking to

reach the highest level to improve his writing skill. Also we will adapt the interface of the ITS to have an improved use on mobile devices.

We also plan to extend the ITS assessing additional aspects in drafts such as coherence and specific language usage for particular sections of proposals.

Acknowledgments. The second author was supported by Conacyt scholarship 1124002, while the third author was partially supported by SNI.

References

1. McNamara, D. S., Raine, R., Roscoe, R., Crossley, S., Jackson, G. T., Dai, J., Cai, Z., Renner, A., Brandon, R., Weston, J., Dempsey, K., Carney, D., Sullivan, S., Kim, L., Rus, V., Floyd, R., McCarthy, P. M., & Graesser, A. C.: The Writing-Pal: Natural language algorithms to support intelligent tutoring on writing strategies. In: Applied natural language processing and content analysis: Identification, investigation, and resolution. Hershey, PA: IGI Global. pp. 298-311 (2012)
2. Rospide, C.G. & Puente, C.: Virtual Agent Oriented to e-learning Processes. In: Proceedings of 2012 International Conference on Artificial Intelligence. Las Vegas, Nevada (2012)
3. Olney, A.; D'Mello, S. K.; Person, N. K.; Cade, W. L.; Hays, P.; Williams, C.; Lehman, B. & Graesser, A. C.: Guru: A Computer Tutor That Models Expert Human Tutors. In: Stefano A. Cerri; William J. Clancey; Giorgos Papadourakis & Kitty Panourgia (eds.) 'ITS', Springer, pp. 256-261 (2012)
4. Graesser, A.C., D'Mello, S.K., Craig, S.D., Witherspoon, A., Sullins, J., McDaniel, B., and Gholson, B.: The Relationship between Affective States and Dialog Patterns during Interactions with Autotutor, *J. Interactive Learning Research*, vol. 19(2), pp. 293-312 (2008)
5. Schwarm, S. and Ostendorf, M.: Reading level assessment using support vector machines and statistical language models. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL '05), pp. 523-530, (2005)
6. Miranda, A., and Calle, J.: Yule's Characteristic K Revisited. *Language Resources and Evaluation*, 39(4), pp. 287-294 (2005).
7. Sucar, L.E., and Noguez, J.: Student Modeling. In: O. Pourret, P. Naim, B. Marcot (eds.), *Bayesian belief networks: a practical guide to applications*, Wiley, pp.173-186 (2008)
8. Roberto, J., Martí, M., and Salamó, M.: Análisis de la riqueza léxica en el contexto de la clasificación de atributos demográficos latentes. *Procesamiento de Lenguaje Natural*, 48, pp. 97-104 (2012)
9. González, L. S. and López-López, A.: Supporting the review of student proposal drafts in information technologies. In: Proceedings of the 13th annual conference on Information technology education (SIGITE '12). ACM, New York, pp. 215-220 (2012)

Moveek: A Semantic Social Network

Pablo Camarillo R., Abraham Sánchez L., and David Núñez R.

Benemérita Universidad Autónoma de Puebla, Puebla,
Mexico

{pablo.camarillo, asanchez, davidn}@cs.buap.mx

Abstract. We know that it is difficult to create semantic-web content because pages must be semantically annotated through processes that are mostly manual and require a high degree of engineering skill. We must therefore devise means for transforming existing, non-semantic social networks into semantic social networks. We propose using information extraction ontologies to handle this challenge. In this work we show how a successful ontology-based data-extraction technique can automatically generate semantic annotations for social networks. We have implemented a prototype of our approach to demonstrate that our proposal works.

Keywords. Social network, semantic annotation, ontologies.

1 Introduction

The web is now a major medium of communication in our society and, as a consequence, an element of our socialization. As the web is becoming more and more social, we are now collecting huge amount of knowledge on-line [1]. Semantic web researchers provide models to capture such activities that have to be fully exploited in order to be turned into collective intelligence.

The “semantic web” represents a major advance in web utility, but it is currently difficult to create semantic-web content because pages must be semantically annotated through processes that are mostly manual and require a high degree of engineering skill. Semantic-web proponents propose making web content machine understandable through the use of ontologies, which are commonly shared, explicitly defined, generic conceptualizations [2]. But then one of the immediate problems we face is how to deal with current web pages. There are billions of pages on the current web, and it is impractical to ask web developers to rewrite their pages according to some new, semantic-web standard, especially if this would require tedious manual labeling of documents.

Web semantic annotation research attempts to resolve this problem. The goal of web semantic annotation is to add comments to web content so that it becomes machine understandable [3]. Unlike an annotation in the normal sense, which is an unrestricted note, a semantic annotation must be explicit, formal, and unambiguous: explicit makes a semantic annotation publicly accessible, formal makes a semantic

annotation publicly agreeable, and unambiguous makes a semantic annotation publicly identifiable. These three properties enable machine understanding, and annotating with respect to ontology makes this possible.

In this work, we present an approach to semantically relate contents posted on a social network. A semantics based on the representation of knowledge through ontology of domain will be used. The domain covering by this ontology is a domain of scientist concepts mentioned in the publications made in the social network. Our work aims to create a platform based on the composition of Web services able to provide the functionality of a social network and at the same time provide various features of the semantic web, such as the semantic annotation and semantic queries to the information published on this network.

Our semantic social network Moveek is briefly presented in Section 2 and the process to semantically relate the contents. In Section 3 we discuss some experimental results and finally in Section 4 we give conclusions and present future work.

2 A Semantic Social Network

The term semantic social network was coined independently by Stephen Downes and Marco Neumann in 2004 to describe the application of semantic Web technologies and online social networks [4].

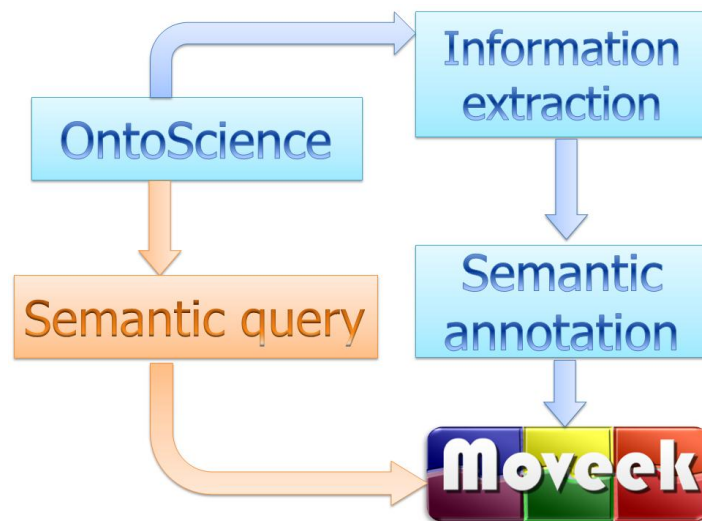


Fig. 1. The overall functionality of Moveek.

The main motivation of this work was the curiosity produced by the strange behavior of the many social networks like Facebook, Twitter, YouTube, etc., i.e., at the time to suggest new friends, products or services. The most important function of our social network is located in the invocation of some Web services. Such Web services will perform the tasks of semantic annotation, information extraction,

semantic query and social interaction with the social network, Moveek. Figure 1 shows our proposal to produce the semantic core of our social network.

In semantic Web applications, ontologies describe formal semantics for applications, and thus make information sharable and machine-understandable. The work of semantic annotation is, however, more than just knowledge representation. Semantic annotation applications must also establish mappings between ontology concepts and data instances within documents so that these data instances become sharable and machine-understandable.

The term Ontology has been used in several disciplines, from philosophy, to knowledge engineering, where ontology is comprised of concepts, concept properties, relationships between concepts and constraints. Ontologies are defined independently from the actual data and reflect a common understanding of the semantics of the domain of discourse. Ontology is an explicit specification of a representational vocabulary for a domain; definitions of classes, relations, functions, constraints and other objects.

The ontology-based knowledge representation needs a robust ontology. We named our ontology OntoScience, since we want to cover a scientific domain in this work. We develop our ontology according to methodology proposed by Rubén Dario Alvarado [5]. The steps of this methodology are shown in Figure 2.

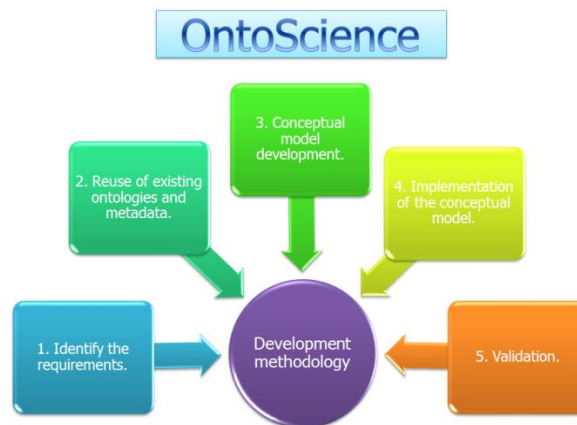


Fig. 2. The development process for the OntoScience.

First, we delimited the domain of our ontology and we set the questions that the ontology is able to answer. Second, we checked many existing ontologies and metadata like WikiOnt¹, Ontology of SCIENCE² and the Dublin Core Metadata Third³, we started to collect the concepts and metadata that we needed and we built the conceptual model with the classes and relationships necessary for our proposal. After of developing the conceptual model, we implemented the ontology with the

¹ <http://sw.deri.org/2005/04/wikipedia/wikiont.html>

² http://protege.stanford.edu/ontologies/ontologyOfScience/ontology_of_science.htm

³ <http://dublincore.org>

Protègè system [6]. Finally we validated the ontology structure using java routines that they programmatically verified the taxonomy and consistence of OntoScience. Information extraction for building the bridge between the modeled concepts in OntoScience and the posts published in our social network, we need to know what terms have a relationship with the concepts in OntoScience. To do this, we developed an information extraction tool that is published as a Web service. This tool receives the post as a string and use an application produced by the GATE environment [7]. The GATE application returns the information of the post as a XML document. This document contains the terms that are mentioned in the post and that are modeled in OntoScience.

The document also contains the ID of the class that coincides with the term mentioned in the post. Figure 3 shows how the information extraction task is performed.



Fig. 3. Information extraction flow.

Pragmatically, queries and assertions are exchanged among software entities using the vocabulary defined by a common ontology. Ontologies are not limited to conservative definitions, which in the traditional logic sense only introduce terminology and do not add any knowledge about the world. To specify a conceptualization we need to state axioms that put constraints on the possible interpretations for the defined terms.

The difficulty in sharing and processing Web content, or resources, derives at least in part from the fact by using the resources are unstructured, and consist of text, video etc. The semantic annotation process is performed by using the extracted XML document; this document is retrieved from the information extraction Web service. Figure 4 illustrates the steps for inserting the semantic annotation in the post.

First, we receive the XML document with the information of the post and programmatically we rebuild this post with an HTML link that, after that, it can be used to formulate semantic queries. This HTML link contains a string that uses the class ID extracted as a GET variable for the PHP engine that will performs the semantic query. In the Figure 5, one can see an example of an annotated post.

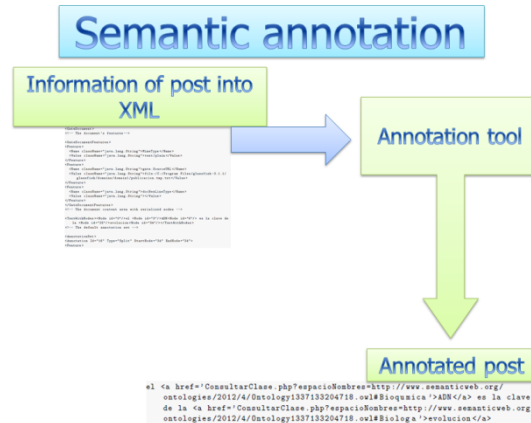


Fig. 4. The proposed semantic annotation solution.

```

e1 <a
href='ConsultarClase.php?espacioNombres=http://www.semanticweb.org/ontologies/2012/4/Ontology1337133204718.owl#Bioquímica'>ADN</a>
es la clave de la evolucion

```

Fig. 5. An example of an annotated post.

Domain ontology provides the standardized terminology and conceptualization of a particular domain. We suppose that certain domain ontology is agreed and used to reconcile the semantics of model contents. The ontological concepts are referenced by model contents through simple URIs or semantic relationships.

3 Experimental Results

Moveek was developed using PHP server-side scripts and AJAX client-side scripts. Our Web services are developed as follows: social interaction Web services were developed with C# and .NET framework 3.5 technologies for a quickly development; semantic Web services, information extraction information Web services and semantic queries Web service were developed with Java technologies since we used the API of GATE and Protégé. We should note that one of the advantages of Web services is the interoperability between many development environments such as our architecture.

The reason for using ontology is because we need answers to many questions about the social network behavior. These answers are solved by using the relationships modeled in the ontology. Following the previous strategy to solve the semantic issues, we used a Web service to execute the semantic queries. At this point, the Moveek users have many annotated terms and related with their corresponding ontology class. The Figure 6 shows the path that follows a post to send the request of a query and

how the response of the semantic Web service is used to show the list of post(s) related with the term originally selected.

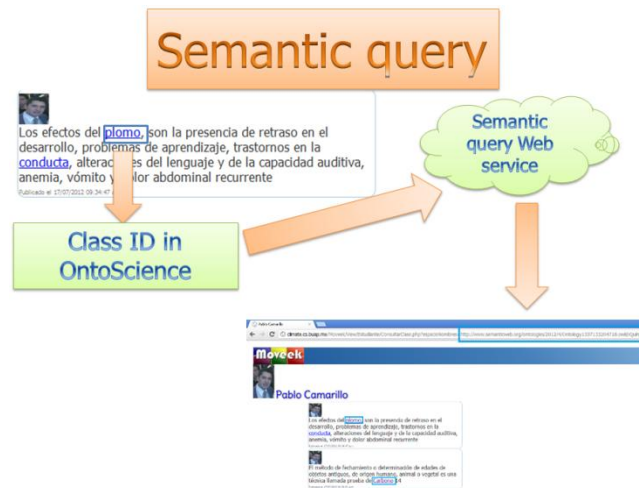


Fig. 6. Example of our semantic query strategy.

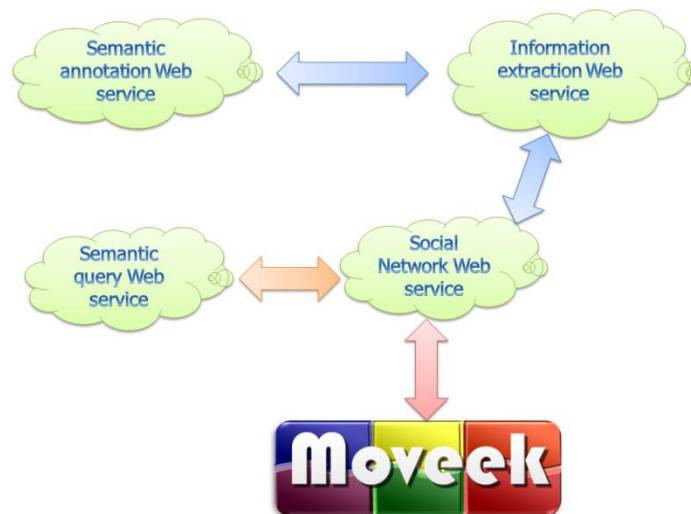


Fig. 7. Combination of the Web services.

For discovering the knowledge, our proposal is completed when we use all Web services mentioned previously. Every Web service has a key role in our architecture. In Figure 7, one can see how the architecture combines the Web services and its relationships with the Moveek social network.

Web services are used to distribute services over the Internet. They make operations of applications available or enable information systems to be invoked over

the network. There are two ways to combine such services: either through orchestration or choreography. In orchestration, the involved web services are under control of a single endpoint central process (another web service). Choreography, in contrast, does not depend on a central orchestrator. Each Web service that participates in the choreography has to know exactly when to become active and with whom to interoperate. In our proposed architecture, we used orchestration for composing Web services.

4 Conclusions and Future Work

We have developed an interesting social network called Moveek and its objective is to create networks of users who share and disseminate publications with some scientist contents. Each publication, automatically, will highlight the concepts that the system has detected as scientists contents in the ontology. In this way, the user can consult more publications semantically related with the highlighted concepts [5].

The study and development of the Semantic Web applications includes various areas ranging from correct modeling of a Web application to the development of a 'good' representation of knowledge and the relationship that should be both ideas to make the Web more human. In this work, we involve all these aspects concerning the semantic Web with the objective to understand, but in fact to develop an application that would reflect the advantages (and disadvantages) of the semantic Web. In the research and in the development of our social network, we have realized that, while the semantic Web meets its mission of making the information queries more efficient on the content of the Web, it also entails certain disadvantages, especially concerning the necessary infrastructure to not affect the performance of the Web applications.

Another interesting aspect that we highlight of this work is the growing presence of multimedia resources on the Web, this makes more challenging the development of the semantic Web. With the latest idea, we can introduce the future work of this work. First, we must improve the social network modeling in order to scale the application and think of more features such as the management of courses to create networks of scientific collaboration on our social network. Another crucial point in our future work is our intention to automate the process of learning of the ontology and thus enrich the scientific domain which it currently covers.

References

1. Mika, P: Social networks and semantic Web (Semantic Web and beyond). Springer-Verlag, (2007)
2. Gruber, T.R.: Translation approach to portable ontology specifications. Knowledge Acquisition, 5(2) 199–200 (1993)
3. Breslin, J.G., Passant, A., and Decker, S.: The social semantic Web. Springer-Verlag (2009)
4. Downes, S.: Semantic networks and social networks. The Learning Organization, 12(5) 411–417 (2005)

Pablo Camarillo Ramírez, Abraham Sánchez L., David Núñez R.

5. Camarillo R., P.: Descubriendo conocimiento en redes sociales a partir de ontologías. BS Thesis, FCC-BUAP (in Spanish) (2012)
6. The Protégé ontology editor and knowledge acquisition system, <http://protege.stanford.edu/>
7. General architecture for text engineering (GATE), <http://gate.ac.uk/>

Possibilistic Safe Beliefs vs. Possibilistic Stable Models

Ruben Octavio Velez Salazar, Jose Arrazola Ramirez, and Ivan Martinez Ruiz

Benemérita Universidad Autónoma de Puebla, Mexico

ruvelsa@yahoo.com

Abstract. In this paper we show an application of possibilistic stable models to a learning situation. Our main result is that possibilistic stable models of possibilistic normal programs are also possibilistic safe beliefs of such programs. In any learning process, the learners arrive with their previous knowledge. In most cases, it is incomplete or it comes with some degree of uncertainty. Possibilistic Logic was developed as an approach to automated reasoning from uncertain or prioritized incomplete information. The standard possibilistic expressions are classical logic formulas associated with weights. Logic Programming is a very important tool in Artificial Intelligence. Safe beliefs were introduced to study properties and notions of answer sets and Logic Programming from a more general point of view. The stable model semantics is a declarative semantics for logic programs with default negation. In [1], the authors present possibilistic safe beliefs. In [2], the authors introduce possibilistic stable models.

Keywords. Normal logic programs, safe beliefs, possibilistic logic, possibilistic normal logic programs, possibilistic safe beliefs.

1 Introduction

In the mid 80's Dubois and Prade [3] introduced Possibilistic Logic, a logic initially based on classical logic, useful for modeling problems where incomplete or partially contradictory information exists. It deals with uncertainty in the following way: in order to express the extent to which the available evidence entails the truth of a formula which is associated to a number between 0 and 1 called its degree of necessity (or its certainty). If we wish to express the extent to which the truth of the formula is not incompatible with the available evidence we may use a degree of possibility. In this paper we will refer only to the degree of necessity of the formula φ , which is denoted by $n(\varphi)$.

Answer Set Programming is a form of declarative programming based on the stable semantics of logic programming. The definition of answer sets for augmented programs (which are a general type of programs) is based on finding minimal models of some reduced logic programs. Stable model semantics [4] is an answer set semantics for logic programs with default negation.

In [2], the authors use possibility theory to extend the non-monotonic semantics of stable models for logic programs with default negation. They define a clear semantics

for such programs by introducing possibilistic stable models, taking into account a certainty level associated with each piece of knowledge.

Any logic whose set of provable formulas lies between intuitionistic and classical logic (inclusive) is known as an Intermediate logic. These logics are able to distinguish between a and $\neg\neg a$, a property which makes these logics suitable to characterize notions of logic programming. Pearce [5] established a link between Answer Set Programming and Intermediate Logics. The authors in [6], present an extension of answer sets, called safe beliefs, which they define based on intuitionistic logic and following ideas found in [5]. Their definition formalizes the idea that non monotonic inference can be achieved determining some formulas that one can *safely believe*.

In [1], the authors develop possibilistic safe beliefs in order to broaden the scope of applications. They present a characterization of possibilistic safe beliefs in terms of possibilistic intuitionistic logic.

2 Background

In this section we first introduce the syntax of logic formulas considered in this paper. Then we present a few basic definitions of how logics can be built to interpret the meaning of such formulas in order to finally give a brief introduction to the logics that are relevant for the results of our later sections.

2.1 Syntax of Formulas

We consider a formal (propositional) language built from: an enumerable set L_0 of elements called *atoms* (denoted a, b, c, \dots); the binary connectives \wedge (*conjunction*), \vee (*disjunction*) and \rightarrow (*implication*); and the unary connective \neg (*default negation*). Formulas (denoted $\phi, \psi, \gamma, \dots$) are constructed as usual by combining these basic connectives together.

We also use $\phi \leftrightarrow \psi$ to abbreviate $(\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$ and, following the tradition in logic programming, $\phi \leftarrow \psi$ as an alternate way of writing $\psi \rightarrow \phi$. A *theory* is just a set of formulas and, in this paper, we only consider finite theories. Moreover, if T is a theory, we use the notation L_T to stand for the set of atoms that occur in the theory T .

2.2 Logic Systems

We consider a *logic* simply as a set of formulas that satisfies the following two properties: (i) is closed under modus ponens (i.e. if ϕ and $\phi \rightarrow \psi$ are in the logic, then also ψ is) and (ii) is closed under substitution (i.e. if a formula ϕ is in the logic, then any other formula obtained by replacing all occurrences of an atom a in ϕ with another formula ψ is still in the logic). The elements of a logic are called *theorems* and the notation $\vdash_X \phi$ is used to state that the formula ϕ is a theorem of the logic X

(i.e. $\varphi \in X$). We say that a logic X is *weaker than or equal to* a logic Y if $X \subseteq Y$, similarly we say that X is *stronger than or equal to* Y if $Y \subseteq X$.

Hilbert style proof systems. There are many different approaches that have been used to specify the meaning of logic formulas or, in other words, to define *logics* [7]. In Hilbert style proof systems, also known as axiomatic systems, a logic is specified by giving a set of axioms (which is usually assumed to be closed by substitution). This set of axioms specifies, so to speak, the “kernel” of the logic. The actual logic is obtained when this “kernel” is closed with respect to the inference rule of modus ponens.

The notation $\vdash_X \varphi$ for provability of a logic formula φ in the logic X is usually extended within Hilbert style systems, given a theory T , using $T \vdash_X \varphi$ to denote the fact that the formula φ can be derived from the axioms of the logic and the formulas contained in T by a sequence of applications of modus ponens.

2.3 Intuitionistic Logic

In this subsection we will briefly introduce the intuitionistic logic that will be relevant for our purposes in this paper. We will present a Hilbert style definition for it. We start from a basic logic called *Positive Logic*, to which we add some axioms in order to obtain Intuitionistic Logic.

Definition 1. *Positive Logic is defined by the following set of axioms:*

- Pos 1:* $\varphi \rightarrow (\psi \rightarrow \varphi)$
- Pos 2:* $(\varphi \rightarrow (\psi \rightarrow \gamma)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \gamma))$
- Pos 3:* $\varphi \wedge \psi \rightarrow \varphi$
- Pos 4:* $\varphi \wedge \psi \rightarrow \psi$
- Pos 5:* $\varphi \rightarrow (\psi \rightarrow (\varphi \wedge \psi))$
- Pos 6:* $\varphi \rightarrow (\varphi \vee \psi)$
- Pos 7:* $\psi \rightarrow (\varphi \vee \psi)$
- Pos 8:* $(\varphi \rightarrow \gamma) \rightarrow ((\psi \rightarrow \gamma) \rightarrow (\varphi \vee \psi \rightarrow \gamma))$

Definition 2. *Intuitionistic Logic I is defined as Positive Logic plus the following two axioms:*

- Int1:* $(\varphi \rightarrow \psi) \rightarrow [(\varphi \rightarrow \neg \psi) \rightarrow \neg \varphi]$
- Int2:* $\neg \varphi \rightarrow (\varphi \rightarrow \psi)$

2.4 Normal Logic Programs

In this paper, when a non empty set of atoms that determines the language of the programs is not given explicitly, we will consider it to be the set of atoms occurring in the program, so if P denotes the program then L_P denotes the set of atoms under consideration. A normal logic program is a finite set of formulas, called rules, of the form $r = c \leftarrow a_1, \dots, a_n, \neg b_1, \dots, \neg b_m$, where $n \geq 0$, $m \geq 0$, and $\{c, a_1, \dots, a_n, b_1, \dots, b_m\} \subseteq L_P$.

For such a rule r we use the following notations: $body^+(r) = \{a_1, \dots, a_n\}$, $body^-(r) = \{b_1, \dots, b_m\}$, $head(r) = c$, and $r^+ = head(r) \leftarrow body^+(r)$.

2.5 Possibilistic Logic

The standard possibilistic expressions of possibilistic logic are classical logic formulas associated with a parameter, interpreted as lower bounds of necessity degrees.

A *necessity-valued* formula is a pair $(\varphi \alpha)$, where φ is a propositional formula in some given logic and $\alpha \in (0, 1]$. $(\varphi \alpha)$ expresses that φ is certain to the extent α , that is, $N(\varphi) \geq \alpha$, where N is a necessity measure which models our state of knowledge. The constant α is known as the *valuation* of the formula or its *weight* and is represented as $val(\varphi)$.

To define a possibilistic logic axiomatically, we start with a logic X (in section 4, X will be the Intuitionistic logic, which we denote by I) and define an axiom system by the set of axioms $\{(\varphi 1) : \varphi \text{ is an axiom of } X\}$ with the following rules of inference:

(GMP) $(\varphi \alpha), (\varphi \rightarrow \psi \beta) \vdash_{PXL} (\psi \min\{\alpha, \beta\})$.

(S) $(\varphi \alpha) \vdash_{PXL} (\varphi \beta)$ if $\alpha \geq \beta$.

This defines the possibilistic logic PXL . Let us point out that if $(\varphi \alpha) \vdash_{PXL} (\varphi \beta)$, then $\alpha \geq \beta$, so in the case $\beta > \alpha$ will not be considered. The reasoning behind (GMP) is the principle that the strength of a conclusion is the strength of the weakest argument used in its proof. Observe that every axiom and every theorem are associated with the value 1.

The following lemma can be found in [8] and we will use it in our main result.

Lemma 1. Let Γ be a set of formulas in PXL and $\varphi \in X$. Then $\Gamma \vdash_{PXL} (\varphi 1)$ if and only if $\Gamma^* \vdash_X \varphi$, where Γ^* is the collection of formulas in Γ without the corresponding parameters.

3 Possibilistic Stable Semantics

The stable model semantics was proposed in [4] for logic programs with default negation, i.e., normal logic programs. In order to deal with a reasoning which is non-monotonic and uncertain, the authors in [2] presented possibilistic stable models for possibilistic normal programs. We reproduce some of the main results here.

3.1 Possibilistic Definite Logic Programs

A *possibilistic definite (logic) program* is a set of possibilistic rules of the form

$$r = (c \leftarrow a_1, \dots, a_n \alpha),$$

where $n \geq 0$, $\{c, a_1, \dots, a_n\} \subseteq L_P$, and $\alpha \in (0, 1]$. The *classical projection* of the possibilistic rule is $r^* = c \leftarrow a_1, \dots, a_n$. The *weight* α is less than or equal to $n(r)$, the necessity degree representing the certainty level of the information described by r . If R is a set of possibilistic rules, then $R^* = \{r^* : r \in R\}$ is the definite logic program obtained by ignoring all the weights. By $A \not\models r^*$, we denote that r^* is not a logical consequence of the set of formulas A .

Definition 3.[2] *Let P be a possibilistic definite program.*

- *If M denotes the least Herbrand model of the definite program P^* , then the **necessity measure** of an atom $x \in L_P$ is*

$$N_P(x) = \min_{A \subseteq M} \{ \max_{r \in P} \{ n(r) : A \not\models r^* \} : x \notin A \}.$$

- *The set $\{(x, N_P(x)) : x \in L_P, N_P(x) > 0\}$ is **the possibilistic model** of P .*

$N_P(x)$ evaluates the level at which x is inferred from P . Moreover, $N_P(x) = 0$ if and only if x does not belong to the least Herbrand model of the definite program P^* , hence the definition of the possibilistic model of P .

3.2 Possibilistic Normal Logic Programs

Normal Logic Programs allow default negation, as opposed to Definite Logic Programs, in which all the information described is positive. A *possibilistic normal (logic) program* is a finite set of rules of the form

$$r = (c \leftarrow a_1, \dots, a_n, \neg b_1, \dots, \neg b_m) \alpha,$$

where $n \geq 0$, $m \geq 0$, $\{c, a_1, \dots, a_n, b_1, \dots, b_m\} \subseteq L_P$, and $\alpha \in (0, 1]$.

In [4], the authors define the stable model semantics for normal logic programs in terms of a program reduction. This reduction is extended naturally to the possibilistic case as follows.

Definition 4. [2] *Let P be a possibilistic normal program.*

- *Let $A \subseteq L_{P^*}$. The **possibilistic reduct** of P with respect to A , which we denote by P^A , is the set*

$$\{((r^*)^+ \ n(r)) : r \in P, \text{body}^-(r) \cap A = \emptyset\}.$$

- *Let $M \subseteq L_P$. M is a **possibilistic stable model** of P if M is the possibilistic model of P^{M^*} .*

With the following lemma, the authors in [2] show that there is a one-to-one mapping between the possibilistic stable models of a possibilistic normal logic program P and the stable models of its projection P^* . We will use this fact in our main result.

