Advances in Natural Language Processing and Intelligent Learning Environments

Research in Computing Science

RESEARCH IN COMPUTING SCIENCE, *Año 11, Volumen 65, 04 de Febrero de 2011, es una publicación mensual, editada por el Instituto Politécnico Nacional, a través del 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, Ciudad de México, Tel. 57 29 60 00, ext. 56571. https://www.rcs.cic.ipn.mx. Editor responsable: Dr. Grigori Sidorov. Reserva de Derechos al Uso Exclusivo del Título No. 04-2005-121611550100-102. ISSN: en trámite, otorgado por el Instituto Nacional del Derecho de Autor. Responsable de la última actualización de este número: el Centro de Investigación en Computación, Dr. Grigori Sidorov, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738. Fecha de última modificación 04 de Febrero de 2011.*

RESEARCH IN COMPUTING SCIENCE, *Year 11, Volume 65, February 04, 2011, is a monthly publication edited by the National Polytechnic Institute through the Center for Computing Research. Av. Juan de Dios Bátiz S/N, Esq. Miguel Othón de Mendizábal, Nueva Industrial Vallejo, C.P. 07738, Mexico City, Tel. 57 29 60 00, ext. 56571. https://www.rcs.cic.ipn.mx. Editor in charge: Dr. Grigori Sidorov. Reservation of Exclusive Use Rights of Title No. 04-2005-121611550100-102. ISSN: pending, granted by the National Copyright Institute. Responsible for the latest update of this issue: the Computer Research Center, Dr. Grigori Sidorov, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738. Last modified on February 04, 2011.*

Volume 65

Volumen 65

Advances in Natural Language Processing and Intelligent Learning Environments

Grigori Sidorov (Ed.)

SEP
SECRETARÍA DE EDUCACIÓN PÚBLICA

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

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2013

ISSN: 1870-4069

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

Indexed in LATINDEX and Periodica / Indexada en LATINDEX y Periódica

Printing: 500 / Tiraje: 500

Printed in Mexico / Impreso en México

Preface

(Prefacio)

This volume of the journal "Research in Computing Science" contains selected papers on the two topics related to the field of humanities: computational linguistics/natural language processing and intelligent learning environments. The papers were carefully chosen by the editorial board on the basis of the at least two reviews by the members of the reviewing committee.

Both topics intend to model various cognitive abilities of the human beings: computational linguistics models the usage of natural language, while intelligent learning environment draws attention to the learning processes. In both cases the question is: how computers can manage these very complex human activities.

In case of computational linguistics at its modern stage, computers use methods of machine learning to verify linguistic hypotheses over large quantities of data, both supervised learning (more "easy" algorithms, better results, but manually tagged data necessary for training) and unsupervised learning (more "difficult" algorithms, not so good results, but no training data is necessary).

In case of intelligent learning environments, computers are expected to help in choosing the learning strategy by first evaluating the level of a student, after this proposing the optimal learning plan, then giving support during the learning, and at the end applying the final evaluation. Note that the intention is that all these steps would be automatic or at least computer assisted (semi-automatic).

The volume contains six papers on computation linguistics that deal with the themes such as automatic evaluation of automatic text summarization, semantic annotation of social networks, unsupervised dependency parsing, statistical machine translation, general steps in semantic processing, and special type of evaluation of a sentiment classifier.

There are also six papers on intelligent learning environments, related to the themes of learning styles, user interface for learning, virtual reality and affective computing, learning strategies (SCORM) and strategic learning.

It can be noted that there are several paper that analyze the theme of affective computing and sentiment analysis both in computational linguistics and in intelligent learning environments. This is a general trend in the modern artificial intelligence.

November 2013, *Grigori Sidorov*

Table of Contents

(Índice)

Page/Pág.

Natural Language Processing

Entailment-based Fully Automatic Technique
for Evaluation of Summaries 11
Pinaki Bhaskar, Partha Pakray,
Alexander Gelbukh, Sivaji Bandyopadhyay

A Hybrid Approach for Solving the Semantic Annotation Problem
in Semantic Social Networks 25
Pablo Camarillo-Ramírez, J.Carlos Conde-Ramírez,
Abraham Sánchez-López

A Framework for Unsupervised Dependency Parsing
using a Soft-EM Algorithm and Bilexical Grammars 35
Martín Ariel Domínguez, Gabriel Infante-Lopez

A Novel Reordering Model for Statistical Machine Translation 51
Saeed Farzi, Heshaam Faili, Shahram Khadivi, Jalal Maleki

An Approach to Semantic Natural Language Processing of Russian Texts 65
Valery Solovyev, Vladimir Polyakov, Vladimir Ivanov,
Ivan Anisimov, Andrey Ponomarev

Analysis of Asymmetric Measures for Performance Estimation
of a Sentiment Classifier 75
Diego Uribe, Arturo Urquiz, Enrique Cuan

Intelligent Learning Environments

Learning Styles and Emotion Recognition in a Fuzzy Expert System 87
Ramón Zatarain-Cabada, M. Lucía Barrón-Estrada,
Rosalío Zatarain-Cabada

Overview of UI Patterns on Mobile Platform for Educational Applications 95
Raquel Vázquez-Ramírez, Humberto Marín-Vega,
Giner Alor-Hernández

Results of a Case Study of an Intelligent Tutoring System
for Analyzing of Student Projects Presented as Research Papers 103
Jesús Miguel García Gorrostieta, Samuel González López,
Aurelio López-López

SCORM Sequencing and Navigation Model 111
Liliana Argotte, Gustavo Arroyo, Julieta Noguez

Virtual Reality and Affective Computing for Improving Learning 121
Norma J. Ontiveros-Hernández, Miguel Pérez-Ramírez,
Yasmín Hernández

Strategic Learning, towards a Teaching Reengineering 133
Rafaela Blanca Silva-López, Iris Iddaly Méndez-Gurrola,
Victor Germán Sánchez Arias

Natural Language Processing

Entailment-based Fully Automatic Technique for Evaluation of Summaries

Pinaki Bhaskar,[1] Partha Pakray,[2] Alexander Gelbukh,[3] Sivaji Bandyopadhyay[1]

[1] Department of Computer Science and Engineering,
Jadavpur University, Kolkata – 700032, India
[2] Department of Computer and Information System,
Norwegian University of Science and Technology,
Sem Sælandsvei 7-9, NO-7491, Trondheim, Norway
[3] Center for Computing Research,
National Polytechnic Institute, Mexico City, Mexico

{pinaki.bhaskar, parthapakray}@gmail.com,
gelbukh@gelbukh.com, sivaji_cse_ju@yahoo.com

Abstract. We propose a fully automatic technique for evaluating text summaries without the need to prepare the gold standard summaries manually. A standard and popular summary evaluation techniques or tools are not fully automatic; they all need some manual process or manual reference summary. Using recognizing textual entailment (TE), automatically generated summaries can be evaluated completely automatically without any manual preparation process. We use a TE system based on a combination of lexical entailment module, lexical distance module, Chunk module, Named Entity module and syntactic text entailment (TE) module. The documents are used as text (T) and summary of these documents are taken as hypothesis (H). Therefore, the more information of the document is entailed by its summary the better the summary. Comparing with the ROUGE 1.5.5 evaluation scores over TAC 2008 (formerly DUC, conducted by NIST) dataset, the proposed evaluation technique predicts the ROUGE scores with a accuracy of 98.25% with respect to ROUGE-2 and 95.65% with respect to ROUGE-SU4.

Keywords: Automatic text summarization, summary evaluation, recognizing textual entailment.

1 Introduction

Automatic summaries are usually evaluated using human generated reference summaries or some manual efforts. Summaries generated automatically from the documents are difficult to evaluate using completely automatic evaluation process or tool.

The most popular and standard summary evaluation tools are ROUGE and Pyramid. ROUGE evaluate the automated summary by comparing it with the set of human generated reference summary. Whereas Pyramid method needs to identify the nuggets manually. Both the process are very hectic and time consuming. Therefore, automatic evaluation of summary is very important when a large number of summaries are to be evaluated, especially for multi-document summaries. For summary evaluation, we have developed an automated evaluation technique based on textual entailment.

Recognizing Textual Entailment (RTE) is one of the recent research areas of Natural Language Processing (NLP). Textual Entailment is defined as a directional relationship between pairs of text expressions, denoted by the entailing "Text" (T) and the entailed "Hypothesis" (H). T entails H if the meaning of H can be inferred from the meaning of T. Textual Entailment has many applications in NLP tasks, such as Summarization, Information Extraction, Question Answering, Information Retrieval.