Lemma 2. *Let P be a possibilistic normal program and $M \subseteq L_P$. If M is a possibilistic stable model of P then M^* is a stable model of P^* .*

We will also use this result, found in [5].

Lemma 3. *Let P be a logic program, $M \subseteq L_P$ and X be an intermediate logic. M is a stable model of P if and only if $P \cup \neg \tilde{M} \vdash_X M$.*

In the next section, we will show that the possibilistic stable models of this possibilistic normal program are also its possibilistic safe beliefs.

4 Possibilistic Safe Beliefs

In [9], the authors present *safe beliefs* as an extension of answer sets in terms of completions of a program. In [1], the authors extend this notion to the possibilistic case. We reproduce some of their findings in this section, in which a *possibilistic theory* is a finite set of possibilistic formulas.

Definition 5. [1] *Let Γ be a possibilistic theory.*

- The **inconsistency degree** of Γ , denoted as $Incon(\Gamma)$, is defined as the following number:

$$\max\{\alpha : \Gamma \Vdash_{PIL} (\perp \ \alpha)\}.$$

- Γ is **consistent** if $Incon(\Gamma)=0$.

We note that we need to extend the domain of α from $(0,1]$ to $[0,1]$. If M is a set of possibilistic atoms, we write \tilde{M}^* to denote the complement of M^* in L_{Γ^*} . Also, $\neg\neg M^*$ denotes the set $\{\neg\neg x : x \in M^*\}$ and $\Gamma \Vdash_{PIL} (\varphi \ \alpha)$ denotes that Γ is consistent and $\Gamma \vdash_{PIL} (\varphi \ \alpha)$.

If M is any subset of L_{Γ} we denote by $(\neg \tilde{M}^* \ 1)$ and $(\neg\neg M^* \ 1)$ the sets $\{(x \ 1) : x \in \neg \tilde{M}^*\}$ and $\{(x \ 1) : x \in \neg\neg M^*\}$, respectively.

It is possible to define a partial order in 2^L_{Γ} by defining for every M_1 and M_2 in 2^L_{Γ} , that $M_1 \leq M_2$ if the following two conditions hold:

- a) $M_1 \subseteq M_2$;
- b) If $(\varphi \ \alpha_1) \in M_1$ then there exists $(\varphi \ \alpha_2) \in M_2$ such that $\alpha_2 \leq \alpha_1$.

Definition 6. [1] *Let Γ be a possibilistic theory and M a subset of L_{Γ} . We define M to be a **possibilistic safe belief** of P if the following conditions are met:*

- M is \leq -minimal;
- For every $(a \ \alpha) \in M$, $\Gamma \cup (\neg \tilde{M}^* \ 1) \cup (\neg\neg \tilde{M}^* \ 1) \Vdash_{PIL} (a \ \alpha)$.

The following lemma gives us the characterization we will use in our main result.

Lemma 4. [10] *Let P be a possibilistic normal program and $M \subseteq L_P$. M is a possibilistic safe belief of P if and only if $P \cup (\neg \tilde{M}^* \ 1) \Vdash_{PIL} M$.*

5 Contribution

Our main result follows from the previous lemmas.

Theorem 1. *Let P be a possibilistic normal program and $M \subseteq L_P$. If M is a possibilistic stable model of P then M is a possibilistic safe belief of P .*

Proof. If M is a possibilistic stable model of P then, by lemma 2, M^* is a stable model of P^* , which is equivalent, by lemma 3, to the fact that $P^* \cup \neg \tilde{M}^* \vdash_I M^*$. Now, by lemma 1, we have $P \cup (\neg \tilde{M}^* 1) \vdash_{PIL} M$, and therefore, by lemma 4, M is a possibilistic safe belief of P .

□ The converse of theorem 1 does not hold: it is not difficult to verify that if P is the possibilistic normal program defined by the possibilistic rules $(a \leftarrow \neg b \ 0.5)$ and $(b \leftarrow \neg a \ 0.5)$, and if $M = \{(a \ 0.3)\}$ then M is a possibilistic safe belief of P , but not a possibilistic stable model of P .

6 Our Result and Learning Environments

We start this section with an example derived from [2]. Suppose a certain teacher has a student who has a hard time focusing on more than one subject. The teacher wishes to give the student a Math assignment and a History assignment, for each of which the student has some previous knowledge. The problem is that the student can only focus on Math or on History, but not both. We can represent this situation with a normal logic program

$$\{a \leftarrow b \wedge \neg c, c \leftarrow d \wedge \neg a, e \leftarrow a \wedge b, f \leftarrow c \wedge d, b \leftarrow, d \leftarrow\},$$

where the atoms a and c represent, respectively, the fact that the student uses his previous knowledge in Math and History; b and d represent, respectively, the fact that the student must complete the Math and History assignment; and e and f represent the fact that the student completes his assignment described by b and d , respectively. This normal program has two stable models, $\{a, b, d, e\}$ and $\{b, c, d, f\}$. Each one of these two stable models represents an option for the teacher, who now wishes to evaluate the *certainty* of these two options. In order to do so, the teacher uses her expertise, experience, etc. to determine degrees of certainty of each rule in the program. Now, the task is to figure out how the certainty of the rules in the program affect the certainty of the option described in each stable model.

After determining the degrees of certainty of each rule in the normal program

$$\{a \leftarrow b \wedge \neg c, c \leftarrow d \wedge \neg a, e \leftarrow a \wedge b, f \leftarrow c \wedge d, b \leftarrow, d \leftarrow\},$$

the teacher comes up with the possibilistic normal program

$$P = \{(a \leftarrow b \wedge \neg c \ 1), (c \leftarrow d \wedge \neg a \ 1), (e \leftarrow a \wedge b \ 0.7), (f \leftarrow c \wedge d \ 0.3), (b \leftarrow \ 0.9), (d \leftarrow \ 0.7)\}.$$

So she finds the necessity measures for each atom in the previous stable models, which result in two possibilistic stable models for P :

$$M_1 = \{(a \ 0.9), (b \ 0.9), (d \ 0.7), (e \ 0.7)\} \text{ and } M_2 = \{(b \ 0.9), (c \ 0.7), (d \ 0.7), (f \ 0.3)\}.$$

M_1 tells the teacher that she can give the student the Math assignment and the student almost certainly completes it. The certainty degree of the student completing his History assignment is much less. So now, the teacher may consider other factors, such as the students learning styles, in order to prioritize her options.

To end this section, let us reconsider the possibilistic normal program

$$P = \{(a \leftarrow b \wedge \neg c \ 1), (c \leftarrow d \wedge \neg a \ 1), (e \leftarrow a \wedge b \ 0.7), (f \leftarrow c \wedge d \ 0.3), (b \leftarrow 0.9), (d \leftarrow 0.7)\}$$

and its possibilistic stable models

$$M_1 = \{(a \ 0.9), (b \ 0.9), (d \ 0.7), (e \ 0.7)\} \text{ and } M_2 = \{(b \ 0.9), (c \ 0.7), (d \ 0.7), (f \ 0.3)\}.$$

It is not difficult to verify that $P \cup (\neg \tilde{M}_1^* \ 1) \vdash_{PIL} M_1$ and that $P \cup (\neg \tilde{M}_2^* \ 1) \vdash_{PIL} M_2$. Hence M_1 and M_2 are also possibilistic safe beliefs for P .

7 Future Work

Since lemma 3 applies to any logic program, not just normal logic programs, and lemma 4 applies to any possibilistic theory, we believe that our result may be extended to possibilistic disjunctive logic programs.

References

1. Estrada O., Arrazola J., Osorio M.: Possibilistic Safe Beliefs. LANMR 2010 (2010)
2. Nicolas P., Garcia L., Stephan I., Lefevre C.: Possibilistic uncertainty handling for answer set programming. Annals of Mathematics and Artificial Intelligence, Springer (2006)
3. Dubois D., Lang J., Prade H.: Possibilistic Logic. In: Gabbay D, Hogger C, Robinson J. (eds.) Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 3, Clarendon Press Oxford (1994)
4. Gelfond M., Lifschitz V.: The Stable Model Semantics for Logic Programming. Fifth Conference on Logic Programming, MIT Press (1988)
5. Pierce D.: Stable Inference as Intuitionistic Validity. Logic Programming, 38 (1999)
6. Osorio M, Navarro J., Arrazola J.: Applications of Intuitionistic Logic in Answer Set Programming. Theory and Practice of Logic Programming (2004)
7. Mendelson E., Introduction to Mathematical Logic, CRC Press, Fifth Edition (2010)
8. Velez R., Arrazola J., Martinez, I.: Semantics for Some Non-Classical Possibilistic Logic. Under revision for publishing (2013)
9. Osorio M., Navarro J., Arrazola J.: Safe Beliefs for Propositional Theories. Annals of Pure and Applied Logic, Elsevier (2004)
10. Estrada E., Arrazola J., Osorio, M.: Possibilistic Intermediate Logic, International Journal of Advanced Intelligence Paradigms (IJAIP), Vol. 4, No. 2 (2012)

The Memory Map Model used for Personalization in Intelligent Learning Environments

Carlos Ramírez and Benjamín Valdés

Computer Science Department,
Tecnológico de Monterrey, Campus Querétaro,
Mexico

{cramireg, bvaldesa}@itesm.mx

Abstract. The knowledge representation model called The Memory Map [1] was used to represent the expert and student modules of college courses. The students and expert modules were compared and the differences were used to personalize the sequence of learning activities for each student. The personalization scheme is designed to detect secondary knowledge gaps (concepts required to understand more complex concepts), and to estimate the correct time and place where a learning activity should be introduced. An implementation of the model was used in to represent a course to see what impact the personalization had on the students. The model was also used to represent the expert domain of an Intelligent Tutoring System.

Keywords. Knowledge representation, competences, skills, learning.

1 Introduction

Artificial Intelligence and Cognitive Science have played an important role enhancing traditional learning environments to turn them into Intelligent Learning Environments (ILE). Examples of these applications can be found in the works of knowledge representation [1], cognitive tutors [2], learning companions [3], emotion detection [4], and question generation [5], among others. ILEs in general show strong tendencies towards personalization; the sequence of learning activities in the learning flow is a central part of the instructional design process [6] and it is one of the most common types of personalization. This personalization of learning sequence means the selection of the content and the way it should be provided in a way that fulfills the need of each particular student. This is important due to the differences in learning between different students when exposed to same learning experience as a result of the differences in learning rate, different previous knowledge and the differences in abilities and competences that each student has. Indeed, cognitive tutors also referred to as Intelligent Tutoring Systems (ITS), address this phenomenon by using a scheme of one on one tutoring proposed by Benjamin Bloom [7]. Sequence adaptation involves adding or subtracting learning activities, as well as changing the order in which they are presented in the original sequence of learning activities [8]. Learning

environments should present opportunities for students to learn the knowledge they have missed in previous stages of their learning process. This can be done by personalizing their learning sequences to include those concepts missed or forgotten from the previous stages that are necessary to comprehend the current more advanced content at a pace that is in accordance to the student learning rhythm.

To achieve this kind of personalization a deep understanding of the learner's knowledge is required; the Memory Map (MM) model is designed to represent knowledge with flexible depth in a simple and complete way. This computational model focuses on the understanding of the organization of knowledge, and shows how the student's concepts are linked together with skills; therefore, instructional decisions can be made to improve the students learning process. The following sections describe the way in which the model was used for course personalization. In section 2 the knowledge representation model is presented, in section 3 sequencing of learning objectives is integrated to the model, in section 4 personalization of the learning sequences is explained, in section 5 the application of the model is shown and in section 6 conclusions and future work are presented.

2 Knowledge Representation

The definition of concepts in the MM [1] and the way their behavior are described, are influenced by the ideas of Vygotsky's constructivism [9], Fodor's Language of Thought Hypothesis [10] and Hobbes' Representational Theory of the Mind [11]. Concepts in the MM are defined as dynamic computable units which are composed of other concepts in combination with a specific kind of association function which may be seen as attributes for a given context; similar approaches can be found in the works of [12] The MM combines symbolic representation of semantic networks with a distributed local representation approach found in neural networks [13], i.e. non symbolic systems. This definition of concept and its implications in the model implementation enables the representation of several learning domains. We define a context as the integration of one or more domains; each concept has a set of attributes with relevance that changes depending on the context from which the concept is accessed, i.e., some attributes are more relevant in a particular context than others.

Each subject or academic course can be seen as a context integrated by several domains, for example a course in advanced web programming would be the combination of the knowledge domains: web, programming, servers, PHP, Perl, JavaScript and HTML, among others. Several contexts can coexist within a single MM, this is a MM with many different domains, this is treated as a student cognitive profile because it represents a natural reflection of the students' knowledge, i.e., the brain is not a rigid structure divided by separate domains it is an intertwined associative network which can activate different connections depending on the context [14]. The adaptation of a learning sequence begins when the MM is first used to represent the concepts and associations that a group of students is expected to learn during the course. This is called the course-MM and would be equivalent to the expert

domain inside the expert module of cognitive agent architecture; a small extract of a modeled course is presented in Figure 1.

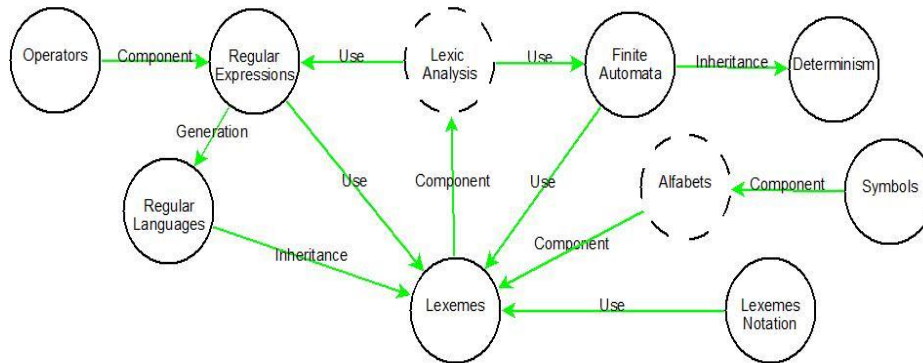


Fig. 1. A graphical representation of a segment of the Theory of Computation course the Memory Map.

3 Associative Learning Path

Once the knowledge of the domain is represented in the course-MM, a sequence layer is added. In this sequence layer the associations are numbered and treated as learning objectives. The numbers in the sequence layer create an order for the learning activities; this ordered sequence is the course Association Learning Path (ALP). The types of sequences that can be created in the ALP are based on Botturi’s Educational Environment Modeling Language E2ML patterns [15], E2ML is a descriptive and rigorous language for modeling learning objects.

Associations in the MM can be used to describe sub-concepts or attributes of concepts, therefore they are more specific and that is the main reason for using them as the learning objectives. Each association, i.e., each learning objective in the ALP corresponds to a Learning Object LO or to a group of LOs related to that specific attribute or attributes of the concept, in some cases the whole concept which implies all the associations of the concept. In [1], an association is equivalent to learning objectives and learning objectives can be achieved through any of their associated LO.

Sequence modeling is traditionally systems such as IMS-Learning Design [16] are centered in activities, more flexible and precise ALPs can be designed through the conceptual approach for the focus is the development of the concept the activities used to achieve it are the means not ends, however, context independent and modular LOs become more difficult to design when pursuing a pure conceptual approach. An example of the sequence layer of the course MM is shown in Figure 2 through the red numbers, where the first learning objective of the lexical analysis is to understand the use of lexemes and their notation, the sequence structure of the ALP represented through EML notation is presented in Figure 3.

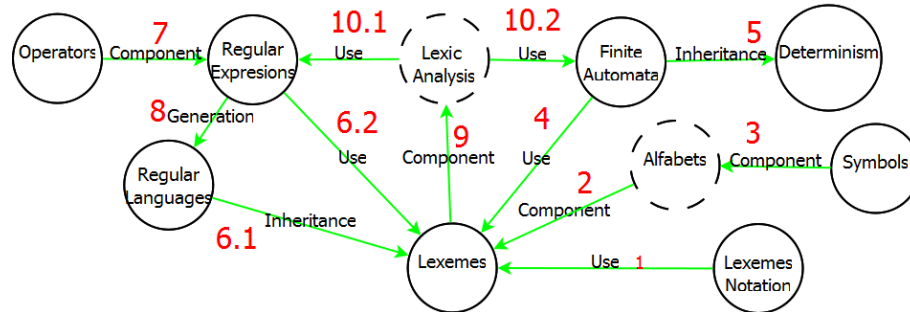


Fig. 2. The numbers indicate the order of the ALP in the Memory Map, this is the order in which learning activities are to be presented to students by default.

Course ALP:

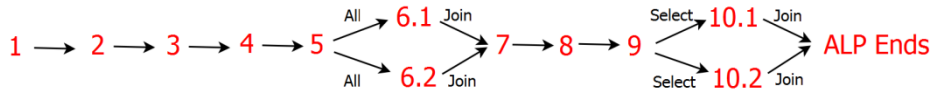


Fig. 3. A graphic representation of the ALP in EML, the type of operation: All, Join, Select or sequence is specified in the association.

4 Learning Path Personalization

An overly Model approach is used for personalization [17], the student model subset for that context (student MM) is a subset of the expert model (course MM) in that context, i.e., the subset of knowledge of the student for domain A, must be a subset of the knowledge for the course knowledge for that same domain. The Stud-MM and the Exp-MM subsets of that context are compared to determine the presence or absence of concepts and associations in the Stud-MM. Taking this into account a Stud-ALP is created for each student, this new Stud-ALP is a personalized version of the course course-ALP. The personalized ALP will integrate recursively all the extra associations and concept a student is missing and remove those concepts and associations which she/he already knows in order to satisfy the course ALP. A general example of the process will is as follows:

1. Figure 1 shows the Exp-MM with a single domain: “lexical analysis”, since there is only that domain then the Exp-Domain is the entire Exp-MM.
2. Figure 2 and 3 show the Exp-Domain with the general sequence for the course-ALP, since the student knows the majority of the content there is no inclusion of new concepts or associations.
3. The personalized ALP is created simply by deleting the associations already known by the user from the Sequence in figure 3, giving as a result the ALP that is presented on figure 4.

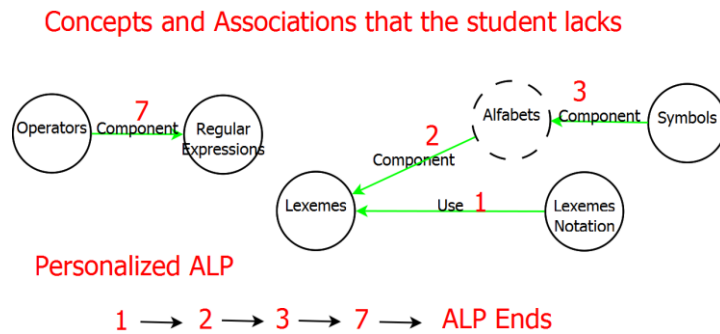


Fig. 4. The personalized ALP of a student that knows most of the concepts from figure 1.

In a different scenario where a Stud-Domain had concepts and associations missing that are required to understand the concepts of the Exp-Domain, the concepts would be included in the paths as is shown in figure 5. This previous knowledge must be specified in the Exp-Domain as such, but it is not included in the course ALP, it will only show up in the personalized ALP of students who lack the previous concepts. Basic operations performed on a course ALP include: Simple Inclusion, Deep/Recursive inclusion, Deletion, Modification. In practice basic ALP personalization is achieved through an iterative process, where student knowledge is periodically measured through evaluations mapped to particular learning objectives, the results of these evaluations are used to modify the students MM. If the students do not yet have a MM then an initial diagnostic evaluation is required, before the learning period starts. The learning sequence personalization could be carried out in real time if evaluation tools for knowledge and skills in real time were available, however since this is not the case, evaluations are carried out at specific stages through the duration of the course.

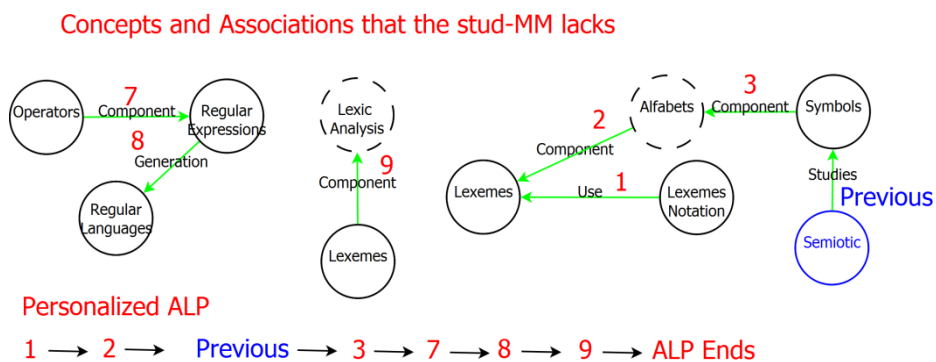


Fig. 5. The personalized ALP of a student who knows some concepts but lacks previous knowledge required to fulfill the ALP.

5 Application

The model has been implemented in a XML Schema (xsd), which describes the model's rules for the composition, and in a set of algorithms to represent model operations; the schema is then mounted to object structure in Java using JAXB. The inference algorithms for knowledge extraction use customized Iterative Depth Search First IDSF with filters. Further reading on the structure of the model and knowledge extraction can be found in [1].

To test the application of the model, the MM was used to model three different kinds of courses: Theory of Computation T.C. a partial course, Artificial Intelligence A.I. a complete course, and Searching Algorithms S.A. for an ITS. The main objective was to see what could be represented and if the model could be used to generate the personalization, the effect of such a personalization will be studied in future work.

In the first two courses (T.C. and A.I.), teachers modeled their course-MMs, the curricula objectives and the course syllabus (the thematic content) were used as a starting point to establish the MM first concepts and associations, i.e., learning objectives. These objectives were refined to match specific Learning Objects. LO were extracted from official repositories such as Merlot and Temoa. At the end of the course students answered a survey regarding their experience of the personalization of the course, performance of the students was also measured to establish correlations among personalization with the MM and performance. Both T.C. and A.I. were tested with groups of 30 students each, the groups were divided into two groups, one with personalized learning courses and one without it. They were not informed to which group they belonged in order to avoid any bias. Through the duration of the course, students were provided with complementary LOs according to what each stud-ALP suggested. The stud-ALP indicated what LO to select, as well as the moment in which to present the activity, this was done by coordinating the dates of course milestones such as weekly labs and evaluations of key concepts, with the sequence dictated by the stud-ALP. In the T. C. course the default ALP was an initial sequence of 27 associations, students who had a perfect score in the diagnostic evaluation remained with the default ALP, it should be noted that this case happened only once, which corroborates that most students come with faulty previous knowledge and misconceptions.

Students lacking previous knowledge had their ALP personalized, the average personalized ALP sequence was 33 associations and there was one student who's personalized ALP had more than 40 associations. The modeling of the A.I. course ALP had 74 associations; the average of personalized ALP was 86 associations.

The ITS follows the tradition Nwana architecture [18] and uses the MM, not to personalize ALPs, but to build entirely new ones by reacting to the student's performance, this was done by using a reacting agent which would respond each time a student answered a quiz and depending on the performance of the quiz the LP would be regenerated and would allow the student to access new content that was related to those concepts he understood better. Each student's order and difficulty in activities

were dynamically assigned as the agent would determine it using as criteria association density, student goals, and tutoring approach.

In the three cases MM was successfully used to model the expert domain and student domain: for partial domains of knowledge, complete domains of knowledge and as the Expert Model for an ITS system.

The domains were different in their granularity, in their size, and one of them was used for different means. This evidence we conclude that the MM is a flexible model that can be used both for ILE as well as for ITS.

6 Conclusions and Future Work

A knowledge representation model was used to create a sequence personalization system. The system was tested with two groups of 30 students each. The MM was successfully used for the representation of both student modeling and expert modeling in both domains and was also used to develop an ITS. Though the experiments were meant to test the applicability of the MM in learning environments, they also pave the road for future work regarding the actual impact in student performance, this is due to the nature of the research, only long term measurements can provide evidence that the personalization of ALPs have a positive impact on the overall learning process.

Acknowledgments. The authors thank Tecnológico de Monterrey campus Querétaro and campus Estado de México, and CONACYT for their financial support.

References

1. Ramirez, Carlos and Valdes, Benjamin. A General Knowledge Representation Model for the Acquisition of Skills and Concepts. *International Journal of Software Science and Computational Intelligence (IJSSCI)* 2, no. 3: 1-20 (2010)
2. Baker, Ryan S J, Albert T Corbett, Kenneth R Koedinger, Shelley Evenson, Ido Roll, Angela Z Wagner, Meghan Naim, Jay Raspat, Daniel J Baker, and Joseph E Beck. Adapting to When Students Game an Intelligent Tutoring System. In: *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, 392-401 (2006)
3. Ramirez J, Boulay BD. Expertise, Motivation and Teaching in Learning Companion Systems. *International Journal of Artificial Intelligence in Education*, 14:67-106 (2004)
4. Ramirez C, Concha C, Valdes B. Cardiac Pulse Detection in BCG Signals Implemented on a Regular Classroom Chair Integrated to an Emotional and Learning Model for Personalization of Learning Resources. In: Richards LG, Curry K, (eds.) *The 40th Annual Frontiers in Education (FIE) Conference*. IEEE, Arlington Virginia (2010)
5. Boyer, K. E., LAHTI, W. J., PHILLIP Sab, R., WALLI Sab, M. D., VOUK, M. A., & LESTER, J. C.. An Empirically-Derived Question Taxonomy for Task-Oriented Tutorial Dialogue. In S. D. Craig & D. Dicheva (Eds.), *the 14 AIED Workshops Proceedings*, pp. 9 (2009)
6. Gagne, Robert M., Walter W. Wager, Katharine Golas, and John M. Keller. *Principles of Instructional Design*. Wadsworth Publishing (2004)

7. Bloom, B. The 2 Sigma Problem: the search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, 13, 4–16. (1984)
8. Brusilovsky, P. Adaptive Hypermedia. *User Modeling and User-Adapted Interaction* 11: 87-110 (2001)
9. Vygotsky, Lev. *Thought and Language*. Ed. Alex Kozulin. MIT Press, New York (1986)
10. Fodors, J. A. *The Language of Thought*. Harvard U. Press, July. Cambridge (1975)
11. Hobbes, T. *Elements of Law, Natural and Political*. Routledge (1969)
12. Wang, Y. The OAR Model of Neural Informatics for Internal knowledge Representation in the Brain. *Int'l Journal of Cognitive Informatics and Natural Intelligence*;1(3):66-77 (2007)
13. Rumelhart, D.E. & McClelland, J.L. *Parallel distributed processing: explorations in the microstructure of cognition*. Foundations vol. 1 Cambridge, MA: MIT Press. (1986)
14. Pinker, S. *How the Mind Works*. W. W. Norton & Company; 672 (1999)
15. Botturi, L. *E2ML Educational Environment Modeling Language*. Thesis, University of Lugano Faculty of Communication Science (2003)
16. Koper, R., & Tattersall, C. *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training*. Springer, Heidelberg (2005)
17. Sosnovsky, S., Dolog, P., & Henze, N. Translation of overlay models of student knowledge for relative domains based on domain ontology mapping. R. Luckin, K. R. Koedinger, & J. Greer, (Eds.) *Frontiers in Artificial Intelligence in Education Building Technology Rich Learning Contexts That Work*, IOSPress, 158, 289–296 (2007)
18. Nwana, H. Intelligent tutoring systems. *Artificial Intelligence Review*, 4(4), 251–277 (1990)

Design and Implementation of an Affective ITS

María Lucía Barrón-Estrada, Ramón Zatarain-Cabada,
Rosalío Zatarain-Cabada, and Arminda Barrón-Estrada

Instituto Tecnológico de Culiacán, Culiacán Sinaloa,
Mexico

{rzatarain, lbarron}@itculiacan.edu.mx,
luciabarron@gmail.com

Abstract. We present the different steps used for developing an Affective and Intelligent Tutoring System for Mathematics Learning. The intelligent tutoring system evaluates cognitive and affective aspects of the users or students in order to present the learning material to them. The whole system applies a fuzzy logic system to decide the next exercise and a neural network to recognize user emotions.

Keywords. Intelligent tutoring systems, social networks, neural networks, emotion recognition.

1 Introduction

The work of a tutor is to teach and train a student through individualized instruction. This personalization that a human tutor achieves with the student is done through different ways of adapting the educational material to student needs. In order to do this job, the tutor makes use of his academic knowledge, his experience, and his observations of the student. On the other hand, an Intelligent Tutoring System (ITS) combines methods of Artificial Intelligence (AI) and education, with the aim of creating a flexible and interactive environment that considers the cognitive and affective states of students [1-3].

In this paper we present the methodology used to implement an affective and intelligent tutoring system for learning multiplication and division operations for third-grade students. The ITS incorporates a fuzzy logic system for assigning the next exercise to the student, artificial neural networks for recognizing student emotions, and Knowledge Space Theory [4] for structuring and representing the knowledge domain.

The paper is organized as follows. In Section 2 we present the design of the ITS. Section 3 gives information about its implementation. Section 4 comments some results and finally conclusions are presented in Section 5.

2 Design of the ITS

Before coding software, you must first make a sketch of software components that perform the functions for which they are intended. Here begins the design phase, where you get an overview of how a system works with the correct distribution of

components based on your proposed software architecture. This will guide the implementation process.

2.1 ITS Architecture

In this section we present the ITS architecture (figure 1) which is formed by three main modules: The Expert Module called Domain Module, which represents the knowledge of the expert and is handled through different concepts related to Knowledge Space Theory. The knowledge base of this module is stored using a particular kind of XML format.

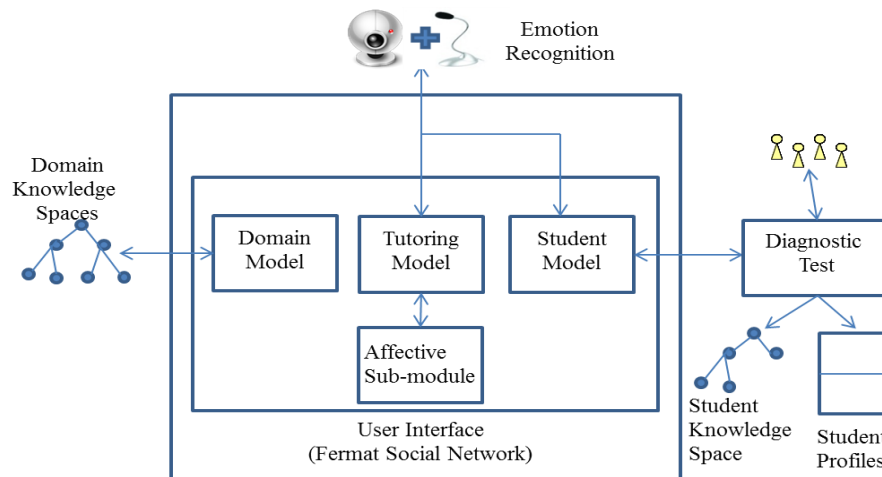


Fig. 1. ITS Architecture.

The Student Module provides the information about student competencies and learning capabilities through a diagnostic test. The student module can be seen as a sub-tree of all knowledge possessed by the expert and a student profile. For every student there is a static profile, which stores particular and academic information, and a dynamic profile, which stores information obtained from the navigation on the tutor and from the recognition of emotions.

The Tutoring Module presents the exercises to the students according to the level of the problem. We implemented production rules (procedural memory) and facts (declarative memory) via a set of XML rules. Furthermore, we developed a new knowledge tracking algorithm based on fuzzy logic, which is used to track student's cognitive states, applying the set of rules (XML and Fuzzy rules) to the set of facts.

2.2 Affective State Recognition

Affective State Recognition is based on Ekman's theory [5], which recognizes ten emotions, but we are only working with seven emotions: anger, disgust, fear, happiness, sadness, surprise, and neutral. We use Kohonen Neural networks [6] with 20X20 input neurons and 7 output neurons representing the emotion.

3 Implementation of the ITS

One of the main advantages of our ITS is that it works within a social network in a Web environment. Creating a web application has a large number of options for its construction. The tools and programming languages that were chosen for the development of intelligent tutoring system were:

- HTML5: For structuring and presenting the contents in the web.
- Java: For programming the ITS.
- JavaScript: For working with the user interface (the social network).
- JSP: (Java Server Pages): For dynamic creation of contents using Java.
- MySQL: For Data base Management.
- JSON (JavaScript Object Notation): For data interchange.

```

division([
    {"divisor":9,"dividendo":[1,0,8],"cociente":[0,1,2],"residuo":[1,0],"mul":[9,18]},
    {"divisor":2,"dividendo":[4,2],"cociente":[2,1],"residuo":[0,0],"mul":[4,2]},
    {"divisor":11,"dividendo":[1,0,0],"cociente":[0,0,9],"residuo":[1],"mul":[99]},
    {"divisor":10,"dividendo":[5,0,0],"cociente":[0,5,0],"residuo":[0,0],"mul":[50,0]},
    {"divisor":5,"dividendo":[7,2,5],"cociente":[1,4,5],"residuo":[2,2,0],"mul":[5,20,25]},
    {"divisor":20,"dividendo":[1,1,2],"cociente":[0,0,5],"residuo":[0,19],"mul":[100]},
    {"divisor":2,"dividendo":[4,0,9],"cociente":[2,0,4],"residuo":[0,0,0],"mul":[4,0,8]},
    {"divisor":20,"dividendo":[5,0,2,0],"cociente":[0,2,5,1],"residuo":[10,2,0],"mul":[40,100,20]},
    {"divisor":14,"dividendo":[1,3,2],"cociente":[0,0,9],"residuo":[6],"mul":[126]},
]);

```

Fig. 2. Structure of a JSON File for a Division.

3.1 The Expert Module

This module was created from JSON files. These contain all the information a student needs to know about how to solve a math division. The structure of these files is presented in Figure 2.

Figure 2 shows the basic structure of a JSON file. This case consists of an array of objects which contain the attributes of "divisor" and "dividend" that are shown to the student. Following are the attributes: "quotient", "remainder" and "mul", which

contain the correct answers. There are other files, which are chosen according to the difficulty determined by the components.

3.2 The Student Module

It was implemented to determine the initial level of knowledge that a student possesses. For this, we implemented in HTML5 and JavaScript a diagnostic test whose answers are stored in a JSON file with a similar structure to that of Figure 2. The Student answers are compared with the contents of the JSON file. Each question has a value and a student score is determined by the following formula:

$$\text{Student Score} = \text{Total Earned Points} / \text{Total points for all questions}$$

The result is evaluated by a small algorithm to determine the difficulty in which the module tutor should start with the course.

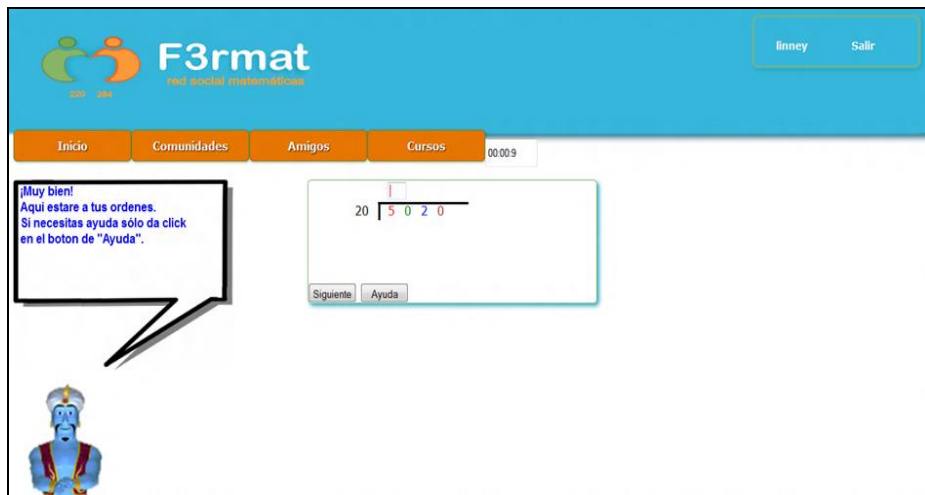


Fig. 3. Main Interface of the Intelligent Tutor.

3.3 The Tutor Module

This module is mainly based on the theory of cognition ACT-R [7]. In this part, the student solves exercises with the help and support of the intelligent tutor. Figure 3 shows the interface of this module.

Once inside the interface, students can enter answers they think are correct, while the tutor dynamically checks the corresponding JSON file. The initial difficulty is that which was determined in the student module. The difficulty of the next exercises can be modified depending on what the fuzzy expert system determines. The functionality of how responses are evaluated and the path taken by the process of solution is shown in Figure 4.

The premise is simple. The ITS waits for the entry of a value, and verifies that the value is correct to move to the next box and wait for the next value. If the answer is incorrect, the tutor sends a message through a pedagogical agent about the type of error. This is repeated until the division is complete. During this process the student can make use of two buttons located below the division operation. Button "Help" sends tips or advices to the student through the pedagogical agent. Button "next" moves to the next operation.

To determine the emotion, we first take the image which is transformed to a more basic form. Based on this picture we get the feature points that minimize the set of input data to the neural network. We use a Kohonen Neural network with 20X20 input neurons and 2 output ones representing the emotion. For the detection of emotions in the voice, this is captured primarily through the computer microphone and then is normalized. Then we apply the technique to characterize components analysis (PCA) to the signal representing the voice. After using the SFFS method [8] we obtain an optimal set of features that will feed the neural network. Each neural network used to recognize emotions produces an output. All outputs of each neural network are integrated using fuzzy logic which gives us a final result that is the emotion of the user that the system recognizes.

Table 1. Fuzzy Values for Variable Difficulty.

	Difficulty (%)	Normalized Values
Very Easy	0% - 10%	0 – 0.1
Easy	0% - 30%	0 – 0.3
Intermediate	20% - 80%	0.2 – 0.8
Difficult	70% - 100%	0.7 – 1.0
Very Difficult	90% - 100%	0.9 – 1.0

Fuzzy Expert System: The tutoring module also has an application made in java for reasoning with fuzzy logic. This program takes input fuzzy variables such as time, number of errors and number of helps. In a way the student's performance is reflected by such variables as he/she works with an exercise. We implemented a Fuzzy Expert System that eliminates arbitrary specifications of precise numbers and creates smarter decisions, taking into account a more human reasoning. Fuzzy sets are described in table 1 and figure 5 for linguistic variable Difficulty with fuzzy values very easy, easy, intermediate, difficult, and very difficult (“muy fácil”, “fácil”, “básico”, “difícil” and “muy difícil” in Spanish).

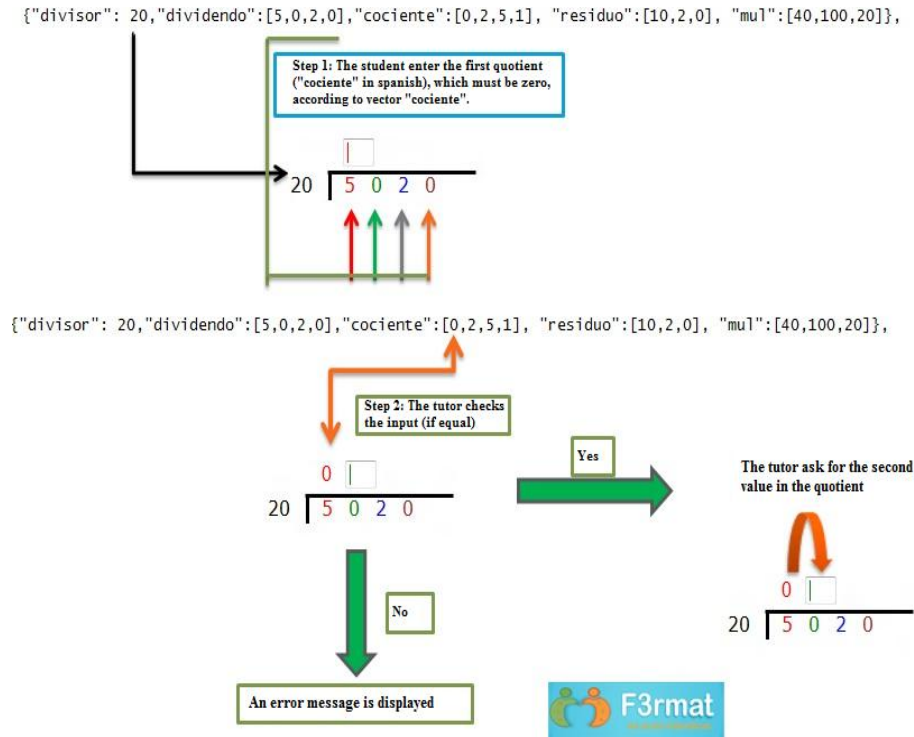


Fig. 4. Evaluation of a division.

On the other hand, some of the fuzzy rules which determine the degree of difficulty of the next student's problem are:

- If (Error is small) and (Assistance is small) and (Time is very fast) then (Difficulty is very difficult)
- If (Error is small) and (Assistance is normal) and (Time is slow) then (Difficulty is difficult)
- If (Error is big) and (Assistance is big) and (Time is very slow) then (Difficulty is very easy)

In this case the first fuzzy rule establishes that if a student (solving an exercise) had few errors, few assistances (helps), and finished in little time, the difficulty of the next exercise must be higher.

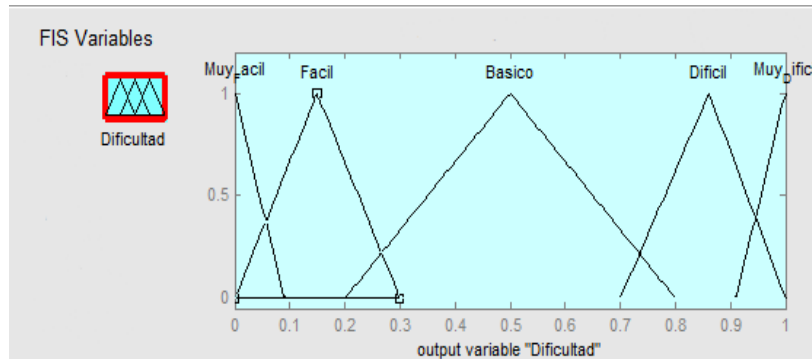


Fig. 5. Fuzzy sets for variable difficulty.

4 ITS Testing

The intelligent tutoring system was evaluated by a group of 72 children from third grade (Figure 6) in public and private schools. Before the evaluation we offered a small introduction of 15 minutes with the environment of the tool. We evaluated the subject of multiplications and divisions. We applied a test with different exercises before and after the students used the ITS. The results showed a good improvement in most students (more so in students with lower initial grades) using one of the two teaching methods for multiplication: traditional and lattice.

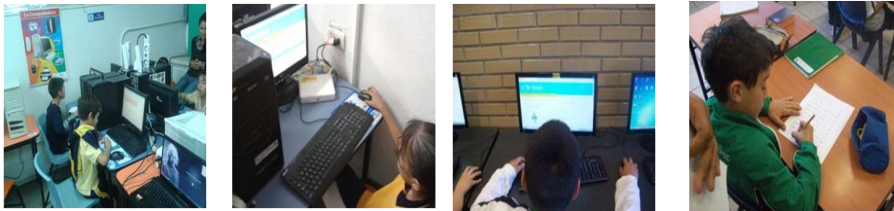


Fig. 6. Children testing the ITS in public and private schools.

5 Conclusions

Our intelligent tutoring system has been implemented for about one year. The results showed good improvement in the students. We are still working to adapt the application of emotion recognition with the intelligent tutoring system. One of the problem is that the ITS run in the web and the recognizer is still being tested in a desktop environment. We believe we will finish it in the next three months.

References

1. Arroyo, I., Woolf, B., Cooper, D., Burleson, W., Muldner, K., Christopherson, R.: Emotions sensors go to school. In: Diminitrova, V., Mizoguchi, R., Du Boulay, B., Graesser, A. (eds.) Proceedings of the 14th International Conference on Artificial Intelligence in Education, pp. 17-24. IOS Press, Amsterdam (2009)
2. D’Mello, S.K., Picard, R.W., Graesser, A. C.: Towards an affective-sensitive AutoTutor. Special issue on Intelligent Educational Systems IEEE Intelligent Systems 22(4), 53-61 (2007)
3. Kort, B., Reilly, R., & Picard, R. W. An Affective Model of Interplay between Emotions and Learning: Reengineering Educational Pedagogy—Building a Learning Companion. Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT '01). IEEE Computer Society, 43-50 (2001)
4. Doignon, J. –P. and Falmagne, J. C.: Knowledge Spaces. Springer-Verlag (1999)
5. Ekman P, Oster, H.: Facial expressions of emotion. Annual Review of Psychology 30:527-554 (1979)
6. Kohonen, T.: Self-Organization and Associative memory, Springer-Verlag, Third Edition, (1989)
7. Anderson, R., Boyle, C. F., Corbett, A. T., & Lewis, M. W.: Cognitive modeling and intelligent tutoring. Artificial Intelligence, 42, 17-49. Doi: 10.1016/0004-3702(90)90093-F. (1990)
8. Pudil P, Novovičová J, Kittler J.: Floating search methods in feature selection. Pattern Recognition Letters 15 (11) (1994)

On a LS-Adaptive Learning Objects Creation Methodology Using LOM Metadata

Aremy Virrueta-Gordillo^{1,2}, Rodolfo Ibarra-Orozco¹, Juan Carlos Lopez-Pimentel¹, and Victor Ramos-FonBon¹

¹ Universidad Politecnica de Chiapas,
Calle J. Selvas s/n. Tuxtla Gutierrez, Chiapas, Mexico
{avirrueta, ribarra, jpimentel, vramos}@upchiapas.edu.mx

² Universidad Politécnic de Valencia
Camino Vera S/N. 46007, Valencia, Spain.
arvirgor@posgrado.upv.es

Abstract In this paper, we present a methodology to generate Adaptive Learning Objects based on the students' learning style orientation. An adaptive hypermedia methodology is applied in order to classify a Learning Object (LO) and, then, the obtained classification is inserted into the LO, so a Learning Management System can present the LOs that correspond with the students learning style preferences. At this current research stage, we are particular interested in developing a framework to build Adaptive Learning Objects, in which the LO and the students' learning styles are manually classified. In the following stage, we will focus on classifying these elements in an automatic way, by means of machine learning techniques.

1 Introduction

The e-learning strategy hasn't shown consistent good results. Several projects have had failed results, e.g., [1], [2], [14], and so the learning process has finished in an incomplete way. Some initiatives, with the goal of achieving the learning digital objective, have been proposed. For example, blended-learning, [13], combines classroom with digital learning courses; Castillo, [6,8], proposes a learning objects development to fit specific student features. Besides, some standards to build learning objects have been reviewed and developed in order to enable an efficient learning objects management into the Learning Management System (LMS). Automation of pedagogical tools have enhanced learning common environments and transferred this experience to virtual scenarios.

The work described in this paper merges several of the mentioned initiatives with the main purpose of defining Learning Styles-based Adaptive Learning Objects (ALO) for courses presentation through a system (LMS).

The remainder of the paper is organized as follows. Section two describes the students and learning objects interaction. Section three describes, in a general way, the LOM-IEEE standard. Section four gives an overview of the learning

styles and adaptive hypermedia concepts. Section four includes a LO standards and theory learning styles theory analysis from an adaptive hypermedia system (AHS) perspective. In section five, the proposed Methodology for LOs creation based on students learning styles is presented. Finally, conclusions are discussed and future work is presented in section six.

2 Students and Learning Objects Interaction.

Learning Objects (LOs) are the fundamental entity inside e-learning courses. The IEEE LOM proposes the following LOs formal definition: A LO is “any entity, digital or non-digital, that can be used, re-used or referenced during technology supported learning”, [3].

Some standards to create LOs have been proposed. These standards specify LOs as an organized metadata collection and allow LO being accessible, adaptable, interoperable and reusable for any LMS. Some Standards used to create LOs are SCORM,³ IMS,⁴ and LOM,⁵. These standards don't integrate elements that can be useful to accomplish a LO and student interaction. The IMS-LD standard, considers interaction into package elements, but the available LMSs are not yet ready to integrate such packages into courses. In this work, metadata is used to describe interaction elements between the student and the LO with the objective of generate adaptive learning objects (ALOs).

In traditional learning environments, learning content and student interaction is guided by a teacher who applies different pedagogical strategies to ensure each student learning. In an e-Learning environment, LOs must be presented to students in a carefully designed sequence to keep the students motivated by the course content and to make them feel that the learning content meets their needs. The ALO presentation involves two main steps: The ALO creation and the student preference profile detection.

In order to include interactive elements in a LO, a classification structure is specified. This classification structure is based on the dimensions of Adaptive Hypermedia Systems (AHS), where the learning styles theory is associated with the AHS's dimensions.

Next sections describe the defined LO learning style based classification and how this classifications elements are integrated into metadata.

3 Learning Object Metadata

IEEE LOM (Learning Object Metadata) is generally accepted as the standard for providing metadata to multimedia learning resources. The aim of using metadata for describing learning objects is to promote the learning material sharing.

³ Sharable Content Object Reference Model, <http://www.adlnet.gov/Technologies/scorm>

⁴ IMS Content Package Specification, http://www.imsglobal.org/content/packaging/cpv1p2pd2/imscp_primerv1p2pd2.html

⁵ Learning Object Metadata. Learning Technology Standards Committee, http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf

LOM uses the following categories to describe resources. These can be seen as a superset of the Dublin Core elements.

1. General: groups the general information that describes this resource as a whole.
2. LifeCycle: describes the history and current state of this resource and those that have affected this resource during its evolution.
3. Meta-MetaData: describes the specific information about the metadata record itself (rather than the resource that this record describes), who created this metadata record, how, when, and with what references.
4. Technical: describes the technical requirements and characteristics of this resource.
5. Educational: describes the key educational or pedagogic characteristics of this resource. This category stores the pedagogical information essential to those involved in achieving a quality learning experience. The audience includes teachers, managers, authors, and learners.
6. Rights: describes the intellectual property rights and conditions of use for this resource.
7. Relation: defines the relationships among this resource and other targeted resources, if there are any. Multiple relationships can be supported.
8. Annotation: provides comments on the educational use of this resource, who created this annotation and when.
9. Classification: describes where this resource is placed within a particular classification system. To define multiple classifications, there may be multiple instances of this category.

Studies about the real use of the LOM standard show that metadata is often misused or not instantiated. Such results are mostly due to the high specification complexity. Besides, some of the metadata values are subjective, so it is difficult to assign a value to them.

4 Learning Objects Classification System

To classify the LO is necessary to define a classification system that integrates a metadata standard into LOs. It is important to understand that a classification system divides a domain of reality in an ordered series of categories and subcategories. In this case the domain is a screening tool of learning preferences and categories are the learning styles that define the tool. Also, we use the dimensions that define Adaptive Hypermedia Systems (AHS) to support the identification of the elements of the classification system

4.1 Adaptive Hypermedia Systems

Adaptive hypermedia is concerned with the functionality of hypermedia, in a way that they become personalized. An adaptive hypermedia system gathers

information about users and their behavior and, according to their needs, goals, settings and actual knowledge the information is adapted and then, presented in a personalized way.

Many systems are based on the principles of adaptive hypermedia, e.g., information retrieval systems, on-line information systems, on-line help systems, educational hypermedia systems, etc.

Examples of educational hypermedia systems are the ISIS-Tutor System, [5], a learning environment adaptive hypertext; The Anatom-Tutor, [4], an intelligent tutor to teach anatomy; Shaboo, [11], a tutor to teach the basic concepts of programming oriented objects; Online SHARP, [15], a system applied to solving mathematical problems. These systems use adaptation techniques for adapting the information presented to the user.

4.2 Learning Styles Detection Tools

Detection tools for learning styles identify preferred ways in which a person can learn. Each person has a learning preferred way. These preferences are grouped into styles and are known as "Learning styles", [10]. Several definitions of learning styles currently exist. Keefe, [16], defines learning styles as being characteristic of the cognitive, affective, and physiological behaviors that serve as relatively stable indicators of how learners perceive, interact with, and respond to the learning environment. Dunn, [7], describes learning style as "... the way each learner begins to concentrate, process, and retain new and difficult information". Morales, [17], defines learning styles as a pedagogical model for classifying student-associated cognitive issues.

Several studies have been done to detect learning styles in students, e.g., the investigation conducted by Fleming [12], which generated the VARK test,⁶; the Honey-Alonso questionnaire⁷; and the model designed by Felder and Silverman⁸, [10], which was implemented by Spurlin, [11]. This latter model seems to be the most appropriate for the use in computer-based educational systems, [9]. Most learning style models classify students in few groups, whereas Felder-Silverman Learning Styles Model (FSLSM) describe the learning style in a more detailed way, distinguishing four learning style "dimensions".

- The first dimension distinguishes between an active and a reflective way of processing information. Active learners learn best by working actively with the learning material, e.g. working in groups, discussing the material, or applying it. In contrast, reflective learners prefer to think about and reflect on the material.
- Sensing-intuitive learning dimension. Learners with preference for a sensing learning style like to learn facts and concrete learning material. Sensing learners tend to be more practical than intuitive learners and like to relate

⁶ VARK: A guide to learning styles, <http://www.vark-learn.com>

⁷ CHAEA Questionnaire, <http://www.estilosdeaprendizaje.es>

⁸ ILS: Index of learningstyles, <http://www.engr.ncsu.edu/learningstyles/ilsweb.html>

the learned material to the real world. Intuitive learners prefer to learn abstract learning material. They like to discover possibilities and relationships, and tend to be more innovative than sensing learners.

- The third, visual-verbal dimension differentiates learners who remember best what they have seen, e.g. pictures, diagrams and flow-charts, and learners who get more out of textual representation, regardless of the fact whether they are written or spoken.
- In the fourth dimension, the learners are characterized according to their understanding. Sequential learners learn in small incremental steps. In contrast, global learners use a holistic thinking process and learn in large leaps. They tend to absorb learning material almost randomly without seeing connections but after learning enough material they suddenly get the whole picture.

5 Our proposed LS-ALO Creation Methodology

The proposed methodology comprises four steps, grouped in two processes:

1. Classification system definition.
 - Step 1. Specification of the adaptive hypermedia dimensions.
 - Step 2. Values specification for each LO category.
2. LS-based Classification and LO integration.
 - Step 3: Select a LO metadata standard.
 - Step 4. Insert classification data into the LO's metadata.

In the following two sections, we use an example to demonstrate how to define the classification system based on the Felder-Silverman learning style model. First, the system relates the adaptive hypermedia dimensions with the learning style model and then, we present a proposal to include the obtained classification data into the IEEE/LOM metadata.

5.1 Classification system definition

Step one: Specification of the adaptive hypermedia dimensions. Table 1 describes the association between the concepts of Learning Styles model and Adaptive Hypermedia dimensions.

Step two: Values specification for each learning object category. The LO categories correspond to the learning styles proposed by the model. The range of values for each category is determined by the student's learning style belonging level.

The Felder-Silverman categories and its belonging levels are shown in Table 2 and Table 3.

When the categories are identified and the belonging levels are established, we have specified the classification system.

For this example we have defined a classification system with four categories. These have a value from a defined range. The value set for each category depends on the features of LO's the content.

Table 1. Adaptive Hypermedia Dimensions

Adaptive Hypermedia Dimensions	Learning Styles Concepts	Example
Where adaptive hypermedia systems can be helpful?	In LMS to ensure that the educational content are properly presented to students	Any LMS such as .LRN, Moodle, Blackboard, etc.
What features of the user are used as a source of the adaptation?	Student profile detected from a model of learning styles.	The Felder-Silverman model that defines a profile based on styles AR, SI, VV and SG
What can be adapted?	Learning objects described by a standard and managed by LMSs.	Learning objects described by IEEE LOM metadata
What are the adaptation goals?	To provide student ALOs associated to his learning profile.	Develop algorithms for the LMS manages the ALO adaptive presentation.

Table 2. Classification Categories

Abbreviature	Description
AR	Active-Reflective
SI	Sensory-Intuitive
VV	Visual-Verbal
SG	Sequential-Global

5.2 LS-based Clasification and LO integration.

A LO is considered as an ALO when classification data is included in the specification of the LO metadata.

The following processes in the methodology involves a) selecting a LO standard and b) including classification categories as an elements' specification.

Step three: Select a LO metadata standard. It is necessary to select a LO standard to analyze metadata in detail. The result of this analysis will be the identification of metadata in which classification elements may be included.

As result of the IEEE/LOM analysis, classification category was identified as the metadata into which the classification specification can be integrated with the standard, because it describes the LO belonging to a particular classification system.

Step four: Insert classification data into the LO's metadata. This step consists in adding items to the categories of the selected LO standard. For this example, the Classification category elements used to describe the classification system are:

Table 3. Belonging Levels

Value	Belonging level
1-3	Appropriate balance
5-7	Moderate belonging
9-11	Strong belonging

- “9.2. Taxon path” is used to define the classification. This item includes other elements we have used: “9.2.1. Source” that indicates the category name of the classification system and “9.2.2. Taxon”, that indicates the category value. “9.2.2. Taxon” has other elements to describe the value of the category. In “9.2.2.1. ID” is placed the belonging value. In “9.2.2.1. Enter” describes the belonging value.
- “9.3. Description” is used to indicate a description of object classified.
- “9.4. Keywords” includes keys for easy search and LOs retrieval.

By following the methodology four steps, we have an ALO classified by Learning Styles. However, in order to accomplish an easy metadata insertion, it is necessary to create the ALO through an automatic applications. Exe Learning⁹ and Reload Editor¹⁰ are applications that allow the inclusion of metadata from the user interface, so we must create the metadata which describes the ALO and let the application generate the object.

It is important to note that the responsibility for classifying the LO corresponds to the object author. The author must be informed about the classification system and the student learning relations.

6 Conclusion and Future Work

In this paper we have presented a methodology to create ALO including classification elements that will be used for the presentation of the LOs content according to the students learning preferences. Our methodology describes how to define a classification system based on a learning style model and explains how to integrate the obtained classification with the standard LO. The object classification is just one of several activities to ensure that students have access to materials that fit their learning preferences in an online course, besides, for example, it is necessary to detect the student profile and register it into a LMS, then an intelligent algorithm to relate the student learning profile with the ALO elements must be developed.

Our future work comprises three initiatives. First, the integration of learning styles models into LMSs. An prototype can be found at <http://moodle.virrueta.org>. Second, the development of intelligent algorithms for an adaptive presentation

⁹ eXe Project, <http://exelearning.org/>

¹⁰ Reusable eLearning Object Authoring and Delivery, <http://www.reload.ac.uk/editor.html>

of objects through LMSs, and third, to develop applications to allow authors to create adaptive learning objects in an efficient way.

References

1. J. Akeroyd. Information management and e-learning: Some perspectives. In *Aslib proceedings*, volume 57, pages 157–167. Emerald Group Publishing Limited, 2005.
2. S. Alexander. E-learning developments and experiences. *Education+ Training*, 43(4/5):240–248, 2001.
3. C. Arteaga and R. Fabregat. Integración del aprendizaje individual y del colaborativo en un sistema hipermedia adaptativo. *JENUI*, 2(2):107–114, 2002.
4. I.H. Beaumont. User modelling in the interactive anatomy tutoring system ANATOM-TUTOR. *User Modeling and User-Adapted Interaction*, 4(1):21–45, 1994.
5. P. Brusilovsky and L. Pesin. ISIS-Tutor: An adaptive hypertext learning environment. In *Proceedings of JCKBSE*, volume 94, pages 10–13, 1994.
6. L. Castillo, L. Morales, A. González-Ferrer, J. Fernández-Olivares, and Ó. García-Pérez. Knowledge engineering and planning for the automated synthesis of customized learning designs. *Current Topics in Artificial Intelligence*, pages 40–49, 2007.
7. R. Dunn. Understanding the dunn and dunn learning styles model and the need for individual diagnosis and prescription. *Reading, Writing, and Learning Disabilities*, 6(3):223–247, 1990.
8. J. Fdez-Olivares, L. Castillo, O. Garcia-Pérez, and F. Palao. Bringing users and planning technology together. Experiences in SIADEX. In *Proc ICAPS*, pages 11–20, 2006.
9. R.M. Felder and L.K. Silverman. Learning and teaching styles in engineering education. *Engineering education*, 78(7):674–681, 1988.
10. R.M. Felder, L.K. Silverman, and B.A. Solomon. *Index of learning styles (ILS)*. North Carolina State University, 1999.
11. R.M. Felder and J. Spurlin. Applications, reliability and validity of the Index of Learning Styles. *International Journal of Engineering Education*, 21(1):103–112, 2005.
12. N.D. Fleming. I’m different; not dumb. Modes of presentation (VARK) in the tertiary classroom. In *Research and Development in Higher Education, Proceedings of the 1995 Annual Conference of the Higher Education and Research Development Society of Australasia (HERDSA)*, HERDSA, volume 18, pages 308–313, 1995.
13. Charles R. Graham. Blended learning systems: Definition, current trends, and future directions. In *In*, pages 3–21. Pfeiffer Publishing, 2005.
14. A. Gunasekaran, R.D. McNeil, and D. Shaul. E-learning: research and applications. *Industrial and Commercial Training*, 34(2):44–53, 2002.
15. R.R. Hernández, A.B.G. González, F.J.G. Peñalvo, and R.L. Fernández. Sharp online: Sistema hipermedia adaptativo aplicado a la resolución de problemas matemáticos. *IX Congreso Internacional Interacción*, pages 271–284, June 2008.
16. J.W. Keefe. Assessing student learning styles: An overview. *Student learning styles and brain behavior*, pages 43–53, 1982.
17. L. Morales and G. Roig. Connecting a technology faculty development program with student learning. *Campus-Wide Information Systems*, 19(2):67–72, 2002.