There have been seven Recognizing Textual Entailment (RTE) competitions from 2005 to 2011: RTE-1 (Dagan et al., 2005), RTE-2 (Bar-Haim et al., 2006), RTE-3 (Giampiccolo et al., 2007), RTE-4 (Giampiccolo et al., 2008), RTE-5 (Bentivogli et al., 2009), RTE-6 (Bentivogli et al., 2010), and RTE-7 (Bentivogli et al., 2011).

2 Related Work

Most of the approaches in textual entailment domain take Bag-of-words representation as one option, at least as a baseline system. The system by Herrera et al. (2005) obtains lexical entailment relations from WordNet[1]. The lexical unit T entails the lexical unit H if they are synonyms, Hyponyms, Multiwords, Negations and Antonyms according to WordNet or if there is a relation of similarity between them. The system accuracy was 55.8% on RTE-1 test dataset.

Kouylekov and Magnini (2005) used a tree-edit distance algorithm applied to the dependency trees of the text and the hypothesis. If the distance (i.e., the cost of the editing operations) among the two trees is below a certain threshold, empirically estimated on the training data, then an 'YES' entailment relation is assigned between the two texts. The system accuracy was 55.9% on the RTE-1 test dataset.

Based on the idea that meaning is determined by context Clarke (2006) proposed a formal definition of entailment between two sentences in the form of a conditional probability on a measure space. The system submitted in RTE-4 provided three practical implementations of this formalism: a bag of words comparison as a baseline and two methods based on analyzing sub-sequences of the sentences possibly with intervening symbols. The system accuracy was 53% on RTE-2 test dataset.

Adams et al. (2007) used linguistic features as training data for a decision tree classifier. These features were derived from the text–hypothesis pairs under examination. The system mainly used ROUGE (Recall–Oriented Understudy for Gisting Evaluation), N-gram overlap metrics, Cosine Similarity metric and WordNet based measure as features. The system accuracy was 52% on RTE-2 test dataset.

[1] http://wordnet.princeton.edu/

In RTE-3, Newman et al. (2006) presented two systems for textual entailment, both employing decision tree as a supervised learning algorithm. The first one is based primarily on the concept of lexical overlap, considering a bag of words similarity overlap measure to form a mapping of terms in the hypothesis to the source text. The accuracy of the system improved to 67% on the RTE-3 test set.

Montalvo-Huhnet al. (2008) guessed at entailment based on word similarity between the hypotheses and the text. Three kinds of comparisons were attempted: original words (with normalized dates and numbers), synonyms and antonyms. Each of the three comparisons contributes a different weight to the entailment decision. The two-way accuracy of the system was 52.6% on RTE-4 test dataset.

Litkowski's (2009) system consists solely of routines to examine the overlap of discourse entities between the texts and hypotheses. The two-way accuracy of the system was 53% on RTE-5 Main task test dataset.

Majumdarand Bhattacharyya (2010) describes a simple lexical based system that detects entailment based on word overlap between the Text and Hypothesis. The system is mainly designed to incorporate various kinds of co-referencing that occur within a document and take an active part in the event of Text Entailment. The accuracy of the system was 47.56% on RTE-6 Main Task test dataset.

Dependency tree structures of input sentences are widely used by many research groups, since it provides more information with quite good robustness and runtime than shallow parsing techniques. Basically, a dependency parsing tree contains nodes (i.e., tokens/words) and dependency relations between nodes. Some approaches simply treat it as a graph and calculate the similarity between the text and the hypothesis graphs solely based on their nodes, while some others put more emphasis on the dependency relations themselves.

The system described by Herrera et al. (2005) is based on the use of a broad-coverage parser to extract dependency relations and a module that obtains lexical entailment relations from WordNet. The work compares whether the matching of dependency tree substructures give better evidence of entailment than the matching of plain text alone. The system accuracy was 56.6% on RTE-1 test set.

The MENT (Microsoft ENTailment) (Vanderwende et al., 2006) system predicts entailment using syntactic features and a general-purpose thesaurus, in addition to an overall alignment score. MENT is based on the premise that it is easier for a syntactic system to predict false entailments. It achieved accuracy of 60.25% on RTE-2 test set.

Wangand Neumannm (2007) present a novel approach to RTE that exploits a structure-oriented sentence representation followed by a similarity function. The structural features are automatically acquired from tree skeletons that are extracted and generalized from dependency trees. The method makes use of a limited size of training data without any external knowledge bases (e.g., WordNet) or handcrafted inference rules. They achieved an accuracy of 66.9% on the RTE-3 test data.

The major idea of Varmaet al. (2009) is to find linguistic structures, termed templates that share the same anchors. Anchors are lexical elements describing the context of a sentence. Templates that are extracted from different sentences (text and hypothesis) and connect the same anchors in these sentences are assumed to entail each other. The system accuracy was 46.8% on RTE-5 test set.

Tsuchida and Ishikawa (2011) combine the entailment score calculated by lexical-level matching with the machine-learning based filtering mechanism using various features obtained from lexical-level, chunk-level and predicate argument structure-level information. In the filtering mechanism, the false positive T-H pairs that have high entailment score but do not represent entailment are discarded. The system accuracy was 48% on RTE-7 test set.

Lin and Hovy (2003) developed an automatic summary evaluation system using *n*-gram co-occurrence statistics. Following the recent adoption by the machine translation community of automatic evaluation using the BLEU/NIST scoring process, they conduct an in-depth study of a similar idea for evaluation of summaries. They showed that automatic evaluation using unigram co-occurrences between summary pairs correlates surprisingly well with human evaluations, based on various statistical metrics, while direct application of the BLEU evaluation procedure does not always give good results.

Harnly et al. (2005) also proposed an automatic summary evaluation technique by the Pyramid method. They presented an experimental system for testing automated evaluation of summaries, pre-annotated for shared information. They reduced the problem to a combination of similarity measure computation and clustering. They achieved best results with a unigram overlap similarity measure and single link clustering, which yields high correlation to manual pyramid scores ($r = 0.942$, $p = 0.01$), and shows better correlation than the *n*-gram overlap automatic approaches of the ROUGE system.

3 Textual Entailment System

In this section we describe a two-way hybrid textual entailment (TE) recognition system that uses lexical and syntactic features. The system architecture is shown in Figure 1.

The hybrid TE system used the Support Vector Machine Learning technique that uses thirty-four features for training. Five features from Lexical TE, seventeen features from Lexical distance measure and eleven features from the rule based syntactic two-way TE system were selected.

3.1 Lexical Similarity

In this subsection, the various lexical features for textual entailment are described in detail.

WordNet based Unigram Match. In this method, the various unigrams in the hypothesis for each text-hypothesis pair are checked for their presence in text. WordNet synset are identified for each of the unmatched unigrams in the hypothesis. If any synset for the hypothesis unigram matches with any synset of a word in the text then the hypothesis unigram is considered as a WordNet based unigram match.

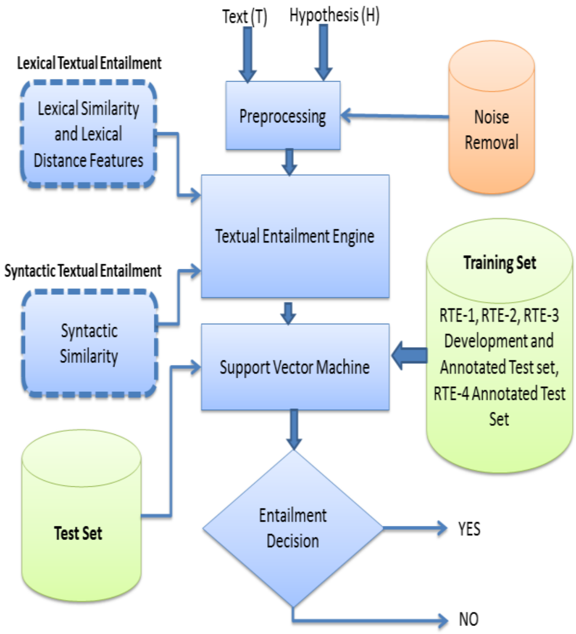


Fig. 1. Hybrid Textual Entailment System

Bigram Match. Each bigram in the hypothesis is searched for a match in the corresponding text part. The measure Bigram_Match is calculated as the fraction of the hypothesis bigrams that match in the corresponding text, i.e.,

$$\text{Bigram\_Match} = \frac{\text{Total number of matched bigrams in a text} - \text{hypothesis pair}}{\text{Number of hypothesis bigrams}}.$$

Longest Common Subsequence (LCS). The Longest Common Subsequence of a text-hypothesis pair is the longest sequence of words which is common to both the text and the hypothesis. LCS(T,H) estimates the similarity between text T and hypothesis H, as

$$\text{LCS\_Match} = \frac{\text{LCS}(\text{T}, \text{H})}{\text{length of H}}.$$

Skip-grams. A skip-gram is any combination of n words in the order as they appear in a sentence, allowing arbitrary gaps. In the present work, only 1-skip-bigrams are considered where 1-skip-bigrams are bigrams with one word gap between two words in order in a sentence. The measure 1-skip_bigram_Match is defined as

$$\text{1\_skip\_bigram\_Match} = \frac{\text{skip\_gram}(\text{T}, \text{H})}{n},$$

where skip_gram(T,H) refers to the number of common 1-skip-bigrams (pair of words in sentence order with one word gap) found in T and H and n is the number of 1-skip-bigrams in the hypothesis H.