A Model for the Representation of Competences Applied to Student's Knowledge Modelling

Carlos Ramirez¹ and Erik Sanchez²

¹Computer Science Department of the Tecnológico de Monterrey, Campus Querétaro, México
cramireg@itesm.mx

²DASL4LTD Research Lab, Tecnológico de Monterrey, Campus Querétaro, México
A00888867@itesm.mx

Abstract. A skill is a basic unit for cognitive processing that allows the use of concepts. A competence is the result of the application of a skill on a concept. Competences and their development play a central role in most educational and training programs. This paper presents a formal model for the representation of competences, called Competences Memory Map, an extension of the Memory Map [14], a model capable of representing knowledge in an integrated, simple and flexible way. The model is described using set theory through concept algebra and is computationally implementable. The formal description of algebraic and algorithmic operations over the model, as well as case studies, demonstrate that all the model properties hold as expected when dealing with complex real knowledge structures.

Keywords: Knowledge representation, Competences, Skills, Learning.

1 Introduction

Competences and their development have acquired a key role in many current teaching and training methods. The way of conceiving the processes of learning and teaching in most current educational models is based on the constructivist theory [6]. These models have become oriented towards the development of skills that can be applied in different contexts, i.e., the development of competences.

A competence is defined as the capacity of a person to use knowledge and skills in different situations either personal or professional [8]. The study of competences requires its analysis in terms of each of its components: knowledge and abilities, for the understanding of their whole functioning. There are different approaches to computationally model the process of learning, i.e., acquiring skills and knowledge and modelling them; among the most prominent models are OAR [16], The Knowledge Spaces Theory (KST) [4] and The Memory Map (MM) [14]. Several of the most relevant models for skills representation are: The Skills Theory [6], The Model of Skills Acquisition [1] and The ACT Model [2]. Between the most recognised models for the representation of competences, are: CbKST [3], The ELEKTRA Ontology Model [3] and The Competency Ontology [11].

In this paper it is proposed a representation model of competences called Competences Memory Map (MM-Competences), a model capable to represent the components of a competence in an integrated and simple way, allowing also the modelling of the relationships between the elements of a competence and the structures created by the association of multiple competences. Section 2 presents a review of the basic elements involved in the model: concepts, skills and competences; Section 3 presents the model and its components; Section 4 presents the properties of the model; Section 5 describes the operations that can be performed within the model; Section 6 describes the competence structures forming process, and the distinctive properties and issues about the model. Finally, Section 7 presents conclusions and future work.

2 Concepts, Skills and Competences

2.1 Concepts

Knowledge is one of the two fundamental components of competences. It is defined according to O*NET as a collection of related facts, information and principles about a particular area that can be acquired through education and experience [10].

Knowledge has a fundamental unit of representation: the concept, without it, knowledge cannot be conceived nor represented. In Hobbes and Fodor works [14] the concept is conceived as the representation of a mental object and its attributes that can be manipulated and expressed symbolically through language.

2.2 Skills

Skill is the other fundamental element of a competence. KING [8] defines a skill as “an ability acquired by training that uses implicit memory to apply knowledge to standard situations and problems.” On the other hand, in [14] Ramirez refers to cognitive skills as mental processes that interact with concepts through its application, with a given goal and with internal or external effects in the person who exercises it.

There are different models of representation of skills, as mentioned above in the introduction. One of the main properties of the skills is that they can be organised hierarchically according to their complexity, in terms of the cognitive processes used in their execution. A skill may require the use or mastering of another skill of less than or equal complexity. Some researchers propose taxonomies such as the O*NET Skills Taxonomy [10], Paquette’s Taxonomy [12], Bloom’s Taxonomy [7] and Revised Bloom’s Taxonomy [7].

2.3 Competences

In order to define a computational model capable of representing competences, its associations and behaviour, these need to be analysed in terms of its basic components: the knowledge and the skills. There are many definitions of competence [3, 4, 8, 10, 11, 12, 13]. From a computational representation approach and based on the

definition of generic skills of Paquette [12] as "processes that act on the knowledge in a domain of application", competences are defined as declarations that can be demonstrated with the application of a generic skill to some knowledge, with a given performance. There are different models of competence representation, as mentioned above in the introduction.

The skills are generic cognitive processes that do not have a fixed hierarchy and complexity and guide the performance a task. Skills act on a given context and generate a competency when applied to the corresponding domain of knowledge. This is the reason why both, skills and concepts, are essential elements of competences.

3 The Competencies and Skills Memory Map

The MM-Competences is intended to model the mental state of a person in terms of his knowledge, skills and competences within a given domain. The operation of this model is based on the competences of a person and its computational representation through a unit called Competences-RU. This unit integrates knowledge, i.e., networks of concepts, and skills.

The MM-Competences Structure is formed by the set of competences developed by a person regarding a set of domains and the associations between them. A substructure of the MM-Competences determines the set of competences to apply in a given domain of knowledge to achieve a goal. This substructure is capable of modelling the sequence in which the competences are required through the associations between them. The MM-Competences is modelled as a directed graph where nodes are representing the competences and the arcs associations between them. The direction of the associations comes from general level competences to specific ones.

3.1 Concept Representation Unit

The Memory Map [14] fundamental unit is the Concept Representation Unit (Concepts-RU). Concepts-RUs operate as nodes of a dynamic and adaptable network. The Concepts-RU attributes are: Name, Identifier, Textual descriptions, Keywords and Concepts-RU Associations.

The Concepts-RU can be defined using set notation as the tuple:

$$c(A^c, A^{cp}) \quad (1)$$

where A^c represents a set of all associations that Concepts-RU c has with other Concepts-RU, and A^{cp} represents a set of all associations with Competences-RU in which c is a member. A^c and A^{cp} are defined as follows:

$$A^c = \{a^c_1, a^c_2, a^c_3, \dots, a^c_n\} \quad (2)$$

$$A^{cp} = \{a^{cp}_1, a^{cp}_2, a^{cp}_3, \dots, a^{cp}_n\} \quad (3)$$

where a^c represents an association between two Concepts-RU and a^{cp} represents an association between a Concepts-RU and a Competences-RU.

3.2 Skills Representation Unit

A Skills-RU is defined by the attributes: Id, Label. A Skills-RU is defined using set notation as:

$$s(A^{cps}) \quad (4)$$

where s is the Skills-RU and A^{cps} is the set of all the associations of s with its corresponding Competences-RU. Given this, the set A^{cps} is defined as:

$$A^{cps} = \{a^{cps}_1, a^{cps}_2, a^{cps}_3, \dots, a^{cps}_n\} \quad (5)$$

where a^{cps} represents an association between a Competences-RU and a Skills-RU.

There are multiple taxonomies that organise skills, all of them intend to be a reference for the use of the skills according to certain purposes or goals. A skill can be found in one or more levels of a taxonomy, according to their cognitive complexity, the skills can be grouped and ordered in a different way at the time of integrating a competency, since the complexity of a competency is determined by the complexity of the skill and the knowledge where it is used. However, hierarchies can be useful to organise the precedence of competences and skills in a learning process.

3.3 Competence Representation Unit

A Competences-RU associates, integrates and organise the tree types of units within an application context that can be measured and quantified, and is described by the following attributes: Id, Name, Description, Associated Concepts-RU, Skills-RU associations, Competences-RU associations. According to set notation a Competences-RU is defined as the following 3-tuple:

$$cp(c, A^{cps}, A^{cp}) \quad (6)$$

where cp is the Competences-RU, c is the Concepts-RU associated to cp , A^{cps} is a non-empty set of associations between cp and the Skills-RU that integrates it. A^{cp} is the non-empty set of associations between cp and another Competences-RU. The A^{cps} set corresponds to the definition given by expression 5, and A^{cp} is defined as follows:

$$A^{cp} = \{a^{cp}_1, a^{cp}_2, a^{cp}_3, \dots, a^{cp}_n\} \quad (7)$$

where a^{cp} represents an association between two Competences-RU whose properties are described later. Fig. 1 shows the structure of a competency and its relationships with other elements of Memory Map.

The structure of concepts determines the domain of knowledge where a skill is used or applied; act that produces a competence. A Competences-RU acquires the domain of knowledge from its Concepts-RU associated, meanwhile a skill is generic, it can be used repeatedly to produce different competences. The number of different competences that can be produced is equivalent to the number of different contexts in a MM.

3.4 Associations

The attributes of the associations between Competences-RU are: Id, Competences Domain Structure identifier, Successor Competences-RU, Order, Role. An a^{cp} association between two Competences-RU is given by the 4-tuple:

$$a^{cp}(cp_{pre}, cp_{suc}, d^{cp}, r^{cp}) \tag{8}$$

where cp_{pre} is the predecessor Competences-RU and cp_{suc} is the successor Competences-RU of the association, d^{cp} is the domain of competences that owns the association, and r^{cp} is the role of the association. The directionality of the associations is determined by cp_{pre} and cp_{suc} , and indicates that cp_{suc} is a sub-competency of cp_{pre} .

On the other hand, the association between Competences-RU and Skills-RU have the following attributes: Id, Associated Competences-RU, Associated Skills-RU, Order. An association a^{cps} between one Competences-RU and one Skills-RU is defined as:

$$a^{cps}(cp, s, r^s) \tag{9}$$

where cp and s represent the Competences-RU and the Skills-RU, respectively, which are involved in the association, and r^s represents the role of the association.

The Skills-RUs can be associated in different ways to Competences-RUs. The *role* in the associations permits to establish these differences and treat each type of association in a different way. It has been identified two *roles* to associate a skill with a Competences-RU: Application and integration.

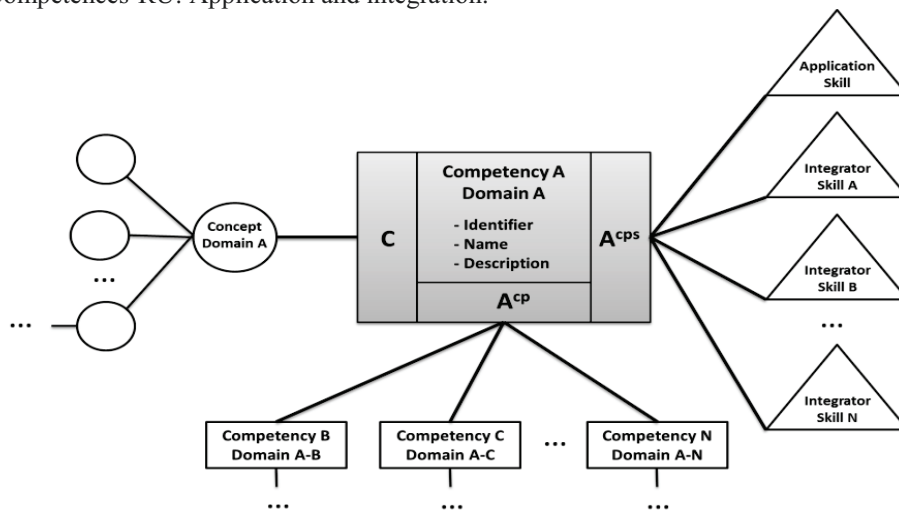


Fig. 1. Competency structure within the Memory Map.

The associations attributes between competences-RU and Concepts-RU are: Id, Associated Concepts-RU, Associated Competences-RU, Competence Domain. An association a^{cp} between one Concepts-RU and one Competences-RU is defined as:

$$a^{cp}(c, cp, d^{cp}) \tag{10}$$

where c is the Concepts-RU linked to the Competences-RU cp and d^{cp} is the Competence Domain defined by the relation between c and cp .

4 MM-Competences Properties

The highest level competence of a given structure determines the main goal of the whole structure, and the lowest or ‘atomic’ competences are similar to generic primitive skills. It has been identified three types of Competences-RU within the MM-Competences graph according to the nature of its associations: Root Nodes, Intermediate Nodes and Leaf Nodes, as seen in Fig. 2.

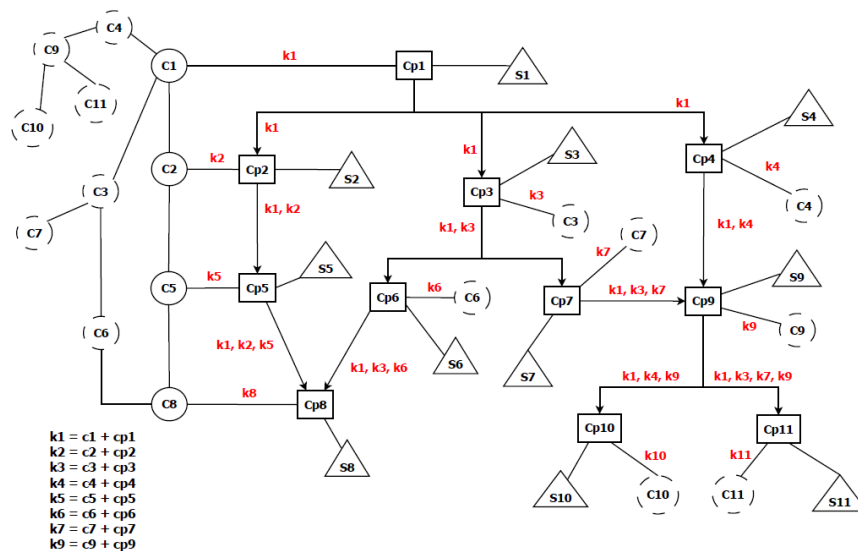


Fig. 2. Competence Domains generation and aggregation within the MM-Competences

Although the competences are aggregative and are organised hierarchically, the taxonomic structures formed by them are not fixed; on the contrary, they are dynamic, they are construed in terms of the present context. The associations are independent and can vary in terms of each attribute, including their direction, then, a Competences-RU can be the parent of another Competence-RU within a given structure and be its child in a different one; that is, they can reverse their hierarchical order in different structures corresponding to different domains of application. This dynamism in the relations created between competences comes from the existence of prior knowledge to the development of any competence.

4.1 Competence Domains

A Competence Domain originates from the association between a Competences-RU and a Concepts-RU, and represent the context for the development of a set of competences linked each other: a Competence Structure. Each association between a Competences-RU and a Concepts-RU creates a given Competence Domain which is propagated to all sub-competences of the related competences-RU, generating a Competence Structure. Competence Structures are aggregative, therefore, a Competence Structure can be a substructure of a larger one in a recursive way. This aggregative property is consistent with the aggregative nature of the knowledge in the brain, as the schools of constructivism propose. Fig. 2 shows the creation of Competence Domains and its aggregative property between Competences-RU for the formation of Competence Substructures. It also shows how a Competence Domain is generated through the associations between Concepts-RUs and Competences-RUs, and its propagation to the structures derived from the associations; it forms an aggregate of domains in each associations between the involved Competences-RU. The Concepts-RUs are represented by the circles, the Competences-RUs are represented by the rectangles and the Skills-RUs are represented by the triangles. Domains, highlighted in red, are represented by k_i .

5 MM-Competences Operations

MM-Competences, as part of the MM [14], has the goal of modelling the knowledge of a person or a machine in terms of competences, skills and concepts. To achieve this objective, the MM-Competences must have a set of operations that makes it functional and allows its manipulation. The Competences-RUs have the following general operations, which have been defined used set theory. There are also equivalent algorithmic definitions of the operations or inference processes, which are not presented here because of limitations of space.

Retrieving a Skills-RU. Let cp be a Competences-RU. According to the definition of a Competences-RU, cp can only be associated to a single application skill, a Skills-RU called $s(cp)$ is defined as:

$$s(cp) = \{a^{cps}(s) | a^{cps}(cp) = cp\} \quad (11)$$

where $a^{cps} \in s(A^{cps})$ and there only exists a single a^{cps} such that $a^{cps}(cp) = cp$.

Retrieving a Concepts-RU. Let c be a Concept-RU. According to (6), there exists one Concepts-RU for every Competences-RU. The Concepts-RU $c(cp)$ is given by:

$$c(cp) = cp(c) \quad (12)$$

where cp is the target competence. This means that even though c is a function of cp , it is only necessary to extract the Concept-RU from the definition of cp in order to obtain its value.

Retrieving Children Competences-RUs. Let cp be a Competences-RU. cp may require the execution of other Competences-RU, which are sub-competences of cp , in order to be achieved. The set of children Competences-RU of cp is given by:

$$U(cp) = \{a^{cp}(cp_{suc}) | a^{cp}(cp_{pre}) = cp\} \quad (13)$$

Where $a^{cp} \in cp(A^{cp})$, the set of all associations between cp and other Competences-RU, according to expression (7).

Retrieving Parent Competences-RUs. Let cp be a Competences-RU, cp may be required by a higher level Competences-RU in a structure of a given domain. The set of parent Competences-RUs of cp is given by the following relation:

$$U(cp) = \{a^{cp}(cp_{pre}) | a^{cp}(cp_{suc}) = cp\} \quad (14)$$

where $a^{cp} \in cp(A^{cp})$ and A^{cp} is defined as the set of all associations between cp and other Competences-RU's according to expression (7).

Retrieve the Competences-RU's associated to a Skills-RU. Let s be a Skills-RU, s is related to at least one Competences-RU; then the following relationship defines the set of all Competences-RU's $CP(s)$ related to a single Skills-RU s :

$$CP(s) = \{a^{cps}(cp) | a^{cps}(s) = s\} \quad (15)$$

where a^{cps} is an association between a Competences-RU cp and a Skills-RU s , according to expression (9).

6 Discussion

A Competence Structure is a set of Competences-RU that are associated together to form a structure represented by a directed graph, with a root competency and a sequence. The elements derived from the root competency are the subcompetences necessary to master it. Subcompetence structures are defined recursively as needed and end when they reach the leaf nodes. The Competence Structures forming the MM-Competences can be manipulated as instances of it; this means that they preserve the properties of it and its operations. A Competence Structure is uniquely identified by the root competence and its domain, this identifier represents the *Competence Domain* of the structure. Competence Domains are aggregative and propagate from the root competence to the children competences. MM-Competences can be defined as the structure that integrates all competence domains of a given person or machine. As the domains are aggregative, the MM-Competences can be extended according to the learning progress, a property consistent with the aggregative nature of knowledge.

7 Conclusions and Future Work

It has been presented a representation model of competences. The MM-Competences is capable of representing the nature and behaviour of the competences and its elements: concepts and skills, in an integrated, consistent and flexible way, and com-

pletely supports the aggregative nature of the knowledge. Also, it was shown how the MM-Competences is capable of modelling the contextualisation of competences through its relation with sets of concepts and skills.

This model is intended to be used into educational applications for the modelling of student's knowledge, skills and competences, through a computational implementation in progress.

References

1. Ackerman, P., "Determinants of individual differences during skill acquisition: Cognitive abilities and information processing", *Journal of experimental psychology: General, American Psychological Association*, 1988, vol. 117, pp. 288.
2. Anderson, J., "Acquisition of cognitive skill", *Psychological review, American Psychological Association*, 1982, vol. 89, pp. 369-406.
3. Conlan, O., Hampson, C., Peirce, N. and Kickmeier-Rust, M., "Realtime Knowledge Space Skill Assessment for Personalized Digital Educational Games", *2009 Ninth IEEE International Conference on Advanced Learning Technologies*, 2009, pp. 538-542.
4. Doignon, J.-P. and Falmagne, J.-C., *Knowledge Spaces*, Springer-Verlag, 1999.
5. Epstein, A., Schweinhart, L. and McAdoo, L., *Models of Early Childhood Education*, High/Scope Press, 1996.
6. Fischer, K. and Corrigan, R., "A skill approach to language development", *Language behavior in infancy and early childhood*, 1981, pp. 245-273.
7. Forehand, M., "Bloom's Taxonomy: From Emerging Perspectives on Learning, Teaching and Technology", *Synthesis*, 2010, pp. 1-9.
8. Hoffmann, M. H. W., Hampe, M., Muller, G., Bargstadt, H.-J., Heis, H.-U. and Schmitt, H., "Knowledge, skills, and competences: Descriptors for engineering education", *IEEE EDUCON 2010 Conference*, 2010, pp. 639-645.
9. Kaye, K., "The Development of Skills", *Academic, Press Inc.*, 1979, pp. 23-55.
10. Langworthy, A., "Skills and knowledge for the future: why universities must engage with their communities", *Technology*, 2004, pp. 1-17.
11. Lefebvre, B., Gauthier, G., Tadić, S., Huu Duc, T. and Achaba, H., "Competence Ontology for Domain Knowledge Dissemination and Retrieval", *Applied Artificial Intelligence*, 2005, vol. 19, pp. 845-859.
12. Paquette, G., "An Ontology and a Software Framework for Competency Modelling and Management Competency in an Instructional Engineering Method (MISA)", *Educational Technology & Society*, 2007, vol. 10, pp. 1-21.
13. Perez, L., "Curriculum Change and Competency-Based Approaches: a World Wide Perspective", *Education, XXXVII*, 2007.
14. Ramirez, C. and Valdes, B., "A general knowledge representation model for the acquisition of skills and concepts", *2010, International Journal of Software Science and Computational Intelligence*, 2(3), 1-20. ISSN: 1942-9045.
15. Thomas, J., "Varieties of Cognitive Skills: Taxonomies and Models of the Intellect", *Research for Better Schools, Inc., 444 North Third Street, Philadelphia, PA 19123*, 1972.
16. Wang, Y., "On Concept Algebra and Knowledge Representation", *5th IEEE International Conference on Cognitive Informatics, IEEE*, 2006, vol. 1, pp. 320-331.

Regular Papers

An Analysis of Web Services Attributes for Discovery Support

Héctor Jimenez Salazar, Christian Sánchez Sánchez,
Carlos Rodríguez Lucatero, and Arturo Wulfrano Luna Ramírez

Universidad Autónoma Metropolitana Unidad Cuajimalpa
División de Ciencias de la Comunicación y Diseño
Departamento de Tecnologías de la Información
Av. Constituyentes 1054, 3er piso, Col. Lomas Altas, Miguel Hidalgo, C.P. 11950,
Mexico D.F., Mexico
{hjimenez, csanchez, crodriguez, wluna} @correo.cua.uam.mx

Abstract. The number of Web Services, available over the Internet, has increased due the Web Services properties that allow them to be reused and to develop loosely coupled systems. Nevertheless, the current number of Services complicates the discovery, that is why it is necessary to improve the current mechanisms, matching and similarity, that support locating them. This approach focuses in analysing the Web Services attributes, which are contained in their descriptions. Our aim is to contribute to identify the role of attributes at the measurement of structural similarity function between web services and, through clustering, to develop resources for classifying and improving discovery. This proposal was tested through experiments over a collection of Service descriptions in WSDL which can be taken as an standard referent for such experiments. We show that our proposal outperforms the baseline taken as the best F -measure when the collection is represented by any attribute.

Keywords. Web services, discovery support.

1 Introduction

Currently, the number of Web Services (WS), available over the Internet, has increased due the Web services properties that allow them to be reused and to develop loosely coupled systems, as it can be noticed in available WS registries (or UDDIs) like Xmethods [1] and Seekda [2]. Web services are described by some languages, such as WSDL, OWL-S or WSMO.

In this paper we worked with WSDL, because is more common to find descriptions in this language than in the others, which are semantic languages. A WSDL[4] document defines services as collections of network ports, this allows the reuse of abstract definitions: messages, which are abstract descriptions of the data being exchanged, and port types, which are abstract collections of operations.

Nevertheless, the current number of services complicates the discovery taking into account that it is hard for human beings to read Web Service descriptions in WSDL and to search inside them.

Regarding to software systems, the accessible UDDI's just match keywords, contained in the user request, within the whole service descriptions, based on a business oriented classification. When a provider registers a service in an UDDI he/she chose a category of business, but also the searching through these categories, some times this is not enough because the classification depends on the provider expertise and/or criteria.

The problem of searching by means of keyword, and to syntactically relate them can produce ambiguity and additionally don't take advantage of the internal WSDL structure.

According to the Semantic Web, service matching can be done analyzing the description of what the services do: through its service profile in the OWL-S language [3]. Those profiles have functional and non-functional properties in order to describe the service. Functional properties describe Service functionalities through showing methods and their Inputs, Outputs, Preconditions and Effects (IOPE). On the other hand, non-functional properties provide extra information as quality of service, country, provider info and so on. Unfortunately, the most of the services does not have a semantic description yet; they just have syntactic descriptions in WSDL.

Because of this problem and with the intention of providing semantics to the descriptions, some approaches like [5], [6] and [7] have been focused on relating (semi-automatically) parameter names and concepts in ontologies, despite of there are few semantic descriptions of services and ontologies.

For those reasons it is necessary: a) to exploit the embedded structure (attributes) of the Web Service description, b) to extract the description semantics, c) to identify attributes that help to define a suitable structural similarity function (matching) and d) to use semantics for developing resources in order to support the WS discovery. In this paper we focus the first point with the purpose of comparing attribute-based representation of WSDL and, then, identify the role of attributes on matching for discovery.

The remainder of this paper is organized as follows: Section 2 contains the state of the art, Section 3 shows how was gotten a non-structured document from a WSDL, Section 4 describes the experiment concerned to the influence of the attributes for representing WSDL, and finally in Section 5 concludes this work.

2 Matching and Clustering of WSDL

Concerning the extraction of the WSDL descriptions embedded semantics, using Natural Language Processing and Information Retrieval tools as well as the things regarding structural matching of the Web Services, it has been developed some approaches, as we will show in the following paragraphs.

Stroulia et al.[8] preprocess information from several WSDL descriptions in order to apply Information Retrieval methods to determine the similarity of the service descriptions, on the other hand also take into account the structure of WSDL documents to determine the similarity of two services, starting by

comparing the data types, and then compare the messages and their parameters to finally finish the execution of the operation. They compare these elements because they have the intuition that the names of these attributes usually reflect the semantics embedded service capabilities.

Using Information Retrieval, there is an approach proposed by Hao et al. [9]. Here the similarity function that help to rank the services, mainly takes into account three aspects:

1. The relevance of the service: according to the terms it shares with the request.
2. The importance of service: through the services most used by others.
3. The service connectivity: analyzing the similarity of the XML tree of the description.

However, in these approaches it has not been really proved that these attributes or aspects, going farther than the intuition, are the more important for obtaining the structural similarity function. On the other side, they use WordNet for obtaining the synonyms, hypernyms, hyponyms, nearer concepts and steems but due to the WordNet generality it can deviate to the true of those domains that contribute to find the description similarity.

Also, seeking to identify major similarity between the descriptions and accelerate the discovery, there are some approaches that seek to categorize Web services, as in the case of Bruno et al. [11], that in addition to using Information Retrieval techniques, they sort the services in classes (predetermined by them) using the concept extracted from the descriptions that vectorize, so services can be classified using support vector machines in order to sort them into a lattice, using IS-A relationships. Unfortunately, these approaches have not scaled up on several collections of WSDL.

Liang et al. [12] show another approach where categorization is used. Their approach is focused on finding similar services working with semantic and syntactic descriptions. They categorized WS schemes to match WS that can operate in heterogeneous domain ontologies. Having an upper ontology and using its OnEx-Cat tool, they can determine when a Web Service is a possible replacement by another. They considered the categorization of ontologies as a term categorization search problem, which also it is similar to the task of document classification. As in the previous approaches, in these works it is not done attribute selection for supporting a good classification.

Working under the hypothesis that the automatically generated clusters will be able to suggest similar services, some authors have grouped web services information, obtained from WSDL and records (information registered by the creator in UDDIs). For instance, Fan et al. [13] joined the recording service information, the service documentation and their operations, and started to form groups using the Hierarchical Agglomerative Cluster (HAC) algorithm and the Jaccard similarity measure. They found a very big amount of noise in the formed groups. By this reason they concluded that many descriptions lacked of documentation, and that there were not enough information in the recorded information for distinguish it from others in the clustering.

Similarly, Dong et al. [14] created an algorithm to group in concepts having semantic meaning, the parameter names of the Web Services operations, by means of which, the similarity of the input and output operations of the services can be determined. Their algorithm works as sequence of refinements of the classical agglomerative clustering, that step by step associates groups that share terms that are nearly related and meet the cohesive property. However, when the groups are found by the algorithm they can be contaminated by noisy information, and then, for trying to improve the obtained results, this noise is eliminated by using some heuristics. For verifying similarity between two services, his algorithm mix the information concerning the services, operations, inputs, outputs, documentation and recording, and conclude that the performance improvement is not greatly enhanced by simple combination of the aforementioned information.

The found problems for classifying WS description documents are: there are not enough efforts to categorize them, and the existent descriptions are keep updated by its providers with different level of background and expertise. Web Services are available dynamically and the classes become obsolete rapidly. As is known the providers do not share the same vocabularies and the most of the times use n-grams or compound words.

As we have seen, all tasks related to discovering and/or reusing web services rely on similarity, therefore representation of WS is an important topic. Furthermore, this problem demands support from domain dependent resources and/or to use adequately the attributes of WS. Motivated by this conclusion we want to determine what WSDL attributes enable us to enhance the clustering of the Web Services, making easier the discovering task by giving hints about the structural similarity function of the elements.