Stemming. Stemming is the process of reducing terms to their root forms. For example, the plural forms of a noun such as ‘boxes’ are stemmed into ‘box’, and inflectional endings with ‘ing’, ‘es’, ‘s’ and ‘ed’ are removed from verbs. Each word in the text and hypothesis pair is stemmed using the stemming function provided along with the WordNet 2.0.

If s_1 is the number of common stemmed unigrams between text and hypothesis and s_2 is the number of stemmed unigrams in Hypothesis, then the measure Stemming_Match is defined as

$$\text{Stemming\_Match} = \frac{s_1}{s_2}.$$

WordNet is one of most important resource for lexical analysis. WordNet 2.0 has been used for WordNet based unigram match and stemming step. API for WordNet Searching[2] (JAWS) is an API that provides Java applications with the ability to retrieve data from the WordNet database.

3.2 Syntactic Similarity

In this section, various syntactic similarity features for textual entailment are described in detail. This module is based on the Stanford Dependency Parser[3], which normalizes data from the corpus of text and hypothesis pairs, accomplishes the dependency analysis and creates appropriate structures Our Entailment system uses the following features.

Subject. The dependency parser generates nsubj (nominal subject) and nsubjpass (passive nominal subject) tags for the subject feature. Our entailment system uses these tags.

Object. The dependency parser generates dobj (direct object) as object tags.

Verb. Verbs are wrapped with either the subject or the object.

Noun. The dependency parser generates NN (noun compound modifier) as noun tags.

Preposition. Different types of prepositional tags are prep_in, prep_to, prep_with etc. For example, in the sentence “A plane crashes in Italy.” the prepositional tag is identified as prep_in(in, Italy).

---

[2] http://wordnetweb.princeton.edu/perl/webwn
[3] http://www-nlp.stanford.edu/software/lex-parser.shtml

Determiner. Determiner denotes a relation with a noun phase. The dependency parser generates det as determiner tags. For example, the parsing of the sentence “A journalist reports on his own murders.” generates the determiner relation as det(journalist,A).

Number. The numeric modifier of a noun phrase is any number phrase. The dependency parser generates num (numeric modifier). For example, the parsing of the sentence “Nigeria seizes 80 tonnes of drugs.” generates the relation num (tonnes, 80).

Matching Module. After dependency relations are identified for both the text and the hypothesis in each pair, the hypothesis relations are compared with the text relations. The different features that are compared are noted below. In all the comparisons, a matching score of 1 is considered when the complete dependency relation along with all of its arguments matches in both the text and the hypothesis. In case of a partial match for a dependency relation, a matching score of 0.5 is assumed.

Subject-Verb Comparison. The system compares hypothesis subject and verb with text subject and verb that are identified through the nsubj and nsubjpass dependency relations. A matching score of 1 is assigned in case of a complete match. Otherwise, the system considers the following matching process.

WordNet Based Subject-Verb Comparison. If the corresponding hypothesis and text subjects do match in the subject-verb comparison, but the verbs do not match, then the WordNet distance between the hypothesis and the text is compared. If the value of the WordNet distance is less than 0.5, indicating a closeness of the corresponding verbs, then a match is considered and a matching score of 0.5 is assigned. Otherwise, the subject-subject comparison process is applied.

Subject-Subject Comparison. The system compares hypothesis subject with text subject. If a match is found, a score of 0.5 is assigned to the match.

Object-Verb Comparison. The system compares hypothesis object and verb with text object and verb that are identified through DObj dependency relation. In case of a match, a matching score of 0.5 is assigned.

WordNet Based Object-Verb Comparison. The system compares hypothesis object with text object. If a match is found then the verb associated with the hypothesis object is compared with the verb associated with the with text object. If the two verbs do not match then the WordNet distance between the two verbs is calculated. If the value of WordNet distance is below 0.50 then a matching score of 0.5 is assigned.

Cross Subject-Object Comparison. The system compares hypothesis subject and verb with text object and verb or hypothesis object and verb with text subject and verb. In case of a match, a matching score of 0.5 is assigned.

Number Comparison. The system compares numbers along with units in the hypothesis with similar numbers along with units in the text. Units are first compared and if they match then the corresponding numbers are compared. In case of a match, a matching score of 1 is assigned.

Noun Comparison. The system compares hypothesis noun words with text noun words that are identified through NN dependency relation. In case of a match, a matching score of 1 is assigned.

Prepositional Phrase Comparison. The system compares the prepositional dependency relations in the hypothesis with the corresponding relations in the text and then checks for the noun words that are arguments of the relation. In case of a match, a matching score of 1 is assigned.

Determiner Comparison. The system compares the determiners in the hypothesis and in the text that are identified through Det relation. In case of a match, a matching score of 1 is assigned.

Other relation Comparison. Besides the above relations that are compared, all other remaining relations are compared verbatim in the hypothesis and in the text. In case of a match, a matching score of 1 is assigned.

3.3 Part-of-Speech (POS) Matching

This module basically matches common POS tags between the text and the hypothesis pairs. Stanford POS tagger[4] is used to tag the part of speech in both text and hypothesis. System matches the verb and noun POS words in the hypothesis with those in the text. A score POS_match is defined as follows:

$$\text{POS\_Match} = \frac{\text{number of verb and noun matched in Text and Hypothesis}}{\text{total number of verbs and nouns in Hypothesis}}. \quad (1)$$

3.4 Lexical Distance

The important lexical distance measures that are used in the present system include Vector Space Measures (Euclidean distance, Manhattan distance, Minkowsky distance, Cosine similarity, Matching coefficient), Set-based Similarities (Dice, Jaccard, Overlap, Cosine, Harmonic), Soft-Cardinality, Q-Grams Distance, Edit Distance Measures (Levenshtein distance, Smith-Waterman Distance, Jaro).

[4] http://nlp.stanford.edu/software/tagger.shtml

3.5 Chunk similarity

The part of speech (POS) tags of the hypothesis and text are identified using the Stanford POS tagger. After getting the POS information, the system extracts the chunk output using the CRFChunker[5]. Chunk boundary detector detects each individual chunk such as noun chunk, verb chunk etc. Thus, all the chunks for each sentence in the hypothesis are identified. Each chunk of the hypothesis is now searched in the text side and the sentences that contain the key chunk words are extracted. If chunks match then the system assigns scores for each individual chunk corresponding to the hypothesis. The scoring values are changed according to the matching of chunk and word containing the chunk. The entire scoring calculation is given in (2) and (3):

$$\text{Match score } M[i] = \frac{W_m[i]}{W_c[i]}, \tag{2}$$

where $W_m[i]$ is the number of words that match in the i-th chunk and $W_c[i]$ is the total number of words containing the i-th chunk;

$$\text{Overall score S} = \sum_{i=1}^{N} \frac{M\,[i]}{\text{N}}. \tag{3}$$

where N is the total number of chunks in the hypothesis.

3.6 Support Vector Machines (SVM).

In machine learning, support vector machines (SVMs)[6] are supervised learning models used for classification and regression analysis. Associated learning algorithms analyze data and recognize patterns. The basic SVM takes a set of input data and predicts, for each given input, which of two possible classes form the output, making it a non-probabilistic binary linear classifier. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

The LIBSVM[7] tool was used to find the textual entailment relation. The system has used LIBSVM for building the model file. The TE system has used the following data sets: RTE-1 development and test set, RTE-2 development and annotated test set, RTE-3 development and annotated test set and RTE-4 annotated test set to deal with the two-way classification task for training purpose to build the model file. The LIBSVM tool is used by the SVM classifier to learn from this data set. For training purpose, 3967

[5] http://crfchunker.sourceforge.net/

[6] http://en.wikipedia.org/wiki/Support_vector_machine

[7] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

text-hypothesis pairs were used. It has been tested on the RTE test dataset and we obtained 60% to 70% accuracy on RTE datasets. We have applied this textual entailment system on summarize data sets and system gives the entailment score with entailment decisions (i.e., "YES" / "NO"). We have tested in both directions.

4 Automatic Evaluation of Summary

Ideally, summary of some documents should contain all the necessary information contained in the documents. So the quality of a summary should be judged on how much information of the documents it contains. If the summary contains all the necessary information from the documents, then it will be a perfect summary. Yet manual comparison is the best way to judge that how much information it contains from the document. However, manual evaluation is a very hectic process, especially when the summary generated from multiple documents. When a large number of multi-document summaries have to be evaluated, then an automatic evaluation method needs to evaluate the summaries. Here we propose textual entailment (TE) based automatic evaluation technique for summary.

4.1 Textual Entailment based Summary Evaluation

Textual Entailment is defined as a directional relationship between pairs of text expressions, denoted by the entailing "Text" (T) and the entailed "Hypothesis" (H). Text (T) entails hypothesis (H) if the information of text (T) is inferred into the hypothesis (H). Here the documents are used as text (T) and summary of these documents are taken as hypothesis (H). Therefore, if the information of documents is entailed into the summary then it will be a very good summary, which should get a good evaluation score.

As our textual entailment system works on sentence level each sentence of documents are taken as text (T) and calculate the entailment score comparing with each sentence of the summary assuming them as hypothesis (H). For example, if T_i is the i^{th} sentence of documents, then it will compared with each sentence of the summary, i.e. H_j, where j = 1 to n; and n is the total number of sentences in the summary. Now if T_i is validated with any one of the summary sentences using our textual entailment system, then it will be marked as validated. After get the entailment result of all the sentences of documents, the percentage or ratio of the marked/validated sentences with respect to unmarked / rejected sentences will be the evaluation score of the summary.

5 Data Collection

We used the Text Analysis Conference (TAC, formerly DUC, conducted by NIST) 2008 Update Summarization track's datasets8 for this experiment. This dataset contains 48 topics and each topic has two sets of 10 documents, i.e. there are 960 documents.

[8] http://www.nist.gov/tac/data/index.html

The evaluation data set has four model summaries for each document set, i.e. 8 model summaries for each topic. In 2008, there are 72 participants, and we used the summaries of all the participants of this year.

6 Comparison of Automatic vs. Manual Evaluation

We considered the evaluation scores of all the 72 participants of TAC 2008 using ROUGE 1.5.5. We calculated the evaluation scores of the same summaries of 72 participants using the proposed automated evaluation technique and compared it with ROUGE scores. The comparison of the evaluation scores on top five participants is shown in the Table 1.

Table 1. Comparison of Summary Evaluation Score

Evaluation method	ROUGE-2 Average_R	ROUGE-SU4 Average_R	Proposed method
Top ranked participant (id:43)	0.111	0.143	0.7063
2nd ranked participant (id:13)	0.110	0.140	0.7015
3rd ranked participant (id:60)	0.104	0.142	0.6750
4th ranked participant (id:37)	0.103	0.143	0.6810
5th ranked participant (id:6)	0.101	0.140	0.6325

For measuring the accuracy of our proposed method, we consider the ROUGE 1.5.5 evaluation score as the gold standard score and then calculate the accuracy of this proposed method using (4):

$$\text{Accuracy} = 1 - \frac{\sum_{i=1}^{n} |(r_i - r_i^R)|}{n^2} \tag{4}$$

where r_i is the rank of i-th summary after evaluated by the proposed method, r_i^R is the rank of i-th summary after evaluated by ROUGE 1.5.5, and n is the total number of multi-document summaries.

After evaluating 48 (only set A) multi-document summaries of 72 participants, i.e total 3456 multi-document summaries using the evaluation method, ROUGE 1.5.5 and the proposed method, the accuracy of this proposed method calculated using (4) comparing with the ROUGE's evaluation scores. The accuracy figures are 0.9825 with respect to ROUGE-2 and 0.9565 with respect to ROUGE-SU4.

7 Conclusions

Evaluating summaries automatically is very useful in batch processing. From the comparison of evaluation scores of the proposed method and those of ROUGE 1.5.5, it is clear that our method can predict the ROUGE ranking. However, ROUGE requires manually preparing the gold standard summaries, which is a very time-consuming task. In contrast, our method is completely automatic.

In our future work, we plan to explore the use of syntactic *n*-grams, which have been shown to be useful on other automatic evaluation tasks (Sidorov et al., 2013).

Acknowledgments. We acknowledge the support of the DST India-CONACYT Mexico project "Answer Validation through Textual Entailment" funded by DST, Government of India and the Department of Electronics and Information Technology (DeitY), Ministry of Communications & Information Technology (MCIT), Government of India funded project "Development of Cross Lingual Information Access (CLIA) System Phase II", and SIP-IPN project 20131702. We also acknowledge the support of the Department of Computer and Information Science, Norwegian University of Science and Technology, and ABCDE Fellowship Programme 2012-1013.

References

1. Adams, R., Nicolae, G., Nicolae, C. and Harabagiu, S. (2007): Textual Entailment Through Extended Lexical Overlap and Lexico-Semantic Matching. In Proceedings of the ACL PASCAL Workshop on Textual Entailment and Paraphrasing. 28–29 June, Prague, Czech Republic, pp. 119–124.
2. Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B., Szpektor, I. (2006): The Second PASCAL Recognising Textual Entailment Challenge. Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Venice, Italy.
3. Bentivogli, L., Dagan, I., Dang. H.T., Giampiccolo, D., Magnini, B. (2009): The Fifth PASCAL Recognizing Textual Entailment Challenge, In TAC 2009 Workshop, National Institute of Standards and Technology Gaithersburg, Maryland USA.
4. Bentivogli, L., Clark, P., Dagan, I., Dang, H. T., Giampiccolo, D. (2010): The Sixth PASCAL Recognizing Textual Entailment Challenge. In TAC 2010 Notebook Proceedings.
5. Bentivogli, L., Clark, P., Dagan, I., Dang, H. T., Giampiccolo, D. (2009): The Seventh PASCAL Recognizing Textual Entailment Challenge. In TAC 2011 Notebook Proceedings.
6. Clarke, D. (2006): Meaning as Context and Subsequence Analysis for Entailment. In Proceedings of the Second PASCAL Recognising Textual Entailment Challenge, Venice, Italy.
7. Dagan, I., Glickman, O., Magnini, B. (2005): The PASCAL Recognising Textual Entailment Challenge Proceedings of the First PASCAL Recognizing Textual Entailment Workshop.
8. Giampiccolo, D., Magnini, B., Dagan, I., Dolan, B. (2007): The Third PASCAL Recognizing Textual Entailment Challenge, In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, Prague, Czech Republic.
9. Giampiccolo, D., Dang, H. T., Magnini, B., Dagan, I., Cabrio, E. (2008): The Fourth PASCAL Recognizing Textual Entailment Challenge. In TAC 2008 Proceedings.
10. Harnly, A., Nenkova, A., Passonneau, R., Rambow, O. (2005): Automation of summary evaluation by the pyramid method. Recent Advances in Natural Language Processing (RANLP). Borovets, Bulgaria.
11. Herrera, J., Peas, A. Verdejo, F. (2005): Textual Entailment Recognition Based on Dependency Analysis and WordNet. In Proceedings of the First Challenge Workshop Recognising Textual Entailment, Pages 21-24, 33–36 April 2005, Southampton, U.K.
12. Kouylekov, M., Magnini, B. (2005): Recognizing Textual Entailment with Tree Edit Distance Algorithms. Proceedings of the First PASCAL Recognizing Textual Entailment Workshop.