3 Getting a Non-structured File from WSDL

Description of a web service by WSDL constitutes a reusable binding. Hence, a WSDL document uses the following elements in the definition of network services:

- **Types**- a container for data type definitions using some type system.
- **Message**- an abstract typed definition of the data being communicated (**parameters**).
- **Operation**- an abstract description of an action supported by the service.
- **Documentation**- description in natural language to describe the service and/or its operations.
- **Port Type**- an abstract set of operations supported by one or more endpoints.
- **Binding**- a concrete protocol and data format specification for a particular port type.
- **Port**- a single endpoint defined as a combination of a binding and a network address.
- **Service**- a collection of related endpoints.

According to Stroulia et al. [8] information like the services' names associated to the methods, parameters and data types is useful, because they reflect the semantics of the underlying capabilities.

Our approach considers a preprocessing step before performing the analysis which is explained in Section 4.

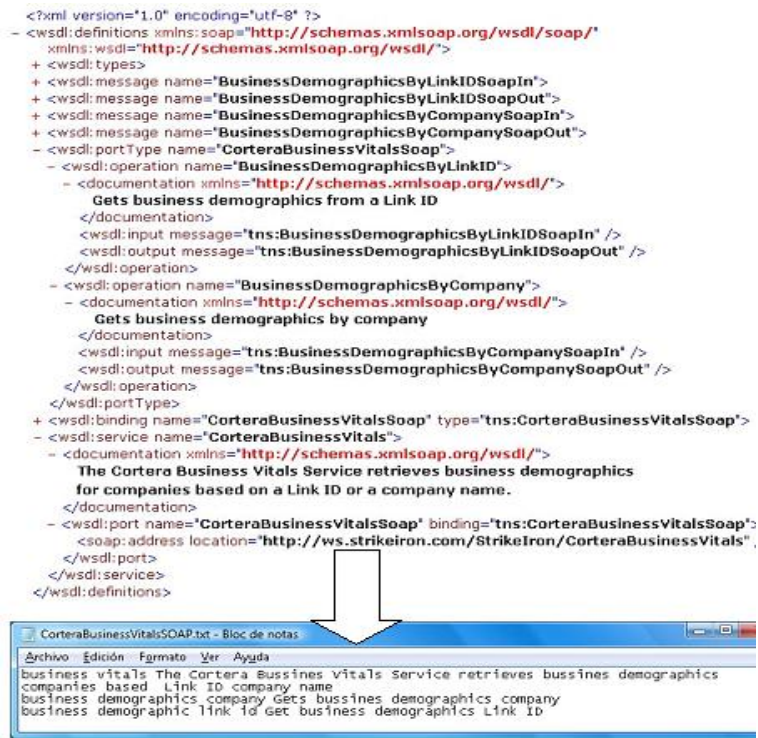


Fig. 1. Preprocessing of a non structured short WSDL document.

The preprocessing consists of extraction of the words contained in the attributes of the descriptions of Web services. The attributes considered for the analysis were the following: Name of the Service, Name of the Operations, Documentation, Name of the Messages, Name of the parameters. Since the documentation is a natural language description, the extraction of words was done by taking all the words contained and separated by spaces.

For the case of other attributes, obtaining words are performed as follows.

- **Step 1.-** Analyzing Names. Identifying the names formed by one known word or composed words, using a dictionary (WordNet), this is done verifying if the name is contained in the dictionary.

- **Step 2.-** Breaking names into single words. For finding a set of terms related to a name, the next alternative procedures are proposed.
 - a) Obtaining all the possible substrings from the names: This method can be applied to every composed word name.
 - For each name a number k of substring are gotten, where $k = \sum_{i=0}^l (l - i)$ and l is the length of the name.
 - When we have all the substrings then we identify which of them are known words, as it is done in Step 1, with the purpose of relating them to each name.
 - b) Obtaining strings which starting prefix is a capital letter. This string treatment is applied to every composed word name that starts with a capital letter and is formed by other capital and small letters.
 - The name is broken into k substrings, where $k = C$ is the number of capital letters contained in the string. The name is cut from each contained capital letter until finding other capital letter or the end of the name.
 - When we have all the substrings then we identify which of them are known words, as it is done in Step 1, with the purpose of relate them to each name.
 - c) Obtaining strings that are separated by special characters This string treatment can be applied to every composed word name that contains special characters; where $(- =:,)$ were all the characters found in the used collection to separate words.
 - The name is broken into k substrings, where $k = S + 1$ if the prefix of the name is not a special character, or $k = S$ if it is. The name is cut from the beginning or where is found the special character +1 until finding other special character or the end of the name.
 - When we have all the substrings then we identify which of them are known words, as it is done in Step 1, with the purpose of relate them to each name.
- **Step 3.-** Deleting stop words. The list of all deleted words, *articles, prepositions, pronouns, etc.*, can be found in [lextek](http://www.lextek.com)¹.

4 Analyzing the Description of WS

As we have said the help for classifying and searching WS has an important implication for the information resources at the existing WS. It does not only deal with the resource compilation helping to describe most of the WS, besides, based on the volatility of the web services, it is convenient that service classes and derived resources being dynamic. So, we propose a methodology in order to build up elements of enhancement for WSDL classification.

The analysis, we will show here, gives importance to WSDL components through the clustering supported in some attributes alone or combined. First of

¹ <http://www.lextek.com/manuals/onix/stopwords1.html>.

all, we measure the performance of each attribute used to represent the whole collection of WSDL: which of them obtains the better clustering or, saying, which of them makes the best representation of WSDL. From these results we analyze some attribute combinations and its meaning aiming to improve the clustering.

4.1 Data Collection

The WSDL document collection that we selected, for the sake of proving this approach, and taking into account there are not few of them, can be found at ASSAM² [10]. That collection is composed by real Web services description documents, obtained from Salcentral and Xmethods. The WSDL documents are organized into a class hierarchy, that in some cases have subclasses with at most two levels of depth. The collection have 814 WSDL distributed in 26 classes. However, to prove our approach we needed to make two modifications:

- flatten classes (subclasses eliminate from the main classes),
- select WSDL documents which could be extracted at least one word (recognized by WordNet) for each attribute (name of service and methods, documentation, messages and parameters) that is part of the description of the WSDL document.

Therefore we gather the collection, which was reduced to 22 classes with 203 WSDL. The next table shows some data about the gathering collection:

Feature	Value
# Classes	22
# WSDL	203
<i>Vocabulary</i>	2,829
WSDL × Class (avg)	9.2
Terms × identifier (avg)	2.7
Terms × documentation (avg)	26.34

4.2 Experiments

The experiments were carried out considering:

Purpose. They focused to know the attribute quality given in a description of web services.

Testing. The analysis of hypothesis on the role of each attribute, given by the set of values that experiments take, is made through a combination of attributes. By instance, the hypothesis: **Name** attribute (of a method in the WS) may be part of a nominal phrase which is more precise when includes the **Parameter** attribute (names of); it is tested by means of the combination of those attributes.

Preprocessing. The preprocessing previous to the clustering:

² <http://www.andreas-hess.info/projects/annotator/index.html>.

1. selection of attribute(s) for representation,
2. lowerizing and deleting term repetition, and
3. determining a percentage of terms included in the set of values of the attribute(s) through the ranking given by a term selection technique.

It is important to address that we work with a bag of words representation. This decision is based on the low frequencies of terms in descriptions which is not proper for the Vector Space Model [19]. So, we use the Jaccard coefficient to measure similarity between instances.

Attributes. Referring to the list of components of WS (see Sec. 3), the attributes taken into account at the experiments were the following:

- **Name** (of operation): terms taken from the identifier.
- **Parameters**: they are given by type names.
- **Documentation**: this is the only attribute regarding as a phrase of natural language.
- **Message**: terms provided by the procedure call.
- **All**: in this case all the above attributes are joined to represent each WS.

WS representation. Decision on instance representation required to test some Term Selection Techniques (TST). Since our purpose was the comparison among the attributes we did not be exhaustive on this point. Three TST were used: DF (document frequency) gives greater importance to terms that appear in more instances [15]; TP (transition point), the importance of a term is high when its frequency closes to the frequency which divides the vocabulary into the high and low frequencies [16]; and DEN (density) term importance is high if it contributes to form few classes [17]. From these TST we selected the one that obtained the best clustering performance was the best at development phase of the experiment. In the experiments we take percentages from 10 to 100 to representing each instance. Next step was the clustering of the represented instances.

Clustering. We are using a method of the $K - NN$ family, k -star [18], a divisive and non hierarchical method. The resulting clusters are evaluated by a metric supported on precision and recall measures (F).

4.3 Results

A prime measurement was determined applying directly the clustering procedure to the collection without term selection. Next table shows F values of each one of five attributes.

Attribute	F
Name	0.23
Parameter	0.23
Documentation	0.21
Message	0.22
All	0.23

All the attributes gave a poor performance: $F \in [0.21, 0.23]$. So, we used the DEN-TST in order to put into relevance the strength of attributes. Figure 2 shows the curve for each attribute: at horizontal axis, percentages of terms according to the Den-TST, starting from 10 till 100 percent of terms; vertical axis grades the F value determined by the clustering of the collection represented with the given percentage.

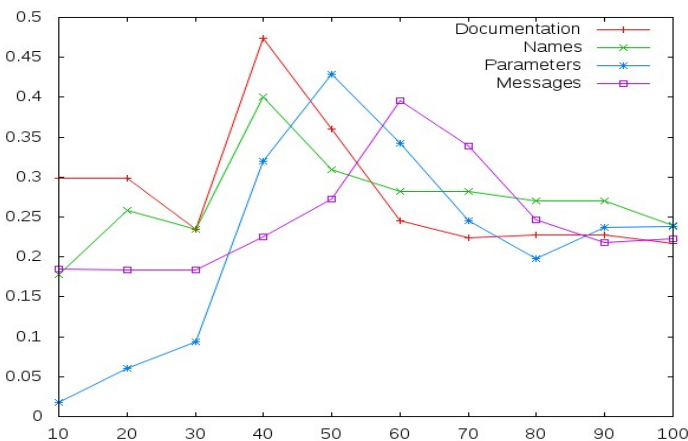


Fig. 2. Clustering performance of WS with four attributes.

Documentation attribute obtained the best performance, $F = 0.47$. The result is understood as: it is easier the matching between sets of terms forming sentences than the sets given by terms extracted from heterogeneous structures; i.e. phrases, or structured identifiers (types). However, this result has no great relevance due to that most of WS lacking of documentation. Rather we will consider the F value of **Documentation** as a baseline.

Our approach was to trying to “reformulate” a documentation from the other attributes. Three hypothesis were raised:

1. Attributes **Name** and **Parameter** have a role on documentation; i.e. they might constitute a paraphrase of some kind: *(To do) name (use) par₁, ... par_n*.
2. Since **Message** attribute has information about the procedure call, it might, in combination with **Name**, both would contribute to some paraphrase of documentation, namely: *Name (is used as) message*.
3. As well we might hope the combination **Name**, **Message** and **Parameter** could paraphrase some kind of documentation: *(The) name (operation takes) par₁, ... par_n (to proceed as) message*.

Therefore, to test the behavior of such combinations of attributes the clustering of collection was carried out. Figure 3 depicts the F values of the three combinations and they are confronted with F values of **Documentation**.

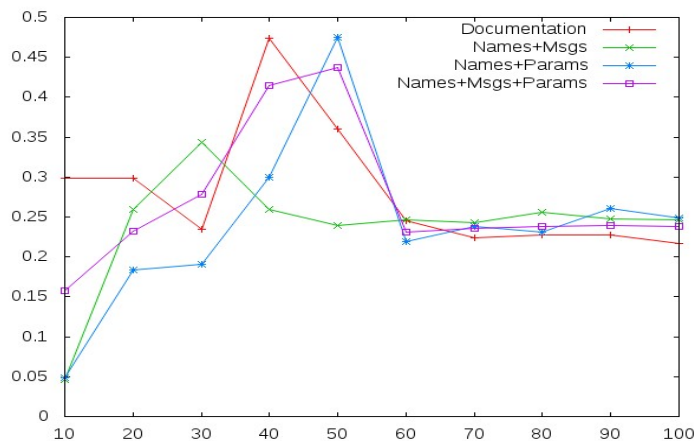


Fig. 3. Clustering performance of WS by attribute combination.

According to this result we claim that, in this collection, **Name-Parameter** ($F = 0.47$) combination may substitute **Documentation**. On the other side, the great difference of performance on the combinations **Name-Message** and **Name-Message-Parameter** is due to the noise provided by the inclusion of non related terms, in the second case.

Then, we can test the “virtual documentation” applying this representation to WS which have not **Documentation**. We used the complete collection, and applied the clustering procedure representing each WS by attributes **Documentation** and **Name-Parameter** combination. The following table summarizes some features of the collection:

Feature	Value
# Classes	25
# WSDL (with doc.)	213
# WSDL (without doc.)	226
$ Vocabulary $	3,228
WSDL \times Class (avg/stdev)	17.5/20.7

The result of the best clustering for those attributes were $F = 0.46$ for **Documentation** and $F = 0.53$ for **Name-Parameters**, which shows a clear en-

hancement for the WS representation through the proposed attribute combination `Name-Parameters`.

5 Remarks and Conclusions

On the context of comparing WS, we analyzed a better way to represent web services instead of a direct use of their attributes. Even though we can select the attribute which gives the best performance for comparing WS, the experiments carried out show the usefulness of combining attributes. We have found that, in the used collection of WS, `Name-Parameters` attribute combination outperforms `Documentation` attribute; which gives the better representation alone.

This result has relevance due to most of WS lack of documentation. Just 51% of the WS contained in the original collection `Documentation` is missing. This fact is also observed in many of the WS retrieved from repositories. For example, Fan et al. [13] reported that almost half the services do not have any documentation for any of the operations supported. Thus, for WS discovery, the lack of `Documentation` is an important problem to be tackled. In this work we have showed that `Documentation` attribute can be “reformulated” by the combination of other two: `Name` and `Parameters`, obtaining a better performance than the one of `Documentation`; $F = 0.47$ and $F = 0.53$ respectively.

Since the representation of data impacts the searching, classification and clustering of WS, it is important to continue the improvement of WS representation. By instance, it is necessary to incorporate, in some way, the rest of attributes contained in the WSDL description.

We have seen certain behaviour of WSDL attributes on a particular public collection, but it is necessary to support such behaviour using more supervised WSDL collections. In spite of the dynamic of WS on the web, it is imperative to consider standard supervised collections of WSDL aiming to compare diverse approaches on the task of discovery of web services.

Acknowledgments. We wish to thank to Conacyt-México by the given support to project grant Nr. 153315.

References

1. Xmethods, <http://www.xmethods.net>.
2. Seekda, <http://webservices.seekda.com/>.
3. Owl-s, <http://www.w3.org/Submission/OWL-S/>.
4. Wsdl, <http://www.w3.org/TR/wsdl>.
5. Guo, H.; Ivan, A.; Akkiraju, R. & Goodwin, R.: Learning ontologies to improve the quality of automatic web service matching. In: Proceedings of the 16th International Conference on World Wide Web, New York, NY, USA, ACM, pp. 1241–1242 (2007)
6. Qu, C.; Zimmermann, F.; Kumpf, K.; Kamuzinzi, R.; Ledent, V. & Herzog, R.: Semantics-Enabled Service Discovery Framework in the SIMDAT Pharma Grid. IEEE Transactions on Information Technology in Biomedicine. pp. 182–190 (2008)

7. Jaeger, M.C.; Rojec-Goldmann, G.; Liebetrueth, C.; Mühl, G. & Geihs, K.: Ranked matching for service descriptions using owl-s. In: KiVS (2005)
8. Stroulia E. & Wang Y.: Structural and semantic matching for assessing web-service similarity. *International Journal of Cooperative Information Systems*. vol. 14, pp. 407–437 (2005)
9. Hao, Y.; Zhang, Y. & Cao J.: Web services discovery and rank: An information retrieval approach, *Future Generation Computer Systems*, vol.26 No.8, pp. 1053–1062 (2010)
10. Hess, A.; Johnston, E. & Kushmerick, N.: ASSAM: A Tool for Semi-automatically Annotating Semantic Web Services. *Lecture Notes in Computer Science*, Vol. 3298, pp. 320–334 (2004)
11. Bruno, M.; Canfora, G.; Di Penta, M. & Scognamiglio, R.: An Approach to support Web Service Classification and Annotation. In: *Proceedings of IEEE International Conference on e-Technology, Hong Kong, China*, pp. 138–143 (2005)
12. Liang, Qianhui Althea & Lam, Herman: Web Service Matching by Ontology Instance Categorization. scc. vol. 1, In: *IEEE International Conference on Services Computing*, pp. 202–209 (2008)
13. Fan, J. & Kambhampati: S.: A snapshot of public web services. *SIGMOD Records*, **34**(1) 24–32 (2005)
14. Dong, X.; Halevy, A.; Madhavan, J.; Nemes, E. & Zhang, J.: Similarity search for web services. In: *Proceedings of VLDB*, pp. 372–383 (2004)
15. Yang, Y. & Pedersen, P.: A Comparative Study on Feature Selection in Text Categorization. In: *Proc. of International Conference on Machine Learning*, pp. 412–420 (1997)
16. Jiménez-Salazar, H.; Pinto, D.& Rosso, P.: Uso del punto de transición en la selección de términos índice para agrupamiento de textos cortos. *Procesamiento del Lenguaje Natural*, **35**, España, pp. 416–421 (2005)
17. Jiménez-Salazar, H.; Sánchez, C.; Rodríguez, C. & Luna, A.: Modelación léxico semántica de descripciones de servicios web, In: 8o. *Taller de Tecnologías del Lenguaje Humano, Tonatzintla* (2011)
18. Shin, K. & Han, Y.: Fast Clustering Algorithm for Information Organization, *Lecture Notes in Computer Science*, vol. 2588, pp. 619–622 (2003)
19. van Rijsbergen: *Information Retrieval*. University of Glasgow (1979)

Analysis of the Quotation Corpus of the Russian Wiktionary

Alexander Smirnov¹, Tatiana Levashova¹, Alexey Karpov^{1,2}, Irina Kipyatkova^{1,2},
Andrey Ronzhin^{1,2}, Andrew Krizhanovsky³, and Nataly Krizhanovsky³

¹ St.Petersburg Institute for Informatics and Automation
of the Russian Academy of Sciences, Russia,

² Saint-Petersburg State University, Department of Phonetics, Russia

³ Institute of Applied Mathematical Research of the Karelian Research Centre
of the Russian Academy of Sciences, Russia

{smir, tatiana.levashova}@iias.spb.su,
{karpov, kipyatkova}@iias.spb.su
andrew.krizhanovsky@gmail.com, nataly@krc.karelia.ru

Abstract. The quantitative evaluation of quotations in the Russian Wiktionary was performed using the developed Wiktionary parser. It was found that the number of quotations in the dictionary is growing fast (51.5 thousands in 2011, 62 thousands in 2012). These quotations were extracted and saved in the relational database of a machine-readable dictionary. For this database, tables related to the quotations were designed. A histogram of distribution of quotations of literary works written in different years was built. It was made an attempt to explain the characteristics of the histogram by associating it with the years of the most popular and cited (in the Russian Wiktionary) writers of the nineteenth century. It was found that more than one-third of all the quotations (the example sentences) contained in the Russian Wiktionary are taken by the editors of a Wiktionary entry from the Russian National Corpus.

Keywords. Wiktionary, corpus, quotations, literary works.

1 Introduction

The progress of computer technologies provides a basis for a new type of dictionaries. This type is an online dictionary, where any interested person can take part in the dictionary development. On the one hand, this way of organizing collective work provides obvious advantages (high intensities of the work, the possibility of online discussion and correction of the articles at any stage of the work); on the other hand, there is a high possibility that some gaps can be presented in the source material and some gaps can be found in the dictionary itself.

One of the possible solutions to the problem of bridging the gaps is to develop a special software tool that can analyze the online dictionary at any stage of the development. Some possible solutions to this problem will be presented in this paper

based on an analysis of the quotations of some literary works contained in the Russian Wiktionary that is a good example of an online dictionary.

The research presented in this paper aims at two purposes: (1) construction of a quotation corpus from the online dictionary, (2) an analysis of the chronological distribution of the quotation corpus within 1750–2012 years. The choice of this period is caused by the fact that the period includes years with more than 10 quotations which refer to this year in the dictionary.

The Wiktionary is a multilingual and multifunctional computer dictionary which combines thesaurus, lexicon and phraseological dictionary. The Wiktionary combines a glossary and explanatory, grammatical, etymological, and translation dictionaries. The Wiktionary contains not only concepts' definitions, semantically related concepts (synonyms, hypernyms, etc.), and multilingual translations, but also the pronunciations (phonetic transcriptions, audio files), hyphenations, etymologies, quotations, parallel texts (quotations with translations), and illustrations (to illustrate meaning of the words).

The Wiktionary data are used:

- In translation:
 - *Dictionary-based* machine translation implemented in the Pandictionary and the Panlingual Translator system [14]. Pandictionary is a sense-distinguished translation dictionary compiled from Wiktionaries and more than 600 machine-readable bilingual dictionaries. Panlingual Translator system uses a lemmatic encoding of the original text as a form of human-aid translation;
 - Translation of the taxonomically organized labels in web-based multilingual resources (the folktale domain); translation is based on multilingual lexical entries and semantic categories of English, German and Hungarian language versions of Wiktionary [2];
 - *Machine translation* between Dutch and Afrikaans [10].
- In the text parsing system NULEX, where some Wiktionary data (verb tense) were integrated with WordNet and VerbNet [8];
- In a *speech recognition and speech synthesis* where the Wiktionary is a basis for the rapid pronunciation dictionary creation [12], [13];
- In *ontology matching* [7];
- For extraction of semantic relations [11];
- For knowledge base construction, e.g. Concept Net¹.

The paper [2] discussed several potential shortcomings of the Wiktionary. The Wiktionary may lack basic information or be of poor quality as the result of collaborative work performed by volunteers. The Wiktionary is formatted in a lightweight mark-up language and wiki format is often applied in an inconsistent manner within one dictionary or across different language versions of Wiktionary, which makes the extraction of structured lexical information a challenging task [3].

The advantages of the Wiktionary for the present research are a huge volume of data and a wide variety of the lexicographical material. An analysis of the German and English language editions of Wiktionary [6], [9] has shown that the sizes of these

¹ See <http://conceptnet5.media.mit.edu>.

dictionaries are close to or exceeds thesauri in corresponding languages. For example, the sizes of the German Wiktionary and the German thesauri *GermaNet* and *OpenThesaurus* are comparable, whereas the size of the English Wiktionary exceeds the size of the English thesaurus *WordNet*. Any freely available dictionaries in Russian (in the public domain) have not been found. It can be suggested, that the size of Russian Wiktionary is enough for the purposes of the present research.

A word's definition in the Wiktionary is accompanied by quotations, which illustrate the meaning by the surrounding context. The quotations can include references to a source (book, newspaper, blog) with the date of its publication or writing.

The analysis of the dates of literary works, which are used as a source of quotations, is the goal of the paper. The experiments were carried out on the corpus of quotations build on the basis of the Russian Wiktionary. The database of quotation corpus is a part of the machine-readable Wiktionary, which is an open-source project.²

2 Framework of Machine-Readable Wiktionary

The conception of the machine-readable Wiktionary is flexible in relation to input data, but it is strict and formal to output data.

Input data. This conception suggests that different wiktionaries can have different article structures (e.g. different names of the article sections and different order of sections), which must be taken into account by a wiktionary-parser. Moreover, even within one Wiktionary, the structure of an article can change with time as new sections appear and templates vary and change. Therefore there is need for a flexible and modular framework in order to parse so much "live" and various wiktionaries (Fig. 1). The specific properties of different wiktionaries are taken into account in the submodules "*ruwikt*" and "*enwikt*" in the module "*Data extraction*" in Fig. 1.

Output data. The data extracted from a Wiktionary are stored in the database of the machine-readable dictionary. The result databases filled by the parser have identical structure independent on the source wiktionary. This ensures compatibility of different machine-readable wiktionaries with external applications.

The framework can be extended with new wiktionaries since many parts of the parser have been already developed and do not depend on a specific wiktionary. These parts are as follows:

- Common application programming interface (API) to the source databases (*input data*).
- "Common part" of the module "Data extraction" (Fig. 1). It contains (1) language codes in accordance with the international standard for language codes ISO 639, (2) names of languages in English and in Russian; now the parser recognizes 370 languages and codes in the English Wiktionary and 274 in the Russian Wiktionary.

² The machine-readable Wiktionary is available at <http://code.google.com/p/wikokit/>.

–Common API to the result databases of the machine-readable wiktionaries (*output data*).

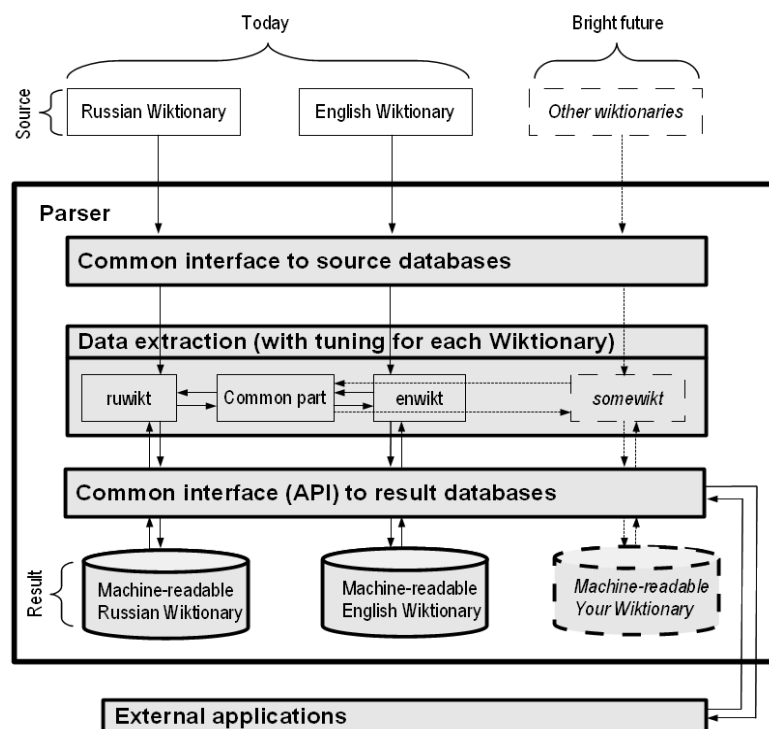


Fig. 1. Machine-readable Wiktionary: the framework.

3 Architecture of the Database of Quotation Corpus

The database of the quotation corpus is a part of the relational database of the machine-readable Wiktionary presented in the paper [5].

The following fields of the quotation template are recognized and added to the database during the extraction of semistructured data from the Wiktionary by the parser (Fig. 2):

- The text of the quotation (stored in the field *text* of the table *quote*).
- The translation into Russian (the table *quot_translation*).
- The transcription of the quotation (the table *quot_transcription* is reserved for the English Wiktionary, it is not used in the Russian Wiktionary).
- Information about a quotation reference is collected in the table *quot_ref*.

This table comprises the following fields:

- Title of the source (the field *title* of the table *quot_ref*).
- Author of the source (the table *quot_author*).
- Publisher (the table *quot_publisher*).

- Publication date (the table *quot_year*).
- Name of the resource or corpus, where the quotation is taken from (the table *quot_source*).

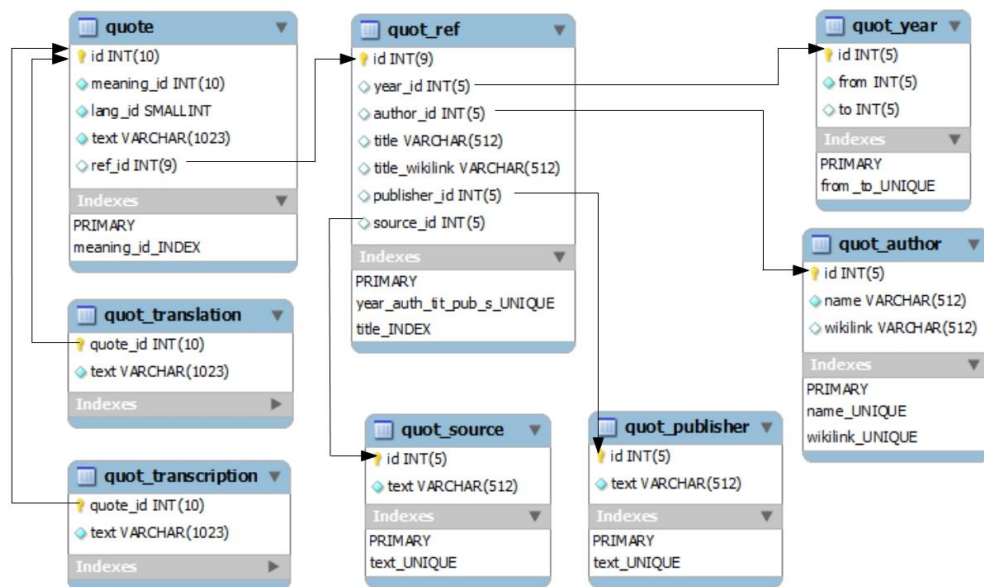


Fig. 2. Tables and relations relevant to quotations in the database of the machine-readable Wiktionary.

4 Database Queries

Various SQL-queries to the database are supported (Fig. 2). For example, using a few queries, one can solve the following search task: *get a list of quotations in English, which refer to books written during more than one year.*³

This task is solved step-by-step:

- 1) Get a list of quotations in English. As of March 2012, there are 1355 quotes in English in the Russian Wiktionary.
- 2) Get a sublist of quotations with non-empty reference (a source). There are 222 quotations where “*ref_id*” is not NULL in the table “*quote*”.
- 3) Get a sublist of quotations, which contain a date in the reference. There are 123 quotations with years.

³ See these queries: http://code.google.com/p/wikokit/wiki/MRDQuote#SQL_queries

- 4) Get a list of quotations, which contain a range of years in the reference. I.e., the value of the field “to” is greater than “from” in the table *quot_year* (Fig. 2). As the result, seven quotes (Table 1) were found.