13. Lin, C. Y., Hovy, E. (2003). Automatic evaluation of summaries using n-gram co-occurrence statistics. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology. Volume 1. Association for Computational Linguistics, pp. 71–78.
14. Litkowski, K. (2009): Overlap Analysis in Textual Entailment Recognition. In TAC 2009 Workshop, National Institute of Standards and Technology Gaithersburg, Maryland USA.
15. Majumdar, D. Bhattacharyya, P. (2010): Lexical Based Text Entailment System for Summarization Settings of RTE6.Proceedings of the Text Analysis Conference (TAC 2010) November 15–16, 2010 National Institute of Standards and Technology Gaithersburg, Maryland, USA.
16. Montalvo-Huhn, O. Taylor, S. (2008): Textual Entailment – Fitchburg State College. In Proceedings of TAC08, Fourth PASCAL Challenges Workshop on Recognising Textual Entailment.
17. Newman, E., Dunnion, J., Carthy, J. (2006): Constructing a Decision Tree Classifier using Lexical and Syntactic Features. In Proceedings of the Second PASCAL Recognising Textual Entailment Challenge.
18. Sidorov, G., Gupta, A., Tozer, M., Catala, D., Catena, A., Fuentes, S (2013): Rule-based System for Automatic Grammar Correction Using Syntactic N-grams for English Language Learning (L2). In: Proc. ACL 2013, 6 p.
19. Tsuchida, M., Ishikawa, K. (2011): IKOMA at TAC2011: A Method for Recognizing Textual Entailment using Lexical-level and Sentence Structure-level features. In TAC 2011 Notebook Proceedings.
20. Vanderwende, L., Menezes, A., Snow, R. (2006): Microsoft Research at RTE-2: Syntactic Contributions in the Entailment Task: an implementation. In Proceedings of the Second PASCAL Challenges Workshop.
21. Varma, V., Bharat, V., Kovelamudi, S., Bysani, P., GSK, S., N, K. K., Reddy, K., Kumar, K., Maganti, N. (2009): IIIT Hyderabad at TAC 2009. In TAC 2009 Workshop, National Institute of Standards and Technology Gaithersburg, Maryland USA.
22. Wang, R., Neumannm G. (2007): Recognizing Textual Entailment Using Sentence Similarity based on Dependency Tree Skeletons. In Proceedings of the Third PASCAL Recognising Textual Entailment Challenge.

A Hybrid Approach for Solving the Semantic Annotation Problem in Semantic Social Networks

Pablo Camarillo-Ramírez, J.Carlos Conde-Ramírez, Abraham Sánchez-López

Facultad de Ciencias de la Computación
Benemérita Universidad Autónoma de Puebla, Mexico
{pablo.camarillo,juanc.conde,asanchez}@cs.buap.mx

Abstract. In this paper, we propose a hybrid method that gives a solution for the semantic annotation problem. We focus our approach to settle the semantic annotation in social networks. Many approaches use a kind of knowledge representation as taxonomies or ontologies to resolve the annotation problem. Recent works have proposed other probabilistic-based approaches to solve the semantic problem as Bayesian Networks. The nature of the Bayesian learning is given by two phases: the data gathering and the query phase, it can be used to settle the semantic annotation problem viewed as a classification one. This work proposes to combine an ontological approach with a Bayesian learning one applied to give a semantic to publications realized in real time in social networks.

Keywords: Semantic annotation, social networks.

1 Introduction

In this paper it is presented a hybrid method that solves the semantic annotation process in semantic social networks. In the study of the Semantic Web, the aim is to describe the content of annotating resources with unambiguous information to facilitate the exploitation of these resources with software agents [1]. Semantic annotation is the process in which one can relate the Web content with a specific knowledge representation. The method described in this paper consist in use an ontology and a Bayesian Network in order to give a semantic for the textual publications in semantic social networks. The firs method of the proposed strategy work uses an ontology to extract the information of publications in social networks. The second method uses a Bayesian Network to classify publications that can not be annotated by using the ontology method.

In [2], we presented an early approach based in Bayesian Networks to classify publications in social networks. Our new work consist in combine the ontology-based-strategy and the Bayesian-Network-method presented in [2]. The section 2 describes briefly the related work about semantic annotation and ontologies in semantic Web, the section 3 presents the generalities of Bayesian networks, the proposed strategy to solve the semantic annotation problem is discussed in the section 4, then, the section 5 shows the experiments realized and the results obtained, and finally our conclusions about the results obtained by combining Bayesian Networks method and the Ontology method are remarked in section 6, also the future work to improve our work.

2 Semantic Web, Semantic Annotation and Semantic Social Networks

The Semantic Web can be understood like the idea to bring structure to the meaningful content of Web pages, creating an environment where software agents can suspect user's needs and more[3].

Since the beginning of the century, the Web has been changing into social Web. According to [4] the Web is becoming more and more social, we are now collecting huge amount of knowledge on-line. Semantic Web researchers propose making Web content machine understandable through the use of ontologies, which are commonly shared, explicitly defined, generic conceptualizations [5]. But one of the most important problems we face is the way that make possible that machines can understand the content of the Web, the semantic annotation problem.

According to [1], the goal of semantic annotation is to add comments to Web content so that is becomes machine understandable. Unlike an annotation in the normal sense, which is an unrestricted note, a semantic annotation must be explicit, formal, an 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 an ontology or any classification method makes this possible. A Semantic annotation tool is a kind of software that allows add and manage semantic annotations linked to at least one given documentary resource. In the Semantic Web context, the annotation tool can use an ontology or at least one formal model in order to formalize and organize annotations produced by the restrictions defined in this ontology.

The term semantic social networks was coined independently by Stephen Downes and Marco Neumann in 2004 to describe the application of semantic Web technologies and online social networks [6]. In these sense, in [7] it is proposed a three-layered-model which involves the network between people (social network), the network between the ontologies (ontology network) and a network between concepts occurring in these ontologies. In the semantic social network described in [8], authors use an ontology to represent the knowledge that is used to the annotation process.

3 Ontologies and Bayesian Networks

An ontology is a formal conceptualization of certain domain, the description of its concepts and its relations [9,10]. Ontologies are domain models with special characteristics that drive to the idea of shared meaning or semantic. Ontologies are expressed with formal languages with a well-defined semantic. They are based in a shared comprehension with the common. In our work we take advantage of the Ontology proposed in [11] in which a scientific domain was proposed and modeled. This Ontology contains 64 classes and 108 terms related to scientific domain, with this ontology, the ontology-based method used in this work describe a semantic for textual publications.

The core of this paper is the use of the Bayesian Networks (BN) to *classify* the social networks publications and in this way to give a semantic of textual publications in a semantic social network. BN are a powerful knowledge representation and reasoning mechanism. Formally, BN are directed acyclic graph (DAG) whose nodes represent random variables in the Bayesian sense: they may be observable quantities, latent variables, unknown parameters or hypotheses. Edges represent conditional dependencies; in this sense, nodes which are not connected represent variables which are conditionally independent of each other. Each node is associated with a probability function that takes as input a particular set of values of the node's parent variables and gives the probability of the variable represented by the node [12]. In this work the edges of the DAG represent the most representative terms obtained from the gain information strategy used in [2] for building the BN.

4 The Proposed Hybrid Method

The proposal in this work consist in combine two strategies for solve the problem of semantic annotations in semantic social networks. The strategy proposed in [11] gives an approximation for solve the semantic annotation problem with an ontology-based method, however not all publications can be annotated with this approach, because the information extraction method only can annotate such publications that contain words present in the ontology individuals. Therefore, our proposal consist in use the results given by the ontology-based method as the evidence needed for build a BN with the methodology presented in [2]. Once we have the BN, it can be used to classify publications that can not be annotated with the ontology-based method. With this hybridization, all publications can be associated with a semantic. In Figure 1 it is shown the flow of our strategy.

4.1 Ontology-based Method

The semantic social network presented in [11] used an ontology to describe the semantic of textual publications (sentences) published. When the users create a publication, an annotation tool is invoked in order to check if the ontology contains any concept presented in the published sentence. If so, the annotation system associates the publication with the parent class of the concept found in the ontology structure. Otherwise the publication is not annotated. The Figure 2 shows a part of the taxonomy of the complete ontology used in this work.