The column “entry” in Table 1 contains the headword of a Wiktionary article. The quotation is placed in the row below and the word in question is marked by **bold** font in the quote. If there is a translation of this quote into the Russian then it is presented in the next row below. The author name, the title of the source book and the publication (or writing) date (in years) are given in the columns “author”, “title”, “from”, “to”, respectively.

Table 1. English quotations from the Russian Wiktionary, which refer to books written during more than one year.

N	Entry	Author	Title	From	To
1	Moscow ⁴	Andrei Platonov	The Ethereal Tract	1926	1927
	Moscow awakened and screamed with trams. ... The summer sun rejoiced over the full-blooded land, and two men appeared before the gaze of a new Moscow — a wonderful city of powerful culture, stubborn labor and intelligent happiness.				
	Москва проснулась и завизжала трамваями. ... Летнее солнце ликovalo над полнокровной землёй, и взорам двух людей предстала новая Москва — чудесный город могущественной культуры, упрямого труда и умного счастья.				
2	cacophony	H. P. Lovecraft	Herbert West: Reanimator	1921	1922
	Not more unutterable could have been the chaos of hellish sound if the pit itself had opened to release the agony of the damned, for in one inconceivable cacophony was centered all the supernal terror and unnatural despair of animate nature.				
3	hoarder	–	[Central News autocue data.] 3623 s-units.	1985	1994
	The picture was owned by antiques hoarder Ronnie Summerfield who died three years ago leaving a collection valued at millions of pounds.				
4	hoarder	–	The Economist. 3341 s-units.	1985	1994
	The biggest official gold hoarder by far is America, which holds 27,9 % of the world’s central-bank gold reserves.				
5	order	Charles Dickens	Oliver Twist	1837	1839
	In pursuance of this determination, little Oliver, to his excessive astonishment, was released from bondage, and ordered to put himself into a clean shirt.				
6	order	Charles Dickens	Oliver Twist	1837	1839
	Oliver was ordered into instant confinement; and a bill was next morning pasted on the outside of the gate, offering a reward of five pounds to anybody who would take Oliver Twist off the hands of the parish.				
7	practitioner	Charles Dickens	The Posthumous Papers of the Pickwick Club	1836	1837
	These sequestered nooks are the public offices of the legal profession, where writs are issued, judgments signed, declarations filed, and numerous other ingenious machines put in motion for the torture and torment of His Majesty’s liege subjects, and the comfort and emolument of the practitioners of the law.				

⁴ See entry “Moscow” in the Russian Wiktionary: <http://ru.wiktionary.org/wiki/Moscow>

5 Experiments

5.1 Corpus of Quotations

The corpus of quotations was built on the basis of the Russian edition of Wiktionary as of March 25, 2012. It was constructed by means of the developed Wiktionary parser [5]. The corpus includes 62 thousand quotations (51.5 thousand in 2011). It is important that 52 thousand quotations (84% of the whole number of quotations) are occurred in the explanations for Russian words (82% in 2011).

In the Russian Wiktionary, 23.8 thousand quotations (38.35% of the whole number) have a reference to the source (17 thousand quotations with references in 2011, i.e. 33%). The main source of quotations in the Russian Wiktionary is the *Russian National Corpus* [1]. There are 94.15% quotations (of the whole number of quotations with references) which refer to the Russian National Corpus.

5.2 Publication Date: Analysis and Hypothesis

In this study publication dates indicated in the sources of quotations are under investigation. These dates are stored in the table *quot_year*, discussed in Section 3. The table *quot_year* contains two fields “*from*” and “*to*” of integer type (Fig. 2) indicating the years of source publication. If the work was published or written during one year, then both fields have equal values. The number of unique pairs (start year “*from*”, finish year “*to*”) is 862 in the Russian Wiktionary (it equals to the number of records in the table *quot_year*).

In order to calculate a number of quotations for each year the algorithm similar to the well-known game “Tetris” was used (Fig. 3). The algorithm traverses all quotations; if a quotation contains a year or a range of years then the number of quotations for years in this range are incremented.

For example, there are years 1926–1927 and 1929 in the following quotations of entries “Moscow” and “medal” in the Russian Wiktionary:

–**Moscow** awakened and screamed with trams. ... The summer sun rejoiced over the full-blooded land, and two men appeared before the gaze of a new **Moscow** — a wonderful city of powerful culture, stubborn labor and intelligent happiness. *Andrei Platonov*. “The Ethereal Tract”. **1926-1927**

–This is a very brave young man. He has been proposed for the silver **medal** of valor. *Ernest Miller Hemingway*. “Farewell to Arms”. **1929**

These quotations are presented in Fig. 3a in the form of bricks on the abscissa in 1929 and in the range 1926–1927. Let’s suppose that during the traversal a quotation from the source written in 1924–1926 has been found. Then the value of the histogram at 1926 in Fig. 3b is two (quotations) and at 1924–1925 is one (quotation).

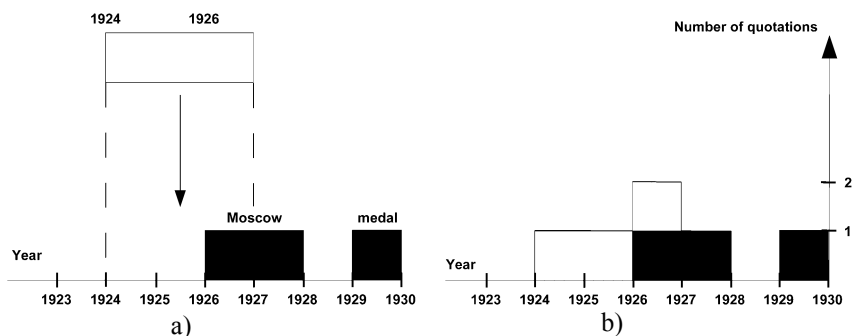


Fig. 3. An idea of calculation of number of quotes for each year in online dictionary (histogram construction).

The traversal of 26,596 quotations (which contains date) makes possible to build the following histogram (Fig. 4), which relates the number of quotations and the source’s publication date in the range 1750-2012.⁵

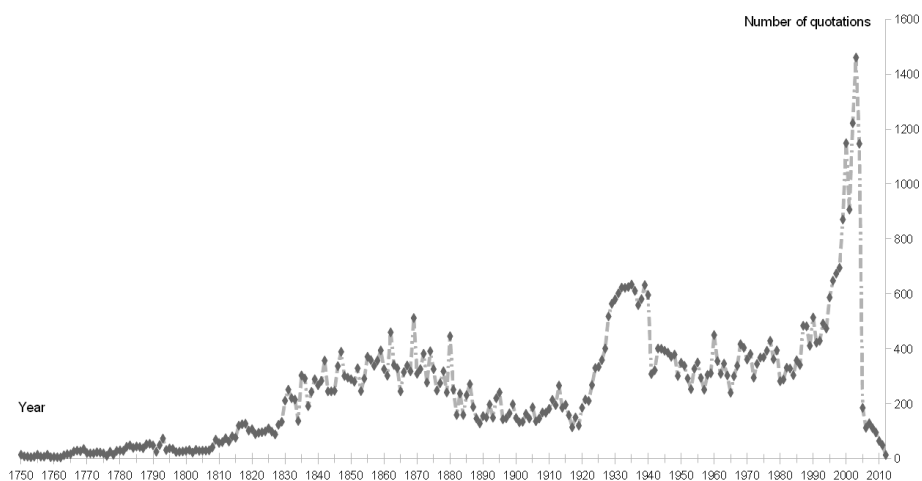


Fig. 4. The dependence of number of quotations with respect to the source’s publication date.

The peak number of quotations in the 2000s might be explained by a relatively high number of newspapers and journals available at the Russian National Corpus within this time range since this Corpus is the main source of quotations for the Russian Wiktionary (see above).

⁵ The source data for Fig. 4 is available at [http://ru.wiktionary.org/Участник:АКА%20МВГ/Статистика:Цитаты%20\(дата\)](http://ru.wiktionary.org/Участник:АКА%20МВГ/Статистика:Цитаты%20(дата))

In order to understand the relatively high number of quotations in Fig. 5 in the time range from the 1830s to the 1880s, the contribution of the most cited in the Russian Wiktionary writers is analyzed.

The writers with the highest number of quotations in the Russian Wiktionary are listed in the column “Author” in Table 2. The second and third columns demonstrate the fast growing size of the dictionary in a number of quotations for these writers in 2011 and 2012.

The main source of quotations in the Russian Wiktionary is the Russian National Corpus hence there is a column labeled “Publication in Russian National Corpus”, which provides the years of the first and last publications of the author presented in the corpus. For the same periods the total numbers of quotations in the Wiktionary were counted (column “Total quotes...”). The last column is the ratio of the number of quotations of the author (third column, 2012) to the total numbers of quotations for the periods, when the publications of the author is presented in the corpus (next to last column).

Table 2. The most popular authors in the Russian Wiktionary.

Author	Number of quotes		Publication in Russian National Corpus	Total quotes in Wiktionary (within this time range)	Contribution (%) 2012
	2011	2012			
Anton Chekhov	716	931	1880-1904	4,704	19,8%
Leo Tolstoy	529	710	1852- 1910	14,954	4,8%
Alexander Pushkin	520	627	1815 -1836	3,217	19,5%
Fyodor Dostoyevsky	500	776	1846-1881	11,853	6,6%
Ivan Turgenev	457	697	1846-1882	12,012	5,8%
Nikolai Gogol	321	473	1831-1847	4,511	10,5%
Nikolai Leskov	245	386	1862-1894	9,039	4,3%
Mikhail Bulgakov	207	267	1920-1940	10,049	2,7%
Arkady and Boris Strugatskye	171	225	1964-1979	5,699	4,0%
Viktor Astafyev	142	199	1967-2001	16,327	1,2%

The total number of quotations of the first seven authors in the period 1815-1910 (*Chekhov – Leskov* in Table 2) is 22429, it is 20.5%, i.e. one fifth part of the whole amount of quotations in this period in the Russian Wiktionary. Most probably, the high citations of these writers is the reason of the peak in Fig. 5 in the time range from the 1830s to the 1880s.

The open question remains, what Russian authors contribute to the peak in Fig. 5 in the period 1920s – 1940s before the World War II?

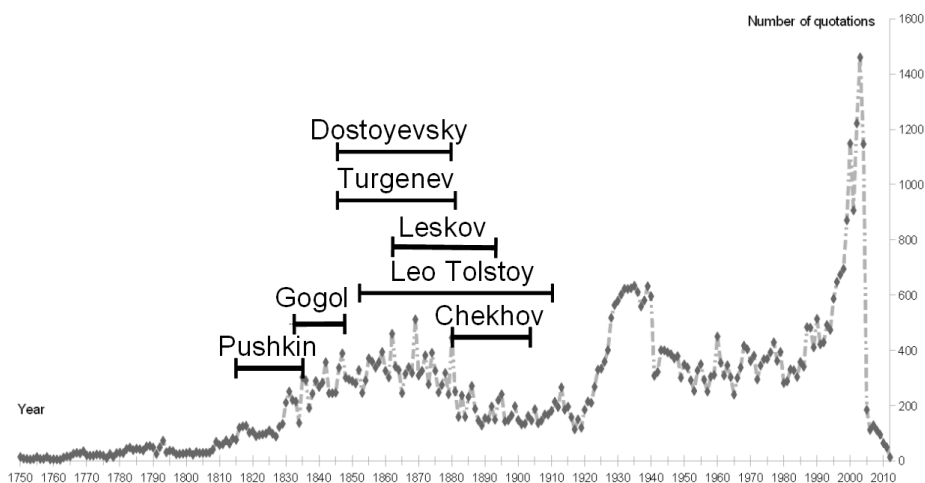


Fig. 5. The dependence of the number of quotations with respect the source’s publication date and the years of literary activity of the most cited in the Russian Wiktionary writers.

5.3 Distribution for Centuries

The analysis revealed that the earliest quotations in the Russia Wiktionary are dated:

- 70 BC, Cicero, “Against Verres”, Latin, the entry “asylum”.
- 1076, «Изборник Святослава» (Svyatoslav’s Miscellanies), Old East Slavic, the entry «воинъ».
- 1364, Guillaume de Machaut, “Dit de la Marguerite”, Old French, the entry “chançon”

In the course of experiments the distribution of quotes from the Russian Wiktionary dating from 17th to 21st century was made (Table 3). The 21st century corresponds to the range 2000-2012, inclusively.

Table 3. The distribution of Wiktionary quotes dating from 17th to 21st century.

Century	Quotes	%
17 th	405	1
18 th	1 576	2
19 th	21 394	32
20 th	36 260	55
2000-2012	6 644	10

It could be seen that each subsequent century contains more quotations than the previous one. Probably, this tendency will remain, since the first 12 years of this century already have given 10% of the whole number of quotations in the dictionary.

6 Conclusion

In this paper a framework of the machine-readable Wiktionary was designed, which emphasizes the possibility to add new wiktionaries to the parser modular architecture.

The architecture of the database of quotation corpus was described. An exemplary search task (to get a list of quotations in English, which refer to books written during more than one year) was solved.

The characteristics of corpus of quotations constructed based on the Russian Wiktionary were investigated. It was found that the number of quotations in the dictionary grows fast (51.5 thousands in 2011, 62 thousands in 2012).

The interesting statement is made in the paper [4] that “*the example sentences contained in Wiktionary are often artificially constructed by the authors of a Wiktionary entry and are, thus, not authentic materials taken from actual text corpora*”. Now it is possible to estimate the percentage of quotations taken from literary works (at least for the Russian Wiktionary). In the Russian Wiktionary, 23.8 thousand quotations (38.35% of the whole number) have a reference to the source in 2012 (17 thousand quotations with references in 2011, i.e. 33%). The percentage of quotations with references is growing.

The main source of quotations in the Russian Wiktionary is the *Russian National Corpus*. There are 94.15% quotations (of the whole number of quotations with references) which refer to the Russian National Corpus. Thus, more than one-third of all the quotations (36.1%) are authentic materials taken from the actual text corpus.

The following shortcomings and drawbacks of the quotation corpus of the Russian Wiktionary were revealed. There are a few quotations with references to texts dated by 17 and 18 centuries (3% of quotations only). Almost there are no quotations dated before the 17th century (Table 3).

The histogram which relates the number of quotations and the source’s publication date in the range 1750–2012 was created. It was made an attempt to explain the characteristics of the histogram by associating it with the years of the most popular and cited (in the Russian Wiktionary) writers of the nineteenth century: Anton Chekhov, Leo Tolstoy, Alexander Pushkin, Fyodor Dostoyevsky, Ivan Turgenev, Nikolai Gogol, and Nikolai Leskov.

Acknowledgments. Some parts of the research were carried out under projects funded by grants # 11-01-00251, # 12-01-00481 and # 12-07-00070 of the Russian Foundation for Basic Research, grant # 12-04-12062 of the Russian Foundation for Humanities and project of the research program “Intelligent information technologies, mathematical modeling, system analysis and automation” of the Russian Academy of Sciences. Some parts of this work were supported by the Ministry of Education and Science of Russian Federation (The Russian Federal Targeted Program “R&D in Priority Fields of S&T Complex of Russia for 2007-2013”, Contract No. 07.514.11.4139). The authors are grateful to Nickolay Teslya for his insightful comments.

References

1. Apresjan, Ju., Boguslavsky, I., Iomdin, B., Iomdin, L., Sannikov, A., Sizov, V. A: Syntactically and Semantically Tagged Corpus of Russian: State of the Art and Prospects. In: Proceedings of LREC. Genova, Italy, pp. 1378–1381 (2006)
2. Declerck, T., Morth, K., Lendvai, P.: Accessing and standardizing Wiktionary lexical entries for the translation of labels in Cultural Heritage taxonomies. In: Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12). Istanbul, Turkey (2012)
3. Hellmann, S., Auer, S.: Towards Web-Scale Collaborative Knowledge Extraction. Theory and Applications of Natural Language Processing pp. 1–27. (preprint) (2012)
4. Henrich, V., Hinrichs, E., Suttner, K.: Automatically Linking GermaNet to Wikipedia for Harvesting Corpus Examples for GermaNet Senses. Journal for Language Technology and Computational Linguistics (JLCL), Vol. 27, Number 1, pp. 1–19 (2012)
5. Krizhanovsky, A.A.: Transformation of Wiktionary entry structure into tables and relations in a relational database schema. Preprint. (2010)
6. Krizhanovsky, A.A.: A quantitative analysis of the English lexicon in Wiktionaries and WordNet. Int. J. of Intelligent Information Technologies (IJIT), accepted. Preprint (2013)
7. Lin, F., Krizhanovsky, A.: Multilingual ontology matching based on Wiktionary data accessible via SPARQL endpoint. In: Proceedings of the 13th Russian Conference on Digital Libraries RCDL'2011. Voronezh, Russia, pp. 19–26 (2011)
8. McFate, C., Forbus, K.: NULEX: An Open-License Broad Coverage Lexicon. In: The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Portland, Oregon, USA. Vol. 2, pp. 363–367 (2011)
9. Meyer, C. M., Gurevych, I.: Wiktionary: a new rival for expert-built lexicons? Exploring the possibilities of collaborative lexicography. Electronic Lexicography. Oxford: Oxford University Press, pp. 259–291 (2012)
10. Otte, P., Tyers, F.M.: Rapid rule-based machine translation between Dutch and Afrikaans. In: 16th Annual Conference of the European Association of Machine Translation, EAMT11 (2011)
11. Panchenko, A., Adeykin, S., Romanov, P., Romanov, A.: Extraction of Semantic Relations between Concepts with KNN Algorithms on Wikipedia. In: Concept Discovery in Unstructured Data Workshop (CDUD) of International Conference On Formal Concept Analysis, Belgium, pp. 78–88 (2012)
12. Qingyue, He: Automatic Pronunciation Dictionary Generation from Wiktionary and Wikipedia. Thesis. Karlsruhe Institute of Technology (2009)
13. Schlippe, T., Ochs, S., Schultz, T.: Wiktionary as a Source for Automatic Pronunciation Extraction. In: Proceedings of the 11th Annual Conference of the International Speech Communication Association, Makuhari, Japan, pp. 2290–2293 (2010)
14. Soderland, S., Lim, C., Mausam, Bo Qin, Etzioni, O., Pool, J.: Lemmatic machine translation. In: Proceedings of Machine Translation Summit XII, Ottawa, Canada (2009)

Linguistic Support of a CAPT System for Teaching English Pronunciation to Mexican Spanish Speakers

Olga Kolesnikova

Superior School of Computer Sciences (ESCOM), Instituto Politécnico Nacional,
Mexico City, 07738, Mexico

kolesolga@gmail.com

Abstract. This paper presents an error prevention approach in the development of a Computer Assisted Pronunciation Training System (CAPT System) for teaching American English pronunciation to adult Mexican Spanish speakers developed for the case of vowels. Prior knowledge of the learner's typical articulatory and auditory perception errors enhances the effectiveness of training. It enables to organize the teaching material in a way that foresees possible errors as well as to select appropriate exercises instead of using general pronunciation drills. Our first contribution is an extended comparative analysis of American English and Mexican Spanish vowels at the level of both phonemes and allophones. To the best of our knowledge, such analysis was not done in previous work. Another contribution of this work is a two-fold application of our analysis results: on the one hand, we use the results in designing error patterns to be employed in the error detection module of the CAPT system, and on the other hand, we employ the results as a basis for creating the linguistic content of the error prevention oriented tutor module of the same system. We illustrate this approach with examples.

Keywords. Teaching of pronunciation, English, Mexican Spanish speakers.

1 Introduction

I can speak, read and write but do not understand what they say! So many times EFL (English as a Foreign Language) teachers heard this perhaps most typical complaint of practically every EFL learner. The main reason of this problem is the absence of some English sounds in the learner's mother tongue or significant differences between the sound systems of English and L1 (First Language). This causes errors in sound recognition, and as a result, a considerable lack of understanding.

One of the means to resolve the problem of auditory phonemic recognition failure is an adequate teaching of English pronunciation since auditory comprehension is tightly connected with the human articulatory skills and phonemic knowledge as shown by research of phonological development in infants [28] and the development of listening/reading comprehension of children in the early grades [31]. These results can be applied to adult English learners since they follow similar language acquisition stages as children learning L1 [10]. Also, it is suggested that phonemic awareness,

and consequently listening comprehension, depends on articulation accuracy alongside with other factors like vocabulary size, topical knowledge, psychological and social aspects [24]. In this work, we focus on teaching pronunciation with the objective to improve listening comprehension suggesting that an EFL student has to get familiarized with English sounds, learn how to articulate them and after that get an appropriate training on phonemic, lexical, and phrase auditory recognition. Specifically, we argue that L1-oriented explanation of English sounds and language therapy based exercises can improve the overall quality of language learning.

This paper describes principles and gives examples of presenting and explaining the articulation of English sounds as well as exercises for the training stage. These principles and exercises are based on comparative articulatory phonetics for the explanation stage and speech and language therapy (SLT) for the training stage. In this work, we chose Mexican Spanish and American English language pair. The underlying idea in our work is that error prevention based on identified language-dependent error patterns is a more effective approach than error correction. It is not only efficient and emotionally comfortable for the learner, but also more feasible to implement in the pronunciation error detection module of CAPT systems, since at present automatic error detection irrespective to L1 has not yet reached a high quality level due to computational complexity of automatic speech recognition (ASR) task, see more detail in an overview of CAPT applications, Section 2.

Comparing phonetic systems of American English and Mexican Spanish helps us to predict specific articulation and recognition difficulties which may be experienced by Mexican Spanish ELT learners due to the assimilation effect. Based on such predictions, we develop error patterns and target phoneme presentations which anticipate the learner's problems in English pronunciation. This way the learner will comprehend a very important fact that pronunciation errors are not only "something that is not correct" but they are natural and normal steps in acquiring English. By means of errors, their adequate comprehension and more specific corrective training, the learner develops new articulatory and auditory habits. Such approach creates a stress-free context and helps learners to get rid of the fear of committing an error which does not allow them to make progress in language learning. Such fear is more typical in adult learners than in children. Besides, viewing errors as a "speech disorder", i.e., considering English sounds as correct and the corresponding Mexican Spanish assimilations as incorrect, we can apply the SLT techniques to deal with such "disorders".

Another consideration on pronunciation training is worth mentioning here. We deliberately use the term "English sounds" instead of "English phonemes" because we suggest that part of the reason of pronunciation and auditory comprehension problems is insufficient training, and sometimes an absence of training, in English allophones. If only phonemes are taught, the learner then is not able to recognize them when exposed to allophones which differ significantly from the "classical" phoneme presentations. Therefore, comparing American English and Mexican Spanish sound systems, we deal with the basic allophones.

The rest of the paper is organized as follows. Section 2 gives a brief overview of CAPT systems with a special attention to pronunciation assessment and error detection as well as a summary of existing ESL materials for Spanish speaking learners. Section 3 presents the basic architecture of a CAPT system. Section 4

contains a detailed comparative analysis of American English and Mexican Spanish vowel system on the level of both phonemes and allophones. Sections 5 and 6 list and explain error patterns for the error detection module of the system. Examples of teaching American English vowel sounds on the basis of comparative phonetic analysis are given in Section 7. Finally, we present conclusions and outline future work in Section 8.

2 Related work

2.1 CAPT Systems

Today it is practically beyond doubt that Computer Assisted Language Learning (CALL) is able to provide many benefits to teachers and learners including stress-free and interaction-rich context where teachers enjoy more opportunity to attend individual needs of students, since not all situations can be provisioned and programed in a computer application, while the students can practice at their own pace and get immediate personalized feedback [13]. Besides, techniques of electronic [11], mobile [14] and ubiquitous [2] learning further increase the effectiveness of acquisition.

The majority of CALL applications are oriented to acquisition of all language aspects: phonetic system, lexicon and word usage, grammar, pragmatics. However, a number of systems have been designed for Computer Assisted Pronunciation Training (CAPT), including the following commercial products: *NativeAccent*TM by Carnegie Mellon University's Language Technologies Institute, www.carnegiespeech.com; *Tell Me More*[®] Premium by Auralog, www.tellmemore.com; *EyeSpeak* by Visual Pronunciation Software Ltd. at www.eyespeakenglish.com, *Pronunciation Software* by Executive Language Training, www.eltlearn.com, among others. In particular, accent reduction software has become very popular, related to English-speaking countries naturalization and employment issues (e.g., in call centers, where intelligible pronunciation and perfect auditory comprehension are indispensable). Examples of accent reduction systems are *Accent Improvement Software* at www.englishtalkshop.com, *Voice and Accent* by Let's Talk Institute Pvt Ltd. at www.letstalkpodcast.com, *Master the American Accent* by Language Success Press at www.loseaccent.com.

The biggest issue in CAPT application design is to effectively implement the process of learner-system interaction for the program to be able to identify the learner's pronunciation errors and provide a necessary feedback. In other words, the system should operate similar to a human ESL teacher via the basic steps in teaching phonetic phenomena as follows.

1. Explanation: the teacher describes what position the articulatory organs must take and how they must move in order to produce the target sound or sound combination.
2. Imitation: the learner listens to words which contain the target sound and repeat them;

3. Adjustment: the teacher corrects the learner's errors while s/he is imitating the sound/s until its/their production is acceptable;
4. Recognition: the learner listens to input and discriminate the words with the target sound and the words without it.

The problem of human-CAPT system interaction is related to another task of Computer Science called Automatic Speech Recognition (ASR). ASR is a highly complex computational problem and much research effort has been devoted to it; the interested reader may consult the latest ASR advances in [5] and [21]. There have been a number of good efforts to apply ASR results in CAPT systems perusing the two-sided objective, i.e., phonemic recognition of the learner's speech and overall pronunciation assessment or individual error detection [7, 19]; the results obtained at this step are used by the system to generate corrective instructions to the learner.

For pronunciation assessment (evaluation of overall similarity to English speech), the following models have been used: hidden Markov models to calculate the score termed "goodness of pronunciation", GOP [32], Bayesian probabilistic scheme to compute intelligibility levels of students [27], Support Vector Machine to estimate the pronunciation quality score [9], auditory periphery models [12], and their combinations.

In combination with HMM, other strategies have been implemented to detect, or localize individual errors: dynamic time warping (DTW) technique [20], error rules of several types based on articulatory, receptive, and orthographic difficulties [16], Linear Discriminant Analysis [26], phonological rules derived from L1/English contrastive phonologic analysis made on the Cantonese-English language pair [15], error pattern definitions [29], etc.

Though much work has been done in the CAPT field, the challenge of creating an efficient interactive CAPT system still remains. Basically, there are two approaches to pronunciation teaching: the so-called universal approach when training is offered to learners with any L1 without taking it into account, and the L1-specific approach which make error detection more accurate due to mispronunciation prediction based on contrastive phonological analysis. We believe that while existing implementation of ASR techniques does not provide a high quality pronunciation assessment and consequently relevant individualized feedback to the learner, it is more effective to employ L1-oriented approach in tutor systems. To this end, we have made a comparative analysis of American English and Mexican Spanish sounds and based of it we defined some error patterns to be implemented in the error detection module of a CAPT system. We also illustrate our approach with some examples of L1-oriented presentation and explanation of English sounds and phonetic exercises designed in a way that prevents pronunciation errors in the learner's speech.

2.2 ESL Pronunciation for Spanish Speakers

There are scarce resources for Spanish learners of English pronunciation. The fullest courses are *English Phonetics and Phonology for Spanish Speakers* [18] and *A Course in English Phonetics for Spanish Speakers* [8], but they teach British English to Castilian Spanish speakers. Such books like *Teaching English Sounds to Spanish Speakers* [25], *English Pronunciation for Spanish Speakers: Vowels* [3], *English*

Pronunciation for Spanish Speakers: Consonants [4] teach American English, but are limited to some aspects of pronunciation and do not consider Mexican Spanish peculiarities. The approach of all the above mentioned courses is teaching phonemes and very few allophones, if any. This work addresses the lack of teaching resources for American English–Mexican Spanish language pair.

3 Overview of CAPT System Architecture

The basic architecture of a CAPT system includes four principal modules shown in Figure 1. The modules of the system interact with the human learner through interface.

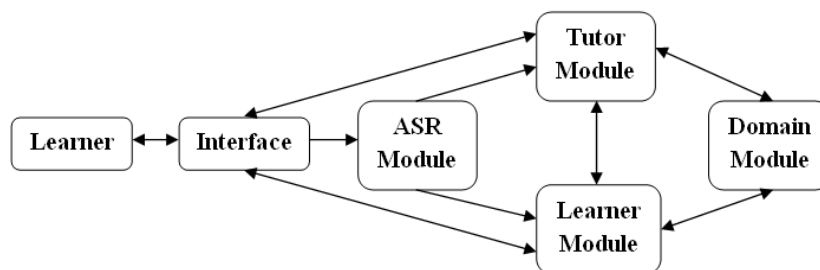


Fig. 1. Basic architecture of a CAPT system.

The tutor module simulates the English teacher; its functions are as follows:

- determine the level of the user (Mexican Spanish-speaking learner of English pronunciation);
- choose a particular training unit according to this learner’s prior history stored in the learner’s module as data introduced previously via the learner’s personal account in the system;
- present the sound or group of sounds corresponding to the chosen training unit and explain its articulation using comparison and analogy with similar sounds in Mexican Spanish;
- perform the training stage supplying the learner with training exercises, determining her errors, generating necessary feedback and selecting appropriate corrective drills;
- evaluate the learner’s performance;
- store the learner’s scores and error history in the learner’s module.

The learner module models the human learner of English; it contains the learner’s data base which holds the following information on the learner’s prior history:

- training units studied;
- scores obtained;

- errors detected during the stage of articulation training and the auditory comprehension stage.

The domain module contains the knowledge base consisting of two main parts:

- patterns of articulation and pronunciation and auditory perception errors typical for MS speakers as well as individual error samples;
- presentation and explanations of sounds, exercises for training articulation and auditory comprehension.

4 Comparison of American English and Mexican Spanish Vowel Sounds

The basic principle for developing a CAPT system according to the approach presented in this paper is pronunciation error prevention. In this work, we focus on errors in pronouncing AE (American English) sounds. By sounds we mean most frequently met allophones of AE phonemes.

Errors in sound generation may appear for two reasons. Firstly, a target sound may be absent in the source language; secondly, a target sound may exist in the source language but differ to some degree from its counterpart in the source language. In both cases, the learner will tend to substitute the AE sounds with the most similar sounds of her first language thus producing an accent in her speech.

A comparative analysis of AE and MS sound system allows us to predict what MS sounds may be used to substitute the AE sounds and design error patterns for the error detection module of the CAPT system as well as specific strategies for explaining and practicing each AE sound in the most effective way via the system's tutor module.

We have made our best attempt to present almost all allophones of AE and MS phonemes in a way that helps the teacher to predict errors in the AE learner's sound generation and offer her relevant explanation and corrective exercises.

It is not an easy task to compare the sounds of two languages due to the fact that phoneticians have different views on the inventory of phonemes and their allophones as well as on their definitions. Also, although much work has been done in AE phonetic studies, the same cannot be said about Mexican Spanish phonology and phonetics. An additional difficulty is the task itself. As mentioned previously, in order to produce a fluent and least accented AE speech on the one hand and to comprehend what is said in AE on the other hand, the learner should master well all basic allophones (sounds), at the same time relating them to the corresponding phonemes, since they are the basis for the English orthographic system.

Therefore, in the following four subsections we represented AE and MS vowel and diphthong phonemes together with their allophones as well as phonetic features of all included sounds using IPA phonetic alphabet and narrow transcription. The description of sounds relies on the works in [1, 6, 17, 22, 23, 30].

The sounds are described using the following pattern. First, we indicate if a given sound is American English (AE) or Mexican Spanish (MS). Then, the phonetic descriptors, or features, are listed. The phoneme sign is given in forward slashes, and

then an example word is presented. After that, the basic allophones of the sound are given: additional phonetic feature/s distinguishing this allophone is/are specified, the allophone symbol is given in brackets followed by an example word (or words) in which this allophone is used; lastly, we explain in what contexts and under what condition this allophone is produced. Besides, every example word is transcribed; its narrow transcription is given in brackets.

It can be noticed in Sections 4.1–4, that we have adopted a simplistic approach to MS phoneme inventory which takes into account only monophthong phonemes. This point of view is used in [22], for example. According to such approach, the sounds viewed by some phoneticians and Spanish teachers as diphthongs are viewed as combinations of respective phonemes. For example, in the word *aire* the first two vowels pronounced as [aĩ] are considered a diphthong in some phonetic literature, while in works of other researchers it is analyzed as a combination of the basic allophone of the /a/ phoneme and the allophone [ĩ] of the /i/ phoneme. By now in this work, we do not consider MS diphthongs.

4.1 Front Vowels

1. **MS high-front /i/ as in *ipo* [ˈipɔ].** Allophones: nasalized [ĩ] as in *instante* [ɪnˈstɑ̃nte], *mimo* [ˈmĩmo]; occurs between a pause and a nasal consonant or between two nasal consonants; palatal semi-consonant [j] as in *pasión* [paˈsjon], occurs in a prenuclear position; palatal semi-vowel [i̯] as in *aire* [ˈaĩre].
2. **AE high-front tense unrounded /i/ as in *neat* [niːt].** Allophones: diphthongized [iɪ] as in *flee* [fliː], occurs in open and stressed syllables; diphthongized [iə] as in *seal* [siəl], occurs before a liquid; reduced [ə] or [ɪ] as in *revise* [rəˈvaɪz] or [rɪˈvaɪz], in unstressed syllables; lengthened [i:] as in *bee* [bi:], word-finally; semi-lengthened [iː] as in *been* [biːn], before a voiced consonant; shortened [i] as in *beat* [biːt], before a voiceless consonant.
3. **AE lower high/front lax unrounded /ɪ/ as in *bit* [bɪt].** Allophones: reduced [ə] as in *chalice* [ˈtʃæləs], in unstressed syllables; lengthened [ɪ:] as in *carrying* [ˈkæriːŋ], in the position where two /ɪ/ sounds belonging to different morphemes meet.
4. **MS mid-front /e/ as in *este* [ˈeste].** Allophones: nasalized [ẽ] as in *entre* [ˈẽntre], *nene* [ˈnẽne], between a pause and a nasal consonant or between two nasal consonants.
5. **AE mid-front tense unrounded /e/ as in *ate* [et-].** Allophones: diphthongized [eɪ] as in *take* [teɪk-], in an open syllable; diphthongized and lengthened [eːɪ] as in *say* [seːɪ], word-finally; diphthongized and semi-lengthened [eːɪ] as in *name* [neːɪm] before a voiced consonant; diphthongized and shortened [eɪ] as in *lake* [leɪk-], before a voiceless consonant; changes to [i] or [ɪ] as in *Monday* [ˈmʌndɪ], in words with “-day”.
6. **AE lower mid-front lax unrounded /ɛ/ as in *get* [get-].** Allophones: diphthongized, r-colored and lengthened [ɛːə] as in *tear* [tʰɛːə], word-finally before the letter “r”; diphthongized, r-colored and semi-lengthened [ɛːə] as in *scared* [ˈskɛːəd-], before the letter “r” followed by a voiced consonant; diphthongized, r-colored and shortened [ɛə] as in *scarce* [skɛəs], before the letter “r” followed by

a voiceless consonant; triphthongized [eɪə] as in *jail* [dʒeɪəl], before /l/; changes to [ɪ] as in *get* [ɡɪt–], in informal speech.

7. **Æ low-front lax unrounded /æ/ as in *bat* [bæt̪]**. Allophones: lengthened [æ] as in *bad* [bæ:d̪] before a voiced consonant; shortened [æ] as in *bat* [bæt̪], before a voiceless consonant.

4.2 Central Vowels

1. **MS low-central /a/ as in *papa* [ˈpapa]**. Allophones: nasalized [ã] as in *ambos* [ˈãmbos], *mano* [ˈmãno], between a pause and a nasal consonant or between two nasal consonants.
2. **Æ lower mid-to-back central lax unrounded /ʌ/ as in *above* [əˈbʌv]**. Allophones: changed to [ɛ] as in *such* [sɛtʃ], in informal speech; changed to [ɪ] as in *just* [dʒɪst], in selected words.
3. **Æ neutral mid-central lax unstressed unrounded /ə/ as in *above* [əˈbʌv]**. Allophones: changed to [ɪ] as in *telephone* [ˈtelɪfɒn], in selected words.
4. **Æ mid-central r-colored tense /ɜː/ as in *perk* [pˈɜːk̪]**. Allophones: lengthened [ɜː:] as in *sir* [sɜː:], word-finally; semi-lengthened [ɜːː] as in *learn* [lɜːn], before a voiced consonant, shortened [ɜː] as in *thirst* [θɜːst–], before a voiceless consonant.
5. **Æ mid-central r-colored lax /ɜ̆/ as in *herder* [ˈhɜ̆dɜ̆]**. Allophones: r-dropped [ə] as in *motherly* [ˈmʌðəlɪ], before /r/.

4.3 Back Vowels

1. **Æ high-back tense rounded close /u/ as in *boot* [buːt–]**. Allophones: diphthongized [uə] as in *stool* [stuəl], before a liquid; diphthongized [uʊ] as in *do it* [ˈduːɪt–], in stressed or open syllables; reduced [ʊ] or [ə] as in *to own* [tʊˈɒn], *to go* [təˈɡo], in unstressed syllables; lengthened [u:] as in *blue* [blu:], word-finally; semi-lengthened [uː] as in *food* [fuːd–], before a voiced consonant; shortened [u] as in *loop* [luːp–], before a voiceless consonant.
2. **Æ high-back lax rounded /ʊ/ as in *book* [bʊk–]**. Allophones: reduced [ʌ] or [ə] as in *would* [wʌd–] or [wəd–], in rapid speech.
3. **MS mid-back /o/ as in *oso* [ˈoso]**. Allophones: nasalized [õ] as in *hombre* [ˈõmbre], *mono* [ˈmõno], between a pause and a nasal consonant or between two nasal consonants.
4. **Æ mid-back tense rounded close /o/ as in *owed* [od–]**. Allophones: diphthongized [ou] as in *go* [ɡou], in stressed and open syllables; reduced [ə] as in *window* [ˈwɪndə], in unstressed syllables; diphthongized and lengthened [oːu] as in *no* [noːu], word-finally; diphthongized and semi-lengthened [oːʊ] as in *load* [loːʊd–], before voiced consonants; diphthongized and shortened [ou] as in *coat* [kˈhɔt–], before voiceless consonants.
5. **Æ low mid-back lax rounded open /ɔ/ as in *bought* [bɔt–]**. Allophones: lengthened [ɔ:] as in *law* [lɔ:], word-finally; semi-lengthened [ɔː] as in *dawn* [dɔːn], before voiced consonants; shortened [ɔ] as in *thought* [θɔt–], before voiceless consonants; lowered [ɒ] or [ɑ] as in *cot* [kɒt–] or [kɑt–], after a velar consonant.

6. **AE low-back lax unrounded open /ɑ/ as in *pot* [pɑt̃].** Allophones: rounded [ɔ] as in *got* [gɔt̃], after a velar consonant; fronted [a] as in *not* [nat̃], after an alveolar consonant; fronted and rounded [ɔ] as in *father* [ˈfɑðə], in platform speech.
7. **MS high-back /u/ as in *pupa* [ˈpupa].** Allophones: nasalized [ũ] as in *un soto* [ˈũnˈsoto], *mundo* [ˈmũndo], between a pause and a nasal consonant or between two nasal consonants; velar semi-consonant [w] as in *cuatro* [ˈkwatro], in a prenuclear position; velar semi-vowel [u] as in *auto* [ˈaũto], in a postnuclear position.

4.4 Diphthongs

1. **AE rising low-front to high-front /aɪ/ as in *kite* [kaɪt̃].** Allophones: triphthongized [aɪə] as in *I'll* [aɪəl], before /l/; reduced [ə] *I don't know* [əˈdɒnˈno], in unstressed syllables in informal speech; lengthened [a:ɪ] as in *lie* [la:ɪ], word-finally; semi-lengthened [aɪ] as in *find* [faɪnd̃], before a voiced consonant; shortened [aɪ] as in *light* [laɪt̃], before a voiceless consonant; elevated [ɜɪ] as in *ice* [ɜɪs], before a voiceless consonant.
2. **AE rising low-front to high-back /aʊ/ as in *now* [naʊ].** Allophones: reduced [ʌʊ] as in *house* [haʊs], before a voiceless consonant.
3. **AE rising mid-back to high-front /ɔɪ/ as in *voice* [vɔɪs].** Allophones: lengthened [ɔ:ɪ] as in *boy* [bɔ:ɪ], word-finally; semi-lengthened [ɔɪ] as in *noise* [nɔɪz], before a voiced consonant; shortened [ɔɪ] as in *exploit* [əksˈplɔɪt̃], before a voiceless consonant.

5 Error Patterns for the Error Detection Module of CAPT System

In this section, some basic error patterns on the phoneme level are presented. They are derived theoretically from the results of comparing AE and MS vowel sound system given in Section 4. Certainly, such theoretical approach is not sufficient to identify all possible errors of an MS learner of English. Practical research is necessary to confirm, clarify, adjust or correct the theoretically predicted errors listed in this section. Also, more error patterns may be discovered in an empirical study of English speech produced by MS learners. We plan to do this research as future work.

Basically, all phoneme errors can be classified into three types:

1. Substitution of an AE phoneme by an MS phoneme.
2. Insertion of an MS phoneme in an AE word.
3. Deletion of an AE phoneme.

In the following three subsections, three types of errors are presented, respectively.

There are two main reasons due to which pronunciation errors are made: the first reason is phonetic, that is, a given AE sound does not exist in MS or if it exists, it differs in some way from it; the second reason is orthographic, when the MS reading rules are applied to AE words. For example, *bat* may be read [bat̃] instead of [bæt̃]

because the letter “a” is read as [a] in all contexts in Spanish. In case an MS learner knows the reading rule of “a” in a closed syllable, she may exhibit a phonetic error of substituting [æ] by [e], since the latter is the MS sound closest the [æ].

In Section 5.1 substitution error patterns are shown. We put the comment “due to orthography”, if an error is made for this reason. In case the reason is phonetic, we give no comment. In Section 5.2 insertion errors are listed, they are caused by the influence of MS orthographic patterns and reading rules. Section 5.3 speaks about deletion errors.

5.1 Substitution

The substitution errors are presented in Table 1.

Table 1. Substitution errors.

AE vowel sounds	Substitution by MS vowel sounds
High-front tense unrounded /i/ as in <i>neat</i> [nit̪]	High-front /i/ as in <i>ipo</i> ['ipo]
Lower high-front lax unrounded /ɪ/ as in <i>bit</i> [bit̪]	
Mid-front tense unrounded /e/ as in <i>ate</i> [et̪]	Mid-front /e/ as in <i>este</i> ['este]
Lower mid-front lax unrounded /ɛ/ as in <i>get</i> [gɛt̪]	
Low-front lax unrounded /æ/ as in <i>bat</i> [bæt̪]	
Mid-central r-colored tense /ɜ:/ as in <i>perk</i> [p ^h ɜ:k̪]	
Mid-central r-colored lax /ɚ/ as in <i>herder</i> ['hɜ:dɚ]	
Neutral mid-central lax unstressed unrounded /ə/ as in <i>above</i> [ə'bʌv]	
Mid-front tense unrounded /e/ as in <i>ate</i> [et̪], due to orthography	Low-central /a/ as in <i>papa</i> ['papa]
Low-front lax unrounded /æ/ as in <i>bat</i> [bæt̪], due to orthography.	
Lower mid-to-back central lax unrounded /ʌ/ as in <i>above</i> [ə'bʌv]	
High-back tense rounded close /u/ as in <i>boot</i> [but̪]	High-back /u/ as in <i>pupa</i> ['pupa].
High-back lax rounded /ʊ/ as in <i>book</i> [bʊk̪]	
High-back tense rounded close /u/ as in <i>boot</i> [but̪], due to orthography	Mid-back /o/ as in <i>oso</i> ['oso]
High-back lax rounded /ʊ/ as in <i>book</i> [bʊk̪], due to orthography	
Mid-back tense rounded close /o/ as in <i>owed</i> [od̪]	
Lower mid-to-back central lax unrounded /ʌ/ as in <i>above</i>	

AE vowel sounds	Substitution by MS vowel sounds
[ə'bʌv], due to orthography	
Low mid-back lax rounded open /ɔ/ as in <i>bought</i> [bɔt-]	
Low-back lax unrounded open /ɑ/ as in <i>pot</i> [pɑt̃]	
Rising low-front to high-back /aʊ/ as in <i>now</i> [naʊ], the nucleus /a/ is substituted by MS /o/ due to orthography	
Rising low-front to high-front /aɪ/ as in <i>kite</i> [kaɪt̃]	Combination of /a/ and /i/
Rising low-front to high-back /aʊ/ as in <i>now</i> [naʊ]	Combination of /a/ and /u/
Rising mid-back to high-front /ɔɪ/ as in <i>voice</i> [vɔɪs]	Combination of /o/ and /i/

5.2 Insertion

In Table 2 we present the insertion errors.

Table 2. Insertion errors

MS insertion	In AE words like
/r/ after /ɜ:/	<i>perk</i> [p ^h ɜ:k̃]
/r/ after /ɜ:/	<i>herder</i> [ˈhɜ:də]
/u/ after /ɔ/	<i>bought</i> [bɔt-]
/a/ after /i/	<i>neat</i> [nit̃]
/e/ after /t/	<i>ate</i> [et-]

5.3 Deletion

Compared to other types of errors, that is, substitution and insertion of phonemes, deletion of a vowel phoneme or its component is not an error very frequently exhibited by MS learners. The phenomenon of phoneme deletion is more typical for consonant sounds, especially in word final positions. Deletion of a vowel sound occurs mainly for orthographic reasons. For example, in the word *note* [nout̃], the second component of the diphthongized allophone of the phoneme /o/ may be deleted because the letter “o” is read as [o] in all context in Spanish.

6 Error Patterns in the Error Detection Module of CAPT System

One of the objectives of learning a second language is to develop speech production and speech recognition abilities. English learners are expected to understand AE speech as well as to realize their communicative intent generating speech in a way that is less accented and intelligible to native speakers. In view of this task, error detection and correction are seen as a very important part of language learning.

In the CAPT system architecture described in Section 3, the learner's speech is processed by the Automatic Speech Recognition (ASR) module, and the error identification function is performed by the Error Detection module.

Automatic error detection at the level of individual sounds is a complex computational task; it remains a challenge in CAPT system development. Compared to human judgment, automatic erroneous sound detection in CAPT systems is not all satisfactory [26]. Error detection rate can be improved if the error detection module is fed with error patterns to be used as guidelines for predicting errors in learner's speech.

Modern CAPT systems commonly use Hidden Markov Models for error detection. Let us consider the process of isolated word recognition since pronunciation training begins with mastering AE sounds in individual words. The process includes two major stages. Firstly, the ASR system is trained using a vocabulary of pronounced words mapped to their transcriptions which constitutes a phonetic database. Pronounced words are represented as speech vectors. Secondly, the system is exposed to unknown words (speech vectors) and generates their transcription. At the training stage, a HMM is trained for each word using a set of examples of that word. At the recognition stage, when an unknown word is presented to the system, it calculates the likelihood of each model generating that word and the most likely model is chosen. To build the system, Hidden Markov Models Toolkit [33] can be used.

In the words used at the training stage, each vowel sound prone to error can be aligned to a list of errors with their respective probabilities. The probabilities can be estimated in an empirical study of English speech produced by MS learners as a part of future research. Here we give the probability values as supposed by us based on theoretic research. These values may be used as a starting guess for initiating HMM models.

Table 3. Vowel pronunciation errors in the word *road*

Correct	Incorrect		
[rou̯d̩]	Transcription	Probability	Reason
	[rou̯d̩]	0.6	Orthographic
	[rou̯d̩]	0.35	Substitution of /u/ with /ʊ/
	[rod̩]	0.05	Deletion of /ʊ/

Consider the word *road* [rou̯d̩] (Table 3). Two transcriptions will be stored in the phonetic database: the correct transcription and the transcription including possible erroneous sounds annotated with their probabilities. In case the word exposed to the system differ significantly from the correct version based on a pre-defined threshold,

the system will take into account error pattern probabilities in order to identify the concrete error. We consider only errors in vowel sound pronunciation. In a complete model, all sounds, vowels and consonants, are considered.

7 Examples of Error-Preventive AE Sound Training for the Tutor Module of CAPT System

In this section we give two examples of teaching AE sounds to MS speakers taking into account the comparative analysis of AE and MS sounds presented in Section 4. The first example is described in more detail, and the second one is presented in a more concise way since the training stages in the second example are the same as in the first one.

The phoneme teaching is realized in the following stages:

1. AE phoneme presentation and explanation of its articulation in comparison with similar MS sound/s.
2. Training of the AE phoneme first using MS words with similar sound/s, then AE words of increasing complexity.
3. Training of auditory recognition of the AE phoneme first using minimal pairs, then words of increasing complexity, word combinations and phrases depending on the learner's level (elementary, intermediate, advanced).

In order to prevent errors in sound generation, the MS learner needs to understand the differences between the AE and MS sounds on the articulatory level as well as on the auditory level. Besides, she has to learn to relate the articulatory movements with the auditory effects produced by them, because this ability is fundamental for adequate speech recognition.

Taking these objectives into account, we suggest introducing, explaining and practicing the AE sounds not in minimal pairs but in so-called **minimal triplets**, adding to each minimal AE word pair a Spanish word containing a similar Spanish sound. Such MS sound may substitute the target AE sound/s and produce misunderstanding of English speech. For this reason, the learner should understand the difference in articulation and acoustics of AE and MS sounds in order to be able to produce less accented speech and avoid misunderstanding in oral recognition. Minimal triplets can be used at the elementary level of AE pronunciation training. When the AE sounds are pronounced sufficiently well by the learner, only AE words and phrases will be used in further stages.

As an example, we chose two AE phonemes: /u/ and /ʊ/. The phonemes /u/ as in *boot* [but-] and /ʊ/ as in *book* [buk-] are similar to MS /u/ as in *pupa*. For such reason these AE phonemes are substituted by the MS /u/, see substitution error patterns in Table 1. The phoneme /u/ as in *boot* [but-] is high-back tense rounded close, the phoneme /ʊ/ as in *book* [buk-] is high-back lax rounded, while the MS phoneme /u/ as in *pupa* is high-back.

Due to the fact that the phoneme pair /u/ – /ʊ/ is absent in MS, words having these phonemes are often confused by MS speakers which decreases their auditory

recognition capacity. In speech generation, both /u/ and /ʊ/ of American English may be substituted by MS /u/, as mentioned before.

At Stage 1, both AE phonemes are presented and explained in contrast with the MS /u/. Their similarities and differences should be clarified in detail accompanied by contrastive examples of minimal triplets. For example, the words listed below can be used. Remember, that two of them are English and the last word is Spanish.

<i>curso</i>	<i>cool</i>	<i>cook</i>
<i>julio</i>	<i>who</i>	<i>hook</i>
<i>tubo</i>	<i>tool</i>	<i>took</i>
<i>gusto</i>	<i>goose</i>	<i>good</i>
<i>luz</i>	<i>loom</i>	<i>look</i>
<i>nunca</i>	<i>noon</i>	<i>nook</i>

The articulation of all three sounds can be illustrated by diagrams showing the movements of the speech organs. The main difference between AE and MS articulation is that the AE phonemes are rounded and the corresponding MS phoneme is not.

At Stage 2, we suggest first to train the AE /u/ phoneme since it is closer to the corresponding MS phoneme. Here we suggest a method that can be called **building** of the AE /u/ on the foundation of its MS counterpart.

The learner starts from her familiar sound /u/ in a word like *pupa* and is told that this sound will be used as a basis for building the corresponding AE sound. Since the sounds are different, the learner has to change the articulation of /u/. For this purpose, the following language therapy exercise may be suggested: the learner is invited to pronounce the MS word *pupa* paying special attention to the /u/ sound and prolonging it, at the same time stretching her lips a little with her hands as in a smile thus avoiding the lip forward protraction typical for MS. The learner should listen carefully to the auditory difference which the lip stretching produces. At the same time, the learner is invited to observe her lip position and movement in a mirror and to form a narrow rounded mouth opening while slightly stretching the lips. The mirror also adds visual control for a more effective acquisition. When the learner understands the difference and is able to produce “the English version” of the Spanish /u/, more words are given for articulation training and auditory recognition exercises.

The phoneme /ʊ/ is built on the foundation of the AE /u/ when the latter is generated adequately. At this step, AE **minimal pairs** may be used, since the learner is supposed to have overcome her natural tendency to substitute the AE /u/ sound with the MS /u/. As an additional practice, the learner may be exposed to pairs of words in which the first word is an AE word containing the sound /ʊ/, and the second one is a similar MS word with the /u/ sound. This exercise will give another opportunity to the learner to reinforce her phonetic awareness of the difference between the AE /ʊ/ and the MS /u/ and to improve her pronunciation and recognition skills with respect to this sound pair.

Another example is teaching the AE phonemes /æ/ and /e/ in contrast with the similar MS phoneme /e/. The AE /æ/ as in *bat* [bæt̃] is low-front lax unrounded, the AE /e/ as in *ate* [et̃] is mid-front tense unrounded, and the MS similar phoneme /e/ as in *este* [ˈeste] is mid-front. Since the stages of teaching the AE /æ/ and /e/ are the same as of teaching any AE sounds, we do not explain each and every detail of the teaching

and acquisition process. Instead, we offer an example of minimal triplets and present a language therapy exercise for building /æ/ on the foundation of the MS /e/, then the AE /e/ is built on the foundation of /æ/. The minimal triplets are as follows.

<i>mes</i>	<i>mass</i>	<i>mess</i>
<i>necio</i>	<i>nag</i>	<i>neck</i>
<i>beca</i>	<i>back</i>	<i>beg</i>
<i>seja</i>	<i>sad</i>	<i>set</i>
<i>texto</i>	<i>tan</i>	<i>text</i>
<i>queso</i>	<i>cap</i>	<i>keg</i>

It may seem strange that we propose first to build /æ/ on the basis of the MS /e/, but not the AE /e/ on the basis of the same MS sound. The reason for this choice is that the difference between the AE /e/ and the MS /e/ is more subtle than the difference between /æ/ and the MS /e/. For a phonetically inexperienced learner it will be more difficult to perceive and produce such difference. Therefore, we think it is better to work on /æ/ at the beginning due to a bigger contrast which is easier for the learner to recognize and produce.

To build the /æ/ sound on the basis of the MS /e/, we suggest the following language therapy exercise. First, the learner is invited to pronounce the MS word *mes* slowly, prolonging the sound /e/. Then, while pronouncing the sound /e/ in *mes*, open the mouth wider, stretch the lips a little as in a smile. At this point it is important to avoid generating /a/ instead of /æ/. The learner is explained that opening her mouth will most probably produce the MS /a/ which should not be done. To prevent /a/ production, the learner is asked to hold the middle part of the tongue in a low position with the help of a spoon pressing the tongue slightly (without applying too much force to prevent undesirable physiological reaction), because the tongue elevation causes /a/ production. If the mouth is opened wider than in the MS /e/, the lips are slightly stretched and the middle part of the tongue is in its lower position, then the AE sound /a/ is generated.

The AE /e/ is trained on the basis of /æ/, asking the learner to open her mouth less than for the sound /æ/, but keeping the tongue in the same position as for /æ/ to avoid the MS /e/ generation.

8 Conclusions and Future Work

In this paper, we presented a detailed comparative analysis of American English and Mexican Spanish sound systems on the level of both phonemes and allophones. The results of this analysis can be used as a basis for creating the linguistic content of error prevention oriented CAPT system. Since error detection is a very difficult task of automatic speech recognition in intelligent tutor systems, its performance can be improved if a first language oriented approach in teaching English pronunciation is adopted. In our work, we considered Mexican Spanish and presented examples of how teaching articulation and auditory comprehension can be enhanced when typical error patterns are known in advance. In future, we plan to implement the results of our

phonetic analysis in designing a robust CAPT system for Mexican Spanish speakers and conduct experiments to compare such system with existing generally oriented tutor systems.

References

1. Avery, P., Ehrlich, S.: Teaching American English Pronunciation. Oxford University Press, England (1992)
2. Burbules, N. C.: Ubiquitous Learning and the Future of Teaching. *Encounters on Education*, vol. 13, pp. 3–14 (2012)
3. Dale, P.: English Pronunciation for Spanish Speakers: Vowels. Prentice Hall Regents, NJ (1985)
4. Dale, P., Poms, L.: English Pronunciation for Spanish Speakers: Consonants. Prentice Hall Regents, NJ (1986)
5. DeMori, R., Suen, C. Y.: New Systems and Architectures for Automatic Speech Recognition and Synthesis. Springer-Verlag NY Inc. (2012)
6. Edwards, H. T.: Applied Phonetics: the Sounds of American English. Singular Pub. Group, San Diego, CA (1997)
7. Eskenazi, M.: An overview of spoken language technology for education. *Speech Communication*, vol. 51(10), pp.832–844 (2009)
8. Finch, D. F., Ortiz Lira H.: A Course in English Phonetics for Spanish Speakers. Heinemann Educational Books Ltd, London (1982)
9. Ge, F., Pan, F., Liu, C., Dong, B., Chan, S. D., Zhu, X., & Yan, Y.: An svm-based mandarin pronunciation quality assessment system. In: *The Sixth International Symposium on Neural Networks*, pp.255–265. Springer Berlin Heidelberg (2009).
10. Ipek, H.: Comparing and Contrasting First and Second Language Acquisition: Implications for Language Teachers. *English Language Teaching*, vol. 2(2), pp.155–163 (2009)
11. Khan, B. H.: A Comprehensive E-Learning Model. *Journal of e-Learning and Knowledge Society*, vol. 1, pp.33–43 (2005).
12. Koniaris, C., Salvi, G., Engwall, O.: On mispronunciation analysis of individual foreign speakers using auditory periphery models. *Speech Communication* (2013)
13. Levy, M., Stockwell, G.: *CALL Dimensions: Options and Issues in Computer-Assisted Language Learning*. Lawrence Erlbaum Associates, Inc., NJ (2006)
14. Liakin, D.: Mobile-Assisted Learning in the Second Language Classroom. *International Journal of Information Technology & Computer Science*, vol. 8(2), pp.58–65 (2013).
15. Meng, H., Lo, W. K., Harrison, A. M., Lee, P., Wong, K. H., Leung, W. K., Meng, F.: Development of automatic speech recognition and synthesis technologies to support Chinese learners of English: The CUHK experience. *Proceedings of APSIPA* (2010)
16. Menzel, W., Herron, D., Bonaventura, P., Morton, R.: Automatic detection and correction of non-native English pronunciations. *Proceedings of INSTILL*, pp.49–56 (2000).