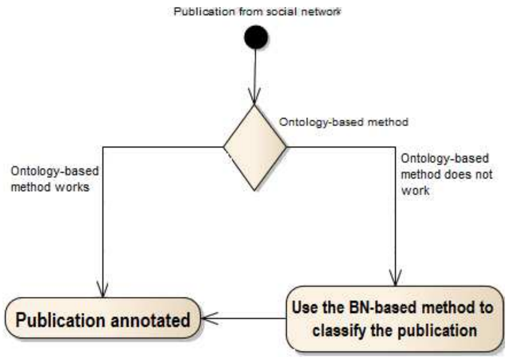


Fig. 1. Strategy flow

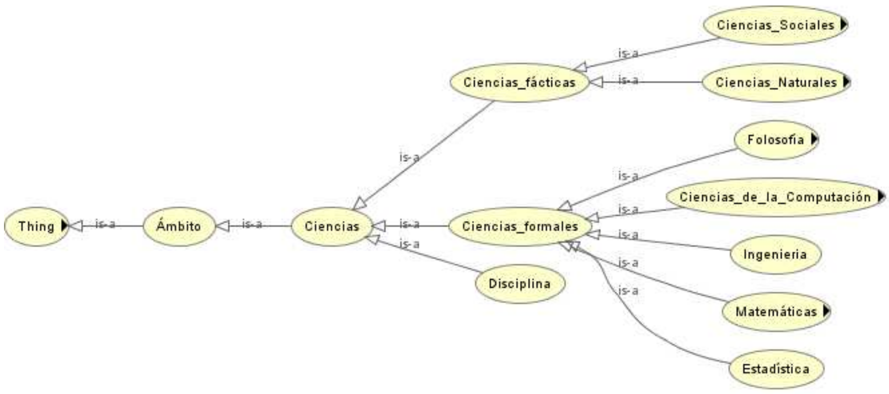


Fig. 2. Part of the ontology taxonomy

4.2 BN-based Method

According with [13], it can be identified three phases in machine learning: data gathering, learning and query phases. In [2] these phases have been implemented as follows.

Data Gathering It has been told that the data gathering must be provide examples, these examples are given by the annotated publications by the ontology-based method, each annotated publication has associated a class. As part of *Data preparation* process, all publications obtained (annotated or not) were subjected to two filters: Terms count and Data binarization. In the first one, a vector is build from the vocabulary, each term in the vocabulary is counted and then an IDF transformation is applied in order to improve the vectorization process. The second one, is a filter applied by using every single publication: each term is converted into binary form depending if each publication contains or not every term in the vocabulary.

Learning Once we have the statistics from annotated publications, it can be possible build the BN. It has been done by using the K2 algorithm proposed in 1992 by Cooper [14]. This method receive the set of the most representative terms and its frequency for building the network. The K2 Algorithm produces a BN in which all terms(nodes) have its respective parents depending of the evidence given by the statistics obtained in the Data gathering phase. A BN as show in Figure 3, is the result of this phase. In this BN, the nodes are considered discrete random variables because they can take two values:1 in case of the sentence that contains the related term to this node, or 0 in otherwise. The BN is produced in a XMLBIF format (XML-based BayesNets Interchange Format) [1].

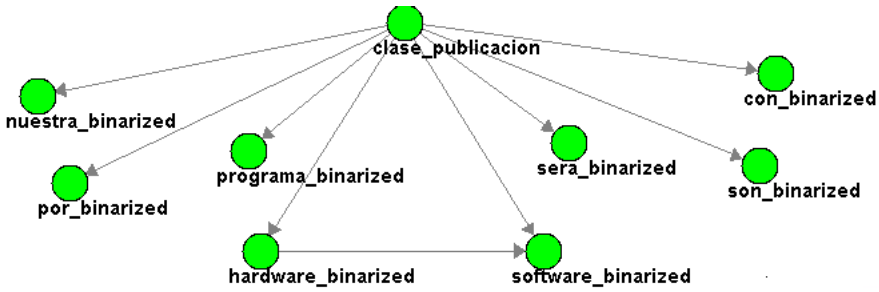


Fig. 3. BN produced by the K2 Algorithm

Query This process is one of the most important contribution in this work. We proposed and implemented a method that *automatically* classify any publication given through a query to the BN generated in the learning phase. The classification algorithm used for classify each incoming publication is the Naïve Bayes Algorithm. Figure 4 resume this phase. First, it is necessary to prepare each publication by removing all Spanish punctuation marks. Then, an evidence must be assigned in the generated BN before, giving the correct value for each node in the BN: if the term represented in the node exists in the incoming publication, the value for this node is 1, otherwise 0. Any value for the node give us enough evidence to obtain a conclusion. The Algorithm 1 explains this phase in detail. The most important aspect in this approach is that the absence (is not present) or the presence (is present) of evidence contributes to the computing of causal probability. With this method any publication can be classified, because always is possible get a trust classification. Finally, when the evidence has been assigned, it can be possible to make an inference through the given evidence. To

[1] The format has been designed primarily by Fabio Cozman, with important contributions from Marek Druzdzel and Daniel Garcia. The XMLBIF format is very simple to understand and can yet represent directed acyclic graphs with probabilistic relations, decision variables and utility values

get the inference needed is necessary to apply the *Variable Elimination Algorithm* to the root of the BN. This query give us the probability of each variable (class), with this information is possible to make a choice about the class of the given publication. The class that contains the maximum probability will be chosen as the class of this publication. This process has been implemented as a Web service. This Web service is invoked by the semantic social network when the ontology-based method does not annotate any incoming publication.

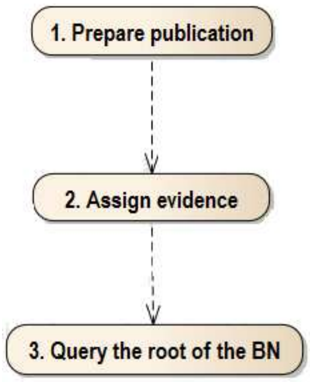


Fig. 4. The Query Process

Algorithm 1 Assign evidence Algorithm

```
for all node N ∈ Bayesian Network do
    if publication contains the node N then
        assign the value ”is present” to node N
    else
        assign the value ”is not present” to node N
    end if
end for
```

5 Experiments and Results

It has been obtained almost one hundred of publications from the social network Moveek [11](the collected data contains publications only in Spanish language).

However, for our study we have been used only the 40% of publications, i.e., the most representative. These annotated publications give us the statistics needed to build correctly the BN. Before applying our annotation strategy, the obtained publications have the distribution shown in Table 1.

Table 1. Ontology-based method accuracy

Annotation success	**Percentage**
Annotated publications	68%
Not Annotated publications	32%

Once that the strategy has been embedded in the semantic social network, it has been taken a sample of new publications in order to proof our strategy. Of course, all publications were annotated by applying the ontology-based method and the BN-based method. The Table 2 summarized the obtained results.

Table 2. Use of annotation methods

Annotation method used	**Percentage**
Ontology-based method	37.5%
BN-based method	62.5%

The results presented in Table 2 show that the new implemented BN-based method is used more times than the Ontology-based method. In fact, the 62.5% of publications could not be annotated without the BN-based method.

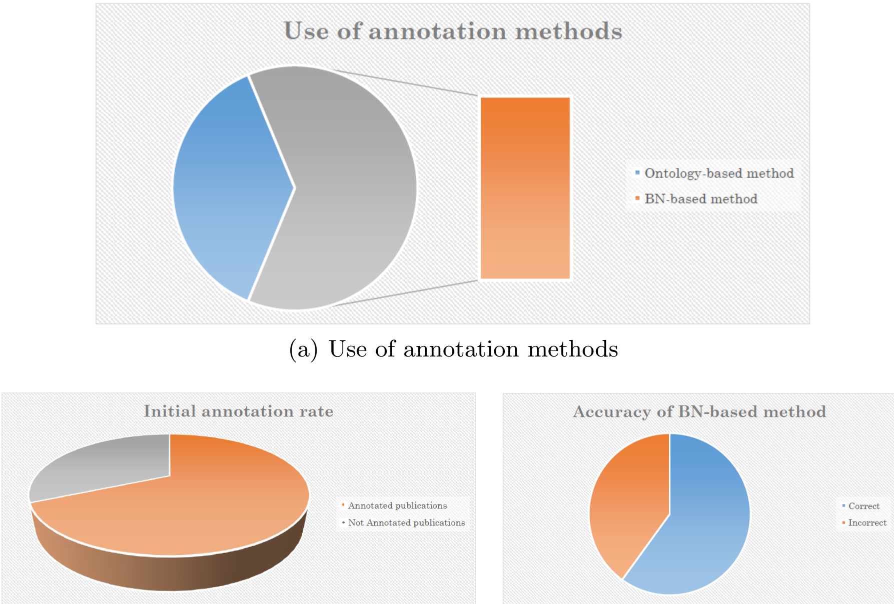
Now, the natural question that one could make is about the accuracy of the classification. For the choose samples, the accuracy of each publication has been verified, in the Table 3 are presented the obtained results.