17. Moreno de Alba, J. G.: *El español en América*. Fondo de cultura económica, México (2001)
18. Mott, B. L.: *English Phonetics and Phonology for Spanish Speakers*. Edicions Universitat de Barcelona, Barcelona (2005).
19. Neri, A., Cucchiari, C., Strik, W.: Automatic speech recognition for second language learning: how and why it actually works. In: *Proceedings of ICPhS*, pp.1157–1160 (2003).
20. Nouza, J.: Training speech through visual feedback patterns. In: *Proceedings of ICSLP* (1998)
21. Pieraccini, R.: *The Voice in the Machine*. MIT (2002)
22. Pineda, L.A., Castellanos, H., Cuétara, J., Galescu, L., Juárez, J., Llisterra, L., Pérez, P., Villaseñor, L.: The Corpus DIMEx100: Transcription and Evaluation. *Language Resources and Evaluation*, vol. 44(4), pp.347–370 (2010)
23. Quilis, A.: *El comentario fonológico y fonético de textos: teoría y práctica*. 3a edición. Arco/Libros, S.L., Madrid (1997)
24. Roberts, T. A.: Articulation Accuracy and Vocabulary Size Contributions to Phonemic Awareness and Word Reading in English Language Learners. *Journal of Educational Psychology*, vol. 97(4), pp.601–616, (2005)
25. Schneider, L. C.: *Teaching English Sounds to Spanish Speakers*. Allied Educational Council (1971)
26. Strik, H., Truong, K., de Wet, F., Cucchiari, C.: Comparing different approaches for automatic pronunciation error detection. *Speech Communication*, vol. 51(10), pp. 845–852, doi: 10.1016/j.specom.2009.05.007 (2009)
27. Tsubota, Y., Kawahara, T., Dantsuji, M.: Practical use of English pronunciation system for Japanese students in the CALL classroom. In: *Proceedings of ICSLP*, pp.849–852 (2004).
28. Vihman, M. M.: *Phonological development: The origins of language in the child*. Blackwell, Oxford (1996)
29. Wang, Y. B., Lee, L. S.: Improved approaches of modeling and detecting error patterns with empirical analysis for computer-aided pronunciation training. In *Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, pp.5049–5052 (2012).
30. Whitley, M.S.: *Spanish-English Contrasts: A Course in Spanish linguistics*. Georgetown University Press, Washington, D.C. (1986)
31. Willson, V. L., Rupley, W. H.: A structural equation model for reading comprehension based on background, phonemic and strategy knowledge. *Scientific Studies of Reading*, vol. 1, pp.45–63 (1997)
32. Witt, S., Young, S.: Computer-aided pronunciation teaching based on automatic speech recognition. In: Jager, S., Nerbonne, J. A., van Essen, A. J. (eds.) *Language teaching and language technology*, pp. 25–35, Swets & Zeitlinger, Lisse (1998)
33. Young, S., Jansen, J., Odell, J., Ollason, D., Woodland, P.: *The HTK Book*. Cambridge University (1996)

Corpus morfológicamente representativo: preparación de datos y compilación para el español

Liliana Chanona-Hernández¹ y Alexander Gelbukh²

¹ESIME-Zacatenco,
Instituto Politécnico Nacional (IPN),
México DF,
México

²Centro de Investigación en Computación (CIC),
Instituto Politécnico Nacional (IPN),
México DF,
México

lchanona@gmail.com, gelbukh@gelbukh.com

Resumen. El problema de compilación automática de los corpus es uno de los problemas importantes de lingüística computacional. En los corpus tradicionales algunas palabras tienen demasiada ocurrencia y algunas tienen poca o ninguna ocurrencia según la ley de distribución de palabras de acuerdo a su rango: la ley de Zipf. En el trabajo proponemos el concepto del corpus representativo morfológicamente —cuando para cada palabra de alguna lista se garantiza por lo menos algún número determinado de contextos— y se describe un ejemplo de aplicación al español: la compilación automática de tal corpus a través de Internet, incluyendo la preparación de los datos iniciales y el filtrado de los contextos.

Palabras clave. Corpus representativo, español, representatividad morfológica, Internet.

1. Introducción

El tesoro más valioso de la raza humana es el conocimiento, es decir, la información. Existe en el mundo volúmenes inmensos de información en forma de lenguaje natural: los libros, los periódicos, las revistas, etcétera. Pero la posesión verdadera de este tipo de tesoro implica la habilidad de hacer ciertas operaciones con la información:

- Buscar la información necesaria.
- Comparar las diferentes fuentes, hacer inferencias lógicas y concluir.
- Manejar los textos, por ejemplo, traducirlos a otros idiomas.

Todo parece estar preparado para el uso de las computadoras para procesar volúmenes grandes de información: los métodos lógicos ya son muy fuertes, los procesadores muy rápidos, muchos textos ya están disponibles en forma digital, tanto en las casas editoriales como en Internet. El único problema para la computadora al procesar los textos es que para manejarlos bien hay que entenderlos [1, 2]. Sin eso, éstos son solamente cadenas de letras sin ningún sentido y no una información útil para el razonamiento lógico, lo que es una de las tareas más importantes es la búsqueda y el uso de la información contenida en el texto. Los motores de búsqueda permiten encontrar una infinidad de documentos aquellos que satisfagan una necesidad del usuario descrita en su petición. En el caso simple, la petición contiene las palabras clave, como «pensar y futuro», que quiere decir que el usuario necesita los documentos que contengan ambas de estas palabras. Hasta en este caso simple, se necesita un análisis bastante complejo: los documentos que contienen, las palabras futuras, piensan y probablemente pensador y pensamiento son también relevantes. Si el usuario indica que necesita los documentos que dicen sobre la acción «pensar en futuro», entonces el documento contiene la frase como «piensan en las futuras aplicaciones» probablemente es relevante, mientras que «en el futuro próximo voy a pensarlo» probablemente no lo es. Para hacer esta decisión es necesario un análisis profundo del texto.

La búsqueda eficaz de documentos depende de las soluciones técnicas. No es posible analizar todos los documentos cada vez que el usuario hace su pregunta. Entonces, se hace y se maneja un índice de los documentos, es decir, una representación corta, simple y formal de los documentos. La representación de este índice, el tipo de información incluida en él y los métodos matemáticos que se emplean en la búsqueda en él afectan mucho los resultados y eficiencia del proceso.

Típicamente, un corpus se construye de tal manera que los textos de diferentes géneros, sobre diferentes temas, escritos por diferentes autores, etc., se representan en él en una mezcla balanceada, para reflejar las propiedades promedio del lenguaje. Esta propiedad del corpus también se llama representatividad (respecto a los temas, géneros, etc.) [7], aunque en este trabajo usamos otro significado de la palabra *representativo*, hablando del corpus representativo respecto a las palabras individuales.

El uso de los corpus en lingüística es de suma importancia [8]. Los métodos actuales para el análisis de lenguaje natural emplean ampliamente los conocimientos sobre el lenguaje, su uso, y el mundo real, es decir, diccionarios de una gran variedad de tipos. En la compilación de éstos, hay dos momentos críticos en que se usan los corpus:

- Extracción de información **simbólica**. En el diccionario de tipo simbólico, se almacenan los datos sobre la existencia (o no existencia) de alguna relación o propiedad. Por ejemplo, el hecho de que la palabra *ayuda* se usa tanto con el verbo *prestar* como con los verbos *dar* y *proporcionar*: *prestar ayuda*, *dar ayuda*, *proporcionar ayuda*. Incluso para las cantidades pequeñas de palabras, a una persona le cuesta trabajo recordar toda la información necesaria y llenar el diccionario apoyándose sólo en introspección. A diferencia, para los diccionarios del tamaño realista, estos datos se deben extraer del uso de las palabras en el corpus –en

Corpus morfológicamente representativo: preparación de datos y compilación para el español

nuestro ejemplo, buscando todos los verbos con los cuales se usa la palabra *ayuda*.

- Extracción de información **estadística**. Sin embargo, la información solamente simbólica no es suficiente en muchas aplicaciones donde hay que resolver incertidumbre y ambigüedad. En muchos casos se necesita saber no sólo con cuales verbos se usa la palabra, sino con cuáles se usa más que con otros: 60%: *prestar ayuda*, 30%: *dar ayuda*, 10%: *proporcionar ayuda*.

Esta información no se puede compilar manualmente con el método de introspección, sino sólo se puede extraer de un corpus amplio con los métodos estadísticos.

En el resto del artículo, primero presentamos el concepto del corpus representativo morfológicamente, después describimos el algoritmo de su compilación de manera general, y posteriormente su aplicación para el caso del idioma español (selección de la lista inicial de palabras y ponderación morfológica). Al final se presentan las características del corpus compilado para el español y se dan las conclusiones.

2. El corpus representativo morfológicamente

El problema de casi cualquier investigación estadística en lexicografía es el hecho de que la mayoría de las palabras se encuentran en el corpus muy pocas veces, aunque un número relativamente pequeño de las palabras ocurren muchas veces y constituyen casi todo el corpus. Esto se conoce como la ley de Zipf: la palabra con el rango estadístico n tiene aproximadamente la frecuencia C/n . (C es una constante) por consecuencia, los resultados estadísticos compilados para la mayoría de las palabras del diccionario con el corpus son de baja confiabilidad, aunque sea un corpus muy grande, véase [9, 10].

De hecho, esta distribución es aún más sesgada y se aproxima más al inverso del cuadrado de n . Es decir, hay un conjunto pequeño de palabras muy frecuentes y muchas que aparecen muy pocas veces o sólo una vez (sea cual sea el idioma usado).

Esto significa que para la mayoría de las palabras no hay suficiente información estadística (suficiente número de contextos) aunque el corpus sea muy grande y ocupe mucho espacio.

Entonces, los corpus tradicionales no son perfectamente adecuados para las tareas de PLN. La ley de Zipf se refiere a que muy pocas palabras en cualquier lenguaje son muy frecuentes, mientras que la mayoría de las palabras son poco frecuentes y de hecho las palabras raras son totalmente carentes de frecuencia. Por consecuencia, cuando se quiere hacer un estudio sobre cuál es el contexto de cualquier palabra con poca frecuencia es muy difícil porque casi no se tiene ninguna (o se tiene muy poca) información acerca de sus contextos. Es decir, en el corpus tradicional la información sobre la mayoría de las palabras no es estadísticamente significativa.

Un peor problema se presenta por la ley de Zipf cuando se trata de la investigación de ocurrencias de palabras. Si cada una de las palabras (tal que *ayuda* o *proporcionar*)

tiene poca frecuencia, su ocurrencia (*proporcionar ayuda*) tiene la frecuencia en el corpus casi nula y totalmente insuficiente para cualquier conclusión estadística.

Por otro lado, casi todo el volumen de los datos (y entonces, espacio y tiempo de procesamiento de datos por la computadora) de un corpus tradicional se ocupa por millones de inútiles repeticiones de las mismas 100–1,000 palabras.

En nuestro trabajo proponemos la solución a este problema: un corpus en el cual cada palabra tiene una frecuencia suficiente para su investigación estadística, este trabajo es una extensión de nuestro trabajo anterior [3], también se hace un cambio del enfoque con respecto al trabajo anterior.

Como la solución del problema de compilación del corpus representativo, se propone usar Internet como una fuente inmensa de los contextos típicos de las palabras. Sin embargo, no es factible técnicamente bajar de Internet todos los documentos en español que en éste existen. Afortunadamente, no es necesario, pues sólo se necesita un número limitado (digamos, de 50 contextos) para un número limitado (aproximadamente 100 mil) de palabras que existen en español (o bien, un millón de formas de palabras).

Este tipo de diccionarios se conoce en literatura como *concordancias* o diccionarios tipo KWIC (por sus siglas en inglés: *key words in context*). Nuestra solución propuesta, entonces, es la compilación de una concordancia tipo KWIC muy grande a través de la extracción de los contextos necesarios de los textos en español que se encuentran en Internet y con las posibilidades de enriquecimiento automático (no realizadas en nuestro trabajo). Sería muy bueno tener los contextos suficientemente grandes para su posterior análisis sintáctico automático, por ejemplo, con el parser libremente disponible FreeLing [1, 6].

La implementación de esta idea se describe en las siguientes secciones.

La Web es un gran repositorio de datos y un nuevo medio de publicación al alcance de más de mil millones de personas. El hacer uso eficiente y adecuado de estos datos depende de las herramientas que existen.

La técnica es usar una máquina de búsqueda (*search engine*) como [AltaVista](#), [Fast](#), [Inktomi](#), [Northern Light](#), [Lycos](#) o [Google](#), que usan el paradigma de recuperación en texto completo. Es decir, todas las palabras de un documento se almacenan en un índice para su posterior recuperación. Aunque en muchos casos las búsquedas en estas máquinas son efectivas, en otros son un total desastre. El problema es que las palabras no capturan toda la semántica de un documento. Hay mucha información contextual o implícita que no está escrita, pero que entendemos cuando leemos. Los problemas principales son la *polisemia*, es decir, palabras que tienen más de un significado, y por lo tanto encontramos páginas que no queremos; y la *sinonimia*, palabras distintas que tienen el mismo significado y por ende si no usamos la palabra correcta, no encontramos lo que queremos.

4. Algoritmo de compilación del corpus para el español

4.1. El algoritmo principal

El esquema general del método se presenta en la Fig. 1

Este tipo de estructura no es permitido, así que la función del analizador léxico es la de transformar este renglón a dos palabras:

trabajador
trabajadora

admitidas más adelante en la *agenda* (que es la tabla donde se guardan las palabras).

Enseguida las palabras pasan por el *filtro* que es el que se encarga de verificar que las cadenas de caracteres que son mezcla de letras y números, no entren en la agenda (que es la tabla donde se guardan las palabras).

Una vez que las palabras ya fueron filtradas son pasadas por un *analizador morfológico* que se encarga de normalizar las palabras. La normalización se realiza porque, es más fácil y práctico trabajar con las palabras normalizadas, así de este modo más adelante las palabras son pasadas por un *generador de formas* que como su nombre indica nos da todas las formas morfológicas de una palabra, asegurándonos que tendremos todas las formas de una palabra.

Conforme cada palabra ha sido filtrada y pasada por el analizador morfológico se almacena en una *agenda* que es la lista de palabras que ya cumplieron con ciertas normas.

Esto da pie a que las palabras sean tomadas por un *módulo de control* que es el que se encarga de pasar las palabras a un generador de formas morfológicas.

El *generador de formas* se encarga de generar todas las formas morfológicas de una palabra, un ejemplo simple es el sustantivo *mesa* y sus formas son *mesa* y *mesas*, este generador de formas es importante porque es necesario tener todas las variantes (formas morfológicas) de una palabra, pues de lo contrario el corpus representativo de palabras de español no estaría completo. Explicado de otra manera se busca tener una representatividad equilibrada de todas las formas morfológicas de una palabra.

Un punto importante que hay que mencionar, es el hecho de que existen tres formas de hacer un corpus representativo de palabras:

- El primero es el de compilar el corpus para las palabras normalizadas, cuando el número determinado de contextos en total se distribuye uniformemente entre sus formas morfológicas.
- La segunda forma, es la de tener todas las palabras normalizadas con todas sus respectivas formas, con un determinado número de contextos para cada forma.
- Finalmente, la que se está usando en este trabajo que es la de calcular para cada forma morfológica de dicha palabra el número de contextos que se desea obtener.

Cuando que ya se tiene todas las formas morfológicas de una palabra, cada una de las formas y la palabra normalizada se pasan a la *interfaz de búsqueda* que es el mediador entre el *generador de formas* y el *buscador de Internet*, una vez que el buscador le contesta a la interfaz, la interfaz de búsqueda a su vez le informa al *módulo de ponderación* (que es el encargado de hacer los respectivos cálculos de cuantos contextos se desea, tanto para la palabra normalizada, como para las formas morfológicas de dicha palabra), cuantos documentos existen en Internet con esas palabras. El módulo hace los cálculos y le dice al generador de formas cuantos

contextos tendrá cada forma de la palabra y la palabra misma. De este modo cuando ya el generador de formas sabe cuántos contextos exactamente tendrá cada palabra, vuelve a pasar las palabras a la interfaz de búsqueda para luego pasarlas al buscador de Internet.

El paso siguiente se da cuando, el buscador le envía todas las direcciones a la interfaz de búsqueda, quien ésta vez, las envía a un módulo *Analizador de respuesta* que se encarga de depurar todo el excedente que acompaña al URL.

Por ejemplo:

Descargue GRATUITAMENTE MSN Explorer en <http://explorer.msn.es/intl.asp>

Todo lo que se encuentra sombreado será eliminado, pues no es de utilidad.

La *Interfaz de documentos* es la que se encarga de conectarse a cada uno de los documentos con las direcciones que le pasa el *Analizador de respuesta*. Cuando ya se ha revisado cada uno de los documentos, sus direcciones son almacenadas en una tabla que es de los URLs visitados, esto es para no volver a visitar dos veces un mismo sitio.

El *Analizador de documento* es el que se encarga como su nombre lo indica de analizar cada uno de los documentos, para esto marca los párrafos que deben considerarse para el estudio y también auto enriquece el corpus, pasando al filtro las palabras nuevas que encuentra.

El *filtro de contextos* es el que se encarga de ver que el contexto no contenga menos de un número determinado de palabras y de que los contextos no sean repetidos, cuando ya la palabra pasa este filtro es almacenada en una tabla que se llama *Resultados*.

Para funcionar correctamente, el programa mantiene los siguientes datos principales:

1. Una lista de palabras para las cuales se pretende encontrar los contextos.
2. Una lista de los URLs encontrados para una palabra.
3. Una lista de contextos para cada palabra.

El algoritmo principal se esboza a continuación.

- Paso 1. Hacer vacías todas las listas.
- Paso 2. Agregar a la lista de palabras, todas las palabras encontradas en el corpus inicial, en este caso de la lista de palabras definidas en el diccionario Anaya.
- Paso 3. Si todas las palabras ya se procesaron, terminar el trabajo.
- Paso 4. Para una palabra todavía no procesada, buscar contextos en Internet.
- Paso 5. Para cada contexto encontrado en el Paso 4, aplicar las heurísticas para determinar si el contexto contiene alguna irregularidad.
 - i. Si la contiene, ignorar el contexto y aumentar el número de contextos irregulares.

- ii. Si no la contiene, agregar el contexto a la lista de los contextos encontrados para la palabra.

Paso 6. Si para la palabra ya se encontraron al menos 50 contextos válidos, marcar la palabra como ya procesada. Si no existen en Internet más contextos para esta palabra, también marcar la palabra como ya procesada. Ir al Paso 3.

4.2. Análisis y filtrado del texto obtenido

Al obtener el texto del documento en el formato HTML se buscan las ocurrencias de la palabra y se evalúa si los contextos son apropiados o no, como se describa a continuación.

El primer paso es quitar el marcado del formato HTML, para lo cual se usa el algoritmo desarrollado previamente en el Laboratorio de Lenguaje Natural. Es importante notar que esta función conserva la estructura de bloques de documento, por ejemplo, cada celda de una tabla pertenece a un bloque diferente. Esto se logra con sustitución de cada elemento del marcado con un símbolo especial, el cual es “#”.

Después en el texto se busca la palabra, lo cual se hace con la función estándar de `C strstr()`. Para hacer esta búsqueda, se tiene que hacer una copia del documento en minúsculas. Ya que nos interesan solo palabras y no partes de palabras, se verifica que es una palabra completa (es decir, que no existen símbolos de letras inmediatamente antes y después del segmento encontrado).

El siguiente paso en el análisis es la búsqueda del contexto de la palabra que se agrega a la lista de contextos potenciales. Más tarde este contexto pasa por el filtro de contextos el cual rechaza los contextos “malos”, véase la sección 4.3.

Como contexto se toma la misma oración donde se encontró la palabra, la oración anterior, y la oración siguiente (si las dos últimas existen). Pero al mismo tiempo se limita el número de palabras a la izquierda y a la derecha de no más de 25; las palabras más lejanas se ignoran. La limitación del número de palabras (25) se debe a la necesidad de limitar el tamaño del contexto para que la base de datos no sea demasiado grande. Además, las palabras más lejanas usualmente no tienen ninguna relación lingüísticamente interesante con la palabra en cuestión.

Para encontrar el contexto, se hace el siguiente análisis:

1. Se busca a la izquierda de la palabra el símbolo de fin de oración (punto, signo de exclamación o de interrogación) y se cuenta el número de palabras (no debe ser mayor que 25). También sólo se permite un símbolo de fin de oración a la izquierda para que sólo se tome la oración anterior. Se toman en cuenta los símbolos de fin de bloque (“#”) para que el contexto no los rebase.
2. Se hace el análisis semejante, a la derecha de la palabra.

Como el resultado de este algoritmo se obtiene el contexto potencial el cual va a pasar al filtro y ser aprobado (se guarda como un resultado final) o rechazado (se ignora).

4.3. Filtros adicionales de contextos encontrados

Después del proceso de extracción del contexto, se tiene el contexto potencial y la lista de contextos encontrados para la palabra. Ahora se toma la decisión si el contexto es bueno para incluirlo en el corpus que se está compilando.

Primero se usa el criterio del **tamaño** del contexto: el contexto debe contener no menos que un número dado de palabras, usamos el valor 8. Aplicamos este criterio porque los contextos demasiado pequeños no contienen suficiente información lingüística y no son de gran interés. Más importante, los contextos pequeños frecuentemente no son expresiones de lenguaje natural sino otros tipos de datos (rótulos cortos de figuras, inscripciones en los controles de la pantalla como los botones, nombres de archivos, etc.).

Otro filtro que se usa tiene como propósito filtrar los apellidos y **nombres propios**. Los experimentos han mostrado que una palabra puede ser usada muchas veces como apellido (por ejemplo, casi todos los contextos encontrados para la palabra *abad* referían al apellido). Entonces, se verifica que la palabra no sea un apellido. Para eso se usa la siguiente heurística: si la palabra empieza con mayúscula y no tiene inmediatamente antes un signo de fin de oración, entonces consideramos que este contexto contiene apellido o nombre propio y por lo tanto es inaceptable.

El último filtro verifica que los contextos no **se repitan**. Para eso se verifican las dos palabras significativas que se encuentran inmediatamente a la izquierda y a la derecha de la palabra con la cual se trabaja. Estas palabras se comparan con todos los contextos ya encontrados; la cadena de las tres palabras no debe repetirse. Si se repiten, es decir, el contexto con estas palabras ya está en la base de datos, entonces el contexto se considera inaceptable. Nota: solo se verifican las palabras significativas, es decir, se ignoran los artículos, pronombres (*te, tí, se, etc.*) y preposiciones (*con, a, por, para, etc.*).

5. Preparación de lista inicial de palabras

Antes que el sistema pueda buscar en Internet los contextos para las palabras, es necesario preparar la lista de las palabras. Para prepararla se usó el diccionario explicativo del español desarrollado por el grupo español Anaya 1996. El diccionario se tiene en el Laboratorio de Lenguaje Natural del CIC-IPN en forma del archivo de texto. Lo que se hizo fue seleccionar las palabras encabezadas (para cuales hay definiciones en el diccionario) y formar la lista de estas palabras. En total el diccionario contiene alrededor de 30,000 palabras definidas.

Los problemas principales en la compilación de esta lista de palabras fueron relacionados con el análisis de la estructura del diccionario.

Hay que mencionar que las palabras definidas en el diccionario explicativo ya están normalizadas. Entonces, la etapa del análisis morfológico fue innecesaria en caso de este tipo de entrada. En caso que esta etapa fuese necesaria (digamos, cuando se usa algún corpus como la fuente de la lista de palabras), se utiliza la herramienta desarrollada en el Laboratorio de Lenguaje Natural: un analizador morfológico [2].

Ahora bien, al tenerse la lista de palabras normalizadas, es necesario para cada palabra normalizada obtener todas sus formas morfológicas y usarlas como peticiones a los motores de búsqueda en Internet. Para generar todas las formas morfológicas de cada una de las palabras se usó otra herramienta desarrollada previamente en el Laboratorio de Lenguaje Natural —un generador de formas de palabras (que es, de hecho, el mismo analizador morfológico invertido).

Aplicando esta herramienta, obtenemos para cada palabra normalizada (lema) la lista de sus formas gramaticales y guardamos todas estas formas en una base de datos.

Al terminar este procedimiento, la lista inicial está hecha. En el sistema desarrollado, el número de formas de palabras es alrededor de 100,000 (para 30,000 lemas).

6. Ponderación de las formas de palabras

Existen varias formas de hacer un corpus representativo de palabras, véase la discusión en [4]. Las más sencillas son:

- Un modo es el de compilar el corpus para las palabras normalizadas, cuando el número determinado de contextos en total se distribuye uniformemente entre sus formas morfológicas.
- La segunda forma, es la de considerar todas las palabras normalizadas con todas sus respectivas formas, con un determinado número de contextos para cada forma morfológica de la palabra dada.
- Finalmente, la que usamos en este trabajo es la de calcular para cada forma morfológica de dicha palabra el número de contextos que se desea obtener, de acuerdo con las estadísticas de uso de las formas específicas.

Ya que escogimos la opción de compilar el corpus usando la representación igual de lemas y hacer la ponderación de las formas de palabras, entonces necesitamos asignar las frecuencias a todas las formas gramaticales correspondientes a cada lema.

Decidimos que la ponderación se haga tomando en cuenta la frecuencia de la palabra en Internet. Entonces, eso se hace en el programa que busca las palabras en Internet.

El proceso consiste en 3 pasos:

1. Buscar en Internet el número de documentos correspondientes a cada forma de la palabra y guardar el resultado en una base de datos.
2. Para cada lema, calcular la sumatoria de los documentos que corresponden a cada de sus formas gramaticales,
3. Para cada forma gramatical, calcular su peso, usando la fórmula:

$$p = n / N$$

donde p es el peso, n es el número de los documentos que existen en Internet para cada forma gramatical, N es el número de documentos que existen en Internet

para el lema (la sumatoria de los números de documentos existentes para cada forma gramatical).

Este proceso tiene un parámetro el cual es el número de los contextos para cada lema. Se escogió el valor de 50 contextos, que da el número de contextos suficientemente grande.

El resultado de la ponderación es el número de contextos para cada forma gramatical del lema, siendo fijo el número de contextos para cada lema.

Por ejemplo, para el lema *ABAD* el cual tiene dos formas gramaticales: *ABAD* y *ABADES*, para la forma *ABAD* existen 57,900 documentos, y para la forma *ABADES* (la cual es mucho menos usada) existen tan sólo 3,310 documentos. Por eso, para la primera forma se buscó 47 contextos, y para la segunda solamente 3 contextos.

7. El corpus compilado

Como el resultado palpable del presente trabajo se obtuvo un corpus representativo de los contextos de palabras en español. El corpus obtenido tiene las siguientes características estadísticas generales, presentes en las Tablas 1 y 2.

Tabla 1. Características generales del corpus.

Propiedad	Valor
Tamaño del corpus, megabytes	221
Número total de palabras encabezado (lemas)	30,198
Número total de formas de palabras encabezado	99,938
Número total de lemas en los contextos (con cadenas no conocidas)	470,417
Número total de formas de palabras en los contextos	556,702
Número promedio de palabras por contexto	37
Número promedio de palabras significativas por contexto	18

El análisis estadístico muestra la siguiente distribución del número de ocurrencias de las palabras en el corpus obtenido, Tabla 2.

Tabla 2. Número de palabras y lemas en el corpus compilado.

Característica	Formas de palabras	Lemas (con cadenas no conocidas)
Total diferentes	556,702	470,417
Máximo	2,302,074 (<i>de</i>)	2,302,074 (<i>de</i>)
Más de 100	23,211	18,402

Característica	Formas de palabras	Lemas (con cadenas no conocidas)
11-100	77,676	38,826
2-10	194,460	162,834
1	261,356	250,357

En la Tabla 2 se presentan varios rangos de frecuencias de las palabras y sus totales respectivas en el corpus.

8. Conclusiones

En este artículo hemos propuesto un método para preparar un diccionario morfológicamente representativo. Nos basamos en la lista de formas gramaticales de palabras ponderadas según sus frecuencias en Internet. Posteriormente, a base de una lista de las palabras iniciales, se generan sus formas morfológicas, se ponderan, y se generan los contextos (concordancias). Hemos construido un corpus así para 30,000 lemas (100,000 formas gramaticales) de palabras de español, que tiene 50 contextos por lema (y proporcional por cada forma).

En el trabajo futuro habrá que realizar el análisis del corpus construido y su comparación con otros corpus. También probar el método propuesto para otros idiomas u para otro número de contextos. Otra dirección de trabajo es aplicar el corpus en varias tareas de PLN.

Agradecimientos. Trabajo realizado con el apoyo parcial del gobierno de México (CONACYT,) e Instituto Politécnico Nacional, México (proyecto SIP 20121823), proyecto FP7-PEOPLE-2010-IRSES: Web Information Quality - Evaluation Initiative (WIQ-EI) European Commission project 269180.

Referencias

1. Manning, C., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge, MA (1999)
2. Gelbukh, A., Sidorov, G.: Procesamiento automático del español con enfoque en recursos léxicos grandes. IPN, 307 p. (2010)
3. Sidorov, G., Gelbukh, A., Chanona-Hernández, L.: Corpus virtual virtual: un diccionario grande de contextos de palabras españolas compilado a través de Internet. In: Proc. of Workshop “Multilingual information access and natural language processing” of IBERAMIA 2002 (8th Iberoamerican conference on Artificial Intelligence), Sevilla, Spain, November, 12, pp 7-14 (2002)
4. Sidorov, G., Barrón-Cedeño, A., Rosso, P.: English-Spanish Large Statistical Dictionary of Inflectional Forms. In: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), Valletta, Malta. European Language Resources Association (ELRA), pp. 277-281 (2010)

5. Carreras, X., Chao, I. Padró L., Padró, M.: FreeLing: An Open-Source Suite of Language Analyzers. In: Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04) (2004)
6. Padró, L., Stanilovsky, E.: FreeLing 3.0: Towards Wider Multilinguality. In: Proceedings of the Language Resources and Evaluation Conference (LREC 2012) ELRA. Istanbul, Turkey (2012)
7. Biber, D.: Representativeness in corpus design. *Literary and linguistic computing*, 8:243-257 (1993)
8. Biber, D., Conrad, S., Reppen, D.: *Corpus linguistics. Investigating language structure and use*. Cambridge University Press, Cambridge (1998)
9. Gelbukh, A., Sidorov, G.: Zipf and Heaps Laws' Coefficients Depend on Language. Proc. CICLing-2001, Conference on Intelligent Text Processing and Computational Linguistics, Mexico City. *Lecture Notes in Computer Science N 2004*, Springer-Verlag, pp. 330–333 (2001)
10. Zipf, G.K.: *Human Behavior and the Principle of Least Effort*. Addison-Wesley Publishers Co., Reading, MA. (1949)

Editorial Board of the Volume

(Comité editorial del volumen)

Ramón Zatarain Cabada, Instituto Tecnológico de Culiacán, México

Carlos A. Reyes García, INAOE, México

María Lucía Barrón Estrada, Instituto Tecnológico de Culiacán, México

María Yasmín Hernández Pérez, Instituto de Investigaciones Eléctricas, México

Rafael Morales Gamboa, Universidad de Guadalajara, México

Jaime Muñoz Arteaga, Universidad Autónoma de Aguascalientes, México

Alejandro Canales Cruz, UNAM, México

Víctor Germán Sánchez Arias, CUAED UNAM, México

Guillermo Rodríguez Ortiz, Instituto de Investigaciones Eléctricas, México

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
diciembre de 2012
Printing 500 / Edición 500 ejemplares