Table 3. Accuracy method

Accuracy of BN-based method	**Percentage**
Correct	60%
Incorrect	40%

We can see that the strategy works well, but it is necesary improve the accuracy percentage of the method. This result has been obtained because the

BN used to classify the publications was built with no much examples that could give a better BN structure, i.e., learning Bayesian Networks structure is a NP problem. According with [2], the most examples used to build the BN were part of the *Ciencias Sociales* class. This is the reason of our results, the publications classified incorrectly, were associated to *Ciencias Sociales* class.



(a) Use of annotation methods

(b) Innitial annotation rate

(c) Accuracy of BN-based method

Fig. 5. Results summary

6 Conclusions and Future Work

The strategy applied in this work is based, essentially in the statistics that give evidence to automatize partially the annotation process in a semantic social network, but despite the BN-based method is a very powerful inference mechanism, it depends strongly of enough information to build a correctly BN that can classify the widest possible publications. In our case, we have seen that the BN used to classify need to be modified *continuously* according with new evidence (new publications). This structure modification will be the main topic research for our future work.

The future work consist in creating a software agent that can rebuild *continuously* the BN structure. We know the methodology to build a BN, now, we need to *automatize* this process and make it available to be used by the other components of the social network platform. In this way, it could be possible improve the BN-based method accuracy.

The rebuild process depends of the accuracy of the classification method, so, it is also necessary develop a mechanism capable of validate the classification by answering directly to social network's user about the accuracy of the classification. With this information could be rebuild the BN when the incorrect classification percentage achieve certain value.

References

1. Prié, Y., Garlatti, S. In: Annotations et métadonnées dans le web sémantique. (2004) Revue I3 Information-Interaction - Intelligence, Numéro Hors-série Web sémantique, 2004, 24 pp.
2. Conde R., J.C., Camarillo R., P., Sánchez L., A.: Clasificación de publicaciones en redes sociales semánticas mediante aprendizaje articial con redes bayesianas. In: Journal Research in Computing Science. (2013) 129–138
3. Berners-Lee, T., Hendler, J., Lassila, O., et al.: The semantic web. Scientific american **284** (2001) 28–37
4. Mika, P.: Social Networks and the Semantic Web. Semantic Web and Beyond, Computing for Human Experience. Springer Science+Business Media, LLC (2007)
5. Gruber, T.R.: A translation approach to portable ontology specifications. Knowl. Acquis. **5** (1993) 199–220
6. Downes, S.: Semantic networks and social networks. The Learning Organization Journal **12** (2005) 411–417
7. Jung, J.J., Euzenat, J.: Towards semantic social networks. In Franconi, E., Kifer, M., May, W., eds.: ESWC. Volume 4519 of Lecture Notes in Computer Science. Springer (2007) 267–280
8. Camarillo R., P., Sánchez L., A., Nuñez R., D.: Towards a semantic social network. In: IEEE CONIELECOMP 2013. (2013) 74–77
9. Borst, W., Akkermans, J., Top, J.: Engineering ontologies. International Journal of Human-Computer Studies (1997) 365–406
10. Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing. Int. J. Hum.-Comput. Stud. **43** (1995) 907–928
11. Camarillo R., P., Sánchez L., A., Nuñez R., D.: Moveek: A semantic social network. In: WILE 2012 (Fifth Workshop on Intelligent Learning Environments). (2012)
12. Ben-Gal, I.E.: Bayesian networks. WWW page (2007) `http://www.eng.tau.ac.il/~bengal/BN.pdf`.
13. Neapolitan, R.: Learning Bayesian networks. Prentice Hall Series in Artificial Intelligence. Pearson Prentice Hall (2004)
14. Cooper, G.F., Herskovits, E.: A bayesian method for the induction of probabilistic networks from data. Mach. Learn. **9** (1992) 309–347

A Framework for Unsupervised Dependency Parsing using a Soft-EM Algorithm and Bilexical Grammars

Martín Ariel Domínguez[1], Gabriel Infante-Lopez[1,2]

[1]Grupo de Procesamiento de Lenguaje Natural,
Universidad Nacional de Córdoba, Argentina
[2]Consejo Nacional de Investigaciones Científicas y Técnicas, Argentina
{mdoming, gabriel}@famaf.unc.edu.ar

Abstract. Unsupervised dependency parsing is acquiring great relevance in the area of Natural Language Processing due to the increasing number of utterances that become available on the Internet. Most current works are based on Dependency Model with Valence (DMV) [12] or Extended Valence Grammars (EVGs) [11], in both cases the dependencies between words are modeled by using a fixed structure of automata. We present a framework for unsupervised induction of dependency structures based on CYK parsing that uses a simple rewriting techniques of the training material. Our model is implemented by means of a k-best CYK parser, an inductor for Probabilistic Bilexical Grammars (PBGs) [8] and a simple technique that rewrites the treebank from k trees with their probabilities. An important contribution of our work is that the framework accepts any existing algorithm for automata induction making the automata structure fully modifiable. Our experiments showed that, it is the training size that influences parameterization in a predictable manner. Such flexibility produced good performance results in 8 different languages, in some cases comparable to the state-of-the-art ones.

Keywords: Unsupervised dependency parsing, bilexical grammars, soft-EM algorithm.

1 Introduction

In the last decade, unsupervised dependency parsing has acquired increasing relevance [3,4,9,11,13,17,18]. The special interest in unsupervised methods comes hand in hand with the growing number of natural languages available in different applications on the Internet. Unlike supervised and semi-supervised methods, unsupervised dependency parsing does not require training from hand-annotated corpora which is usually an expensive process. Therefore, unsupervised parsing becomes a solution for languages and domains with minimal hand-annotated resources, making it a low cost and high performance method of approaching the new challenges of natural language parsing.

Unsupervised dependency parsing looks for regularities in the languages by applying statistical methods to large quantities of data. The resulting linguistic patterns serve

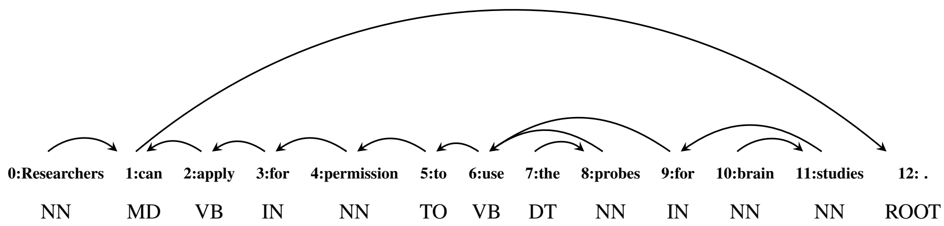


Fig. 1. An example of a dependency tree

as rules for inferring an underlying grammar. For example, the "dependency" pattern considers that the "dependent" of a preposition is usually a noun occurring to its right. This rule can explain -that is, parse- a number of different sentences that other candidate rules cannot. Then, a better method should prefer such a rule over competing alternatives and "discover" that grammatical rule. Usually, dependency relations are modeled as shown in Figure 1, where each arc is a relation between a head and its argument.

Currently, unsupervised dependency parsers exhibit a degree of complexity that can shy away newcomers to the field. We challenge such complexity and present a straightforward soft-EM based framework. We achieve results close to state-of-the-art ones, while making it simple to experiment with sub-components (see below).

Since the task is unsupervised, correct dependency structures are not available and our input consists only of sequences of parts of speech (POS) tags. Our dependency relations are modeled with Probabilistic Bilexical Grammars (PBGs) [8] for which we have implemented a novel learning/training algorithm. Our algorithm is a soft version of the EM-algorithm [5]. As shown in [16] an EM algorithm for inducing grammars can be described as an iteration between an E-step and an M-step. During the E-step a new treebank is computed, while during M-step a grammar together with its probabilities is read out from the treebank. Usual implementations of the EM do not actually compute the treebank; they compute the new grammar using inside-outside probabilities from the previous step.

We take a different approach. We present an algorithm based on the 4 different modules showed in Figure 2, that mainly computes new versions of a treebank. These 4 components are: a supervised PBGS INDUCTOR, (simulating the M-step), a k-BEST PBG PARSER, plus a TREEBANK REWRITER (together simulating the E-step), and an initial TREEBANK GENERATOR (in charge of building the initial seed). In the first step of the algorithm, the grammar is learned from an initial set of trees. Those trees are built based on constraints aimed to start the learning process from simple models. In each iterative step of the algorithm, it parses the set of sentences with the PBG and refines the grammar by contrasting the parsed trees of the input sentences. The quality of the grammar of each step is calculated by the logarithmic likelihood of the treebank obtained using that grammar to parse the set of input sentences.

The resulting soft-EM[1] algorithm is well defined for different PBG learning algorithms and for different initial treebanks. Consequently, these two components can be instantiated differently at almost no effort.

Thanks to the versatility offered by our framework, we are able to test three different ways to generate initial treebanks, and two different schemas for learning automata. Most of the recent work in this area, e.g., [4, 11, 18], has focused on variants of the Dependency Model with Valence (DMV) [12]. DMV was the first unsupervised dependency grammar induction algorithm to achieve accuracy above a right-branching baseline. With all its strengths, DMV is still limited in the type of dependencies it can model. The DMV model can be seen as a sort of PBG with the particularity that all of its automata have similar structures and that they only differ in the probabilities of their arcs. In contrast with our model, DMV and others in the literature are still in need of a well understood learning mechanism. By using a generalization of EM we can tap into a large body of learning expertise.

Our results show a very good performance in 5 languages. Particularly, for English these are very close to the state-of-the-art performance for sentences with a restricted length of up to 10 POS. For languages with enough available training material (German, Portuguese and Danish), we have state-of-the-art results or close to them such as for Swedish. For the rest of languages Turkish, Spanish and Bulgarian, our performance is considerably higher than the standard DMV performance.

The paper is organized as follows: Sections 2 and 3 present our framework and the algorithms for learning automata. Section 4 shows experimental results, Section 5 discusses related work, Section 6 explains possible research lines to continue this work and, finally, Section 7 concludes the paper.

2 Training Architecture

The training or learning framework (Figure 2) consists of 4 modules: the TREEBANK GENERATOR, the PBGS INDUCTOR, a k-BEST PBG PARSER, and a TREEBANK REWRITER. The learning algorithm starts by creating a treebank over a given set of sentences. The resulting treebank is used by the PBGS INDUCTOR module to induce a PBG. Once a grammar has been induced, it is used by the k-BEST PBG PARSER to parse all original sentences. The k-BEST PBG PARSER returns the k-best trees for each sentence with their corresponding probabilities. All these trees are used by the TREEBANK GENERATOR to create a new treebank that reflects the probabilities of all trees. Once the new treebank has been created, the algorithm cycles between the PBGS INDUCTOR, k-BEST PBG PARSER and TREEBANK REWRITER until the likelihood of the k-best trees hopefully converges. We will now describe each component.

[1] It is soft-EM because of the parameter k in the parser. If $k = \infty$ it will be a hard-EM.

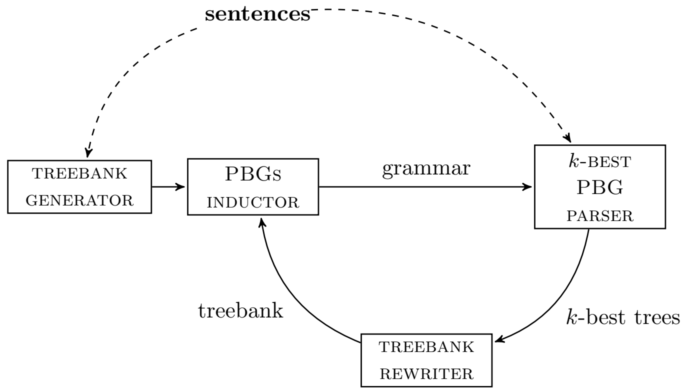


Fig. 2. Framework Schema

PBGS INDUCTOR. This module is one of the key components of our algorithm. Its task is to find a probabilistic bilexical grammar from a collection of dependency trees. From [8], recall that a *Bilexical Grammar* B is a 3-tuple $(W, \{r_w\}_{w \in W}, \{l_w\}_{w \in W},)$ where, W is a set of terminals, plus a distinguished symbol ROOT, and l_w, r_w with $w \in W$ are probabilistic automata with initial symbols S_l^w and S_r^w respectively. For this paper, it is enough to grasp the intuition behind them: the two automata for a word w accept a sub-language of W^* which models the arguments of w to its right and to its left. A Bilexical Grammar makes the strong assumption that the languages defined by the arguments of a word can be modeled with regular languages.

Learning a dependency grammar from a dependency treebank is simple if a learning algorithm for the induction of its automata is given. To induce a PBG from a dependency corpus, first we need to build the bags of strings that are to be used to learn the automata. In this sense, two bags of dependencies are built for each terminal in the set of terminals. These bags are given to the automata learning algorithm and it produces the two automata for that particular terminal. The bags of words are extracted from all trees in the dependency treebank. For example, using the tree in Figure 1 the corresponding left and right bags for POS `VB` are $M_{left}^{VB} = \{$`"VB #"`,`"VB #"`$\}$ and $M_{right}^{VB} = \{$`"VB IN #"`,`"VB NN IN #"`$\}$ respectively[2]. The symbol `#` marks the end of a string.

Once the process of collecting dependents has finished, there are two bags M_{left}^{w} and M_{right}^{w} for each POS w. These bags are used as training material to inducing automata l_w and r_w. We can now define the PBG $B = ($POS$, \{r_w\}_{w \in POS}$,$\{l_w\}_{w \in POS})$. In Section 3, we describe some algorithms capable of induce the automata l_w and r_w.

[2] Note that, for each POS, only the incoming arrows are considered as left or right dependents.

k-BEST PBG PARSER. Since PBGs are a particular case of probabilistic context free grammars, our parser for PBG is based on an implementation of a k-best CYK parser for Chomsky Normal Form PCFGs. The k-BEST PBG PARSER returns the k-best trees together with their probabilities.

TREEBANK REWRITER. Intuitively, this module uses the k-best trees for creating a new treebank that resembles the known probabilities of individual trees. Although we know the probabilities of the sentences, we need to replicate it because it allows us to use any automata inductor, for example MDI, which accepts only a set of sentences as a training material. Since the grammar inductor only takes a treebank as input, it is not aware of their probabilities. The TREEBANK REWRITER module replicates the k-best trees in such a way that the probability mass associated to each tree is proportional to the probability assigned by the parser. The TREEBANK REWRITER produces a new treebank that contains as many trees for each sentence as are required to reproduce the sentence probability. In order to mark the boundaries of the number of possible replicas, we introduce a constant M that states the maximum number of trees a sentence will have in the resulting treebank. Suppose that $C = \{c^1 \dots c^N\}$ of N sentences are the input senteces. Suppose also that $t_1^j \dots t_k^j$ are the k trees returned by the parser for the sentence c^j and let $p_1^j \dots p_k^j$ be their probabilities. t_i^j is replicated R_i^j times, where:$R_i^j =$ round $\left(M * \frac{p_i^j}{\sum_{l=1}^{k} p_l^j}\right)$. Finally, the size of the resulting treebank is $\sum_{j=1}^{N} \sum_{i=1}^{k} R_i^j$. Note that, under this definition, if the probability of a tree is too small, it will not be a part of the new treebank at all. For computational reasons both k and M cannot be too large. In all our experiments, k and M are set to 130 and 150 respectively.

TREEBANK GENERATOR. The aim of the TREEBANK GENERATOR is to build the first treebank that is given to the grammar inductor. This module uses *meta-trees*. A *meta-tree* of size n is a tree with a particular fixed structure of arcs, similar to a dependency tree, but with n variable nodes, where any sentence of length n can be fitted. These meta-trees have the particularity that their leaves are all different and that they are grouped by sentence length. Given an initial sentence of length n, a small treebank for this sentence is built via a two step procedure. First, all meta-trees whose yield has length n are selected and, second, all terminal symbols in the meta-trees are replaced by those in the sentence. The TREEBANK GENERATOR produces a new treebank by joining individual treebanks for all sentences. As a consequence, the resulting treebank contains the same trees for all sentences with the same length independently of the sentence words.

To generate the meta-trees that correspond to all n-long sentences, a special sentence $w_0, \dots, w_n$, with all w_i different symbols, is processed by the parser and the tree rewriter modules. The sentence is parsed with an *ad-hoc* PBG that we built specifically for each possible length. The *ad-hoc* PBG is defined by describing the automata for each word in the sentence. If w_i is the i-th terminal then, its right automaton is like the one shown in Figure 3. That is, the automaton has three states, one is final and absorbing, one is initial and has only one outgoing transition that is labeled with label w_i and

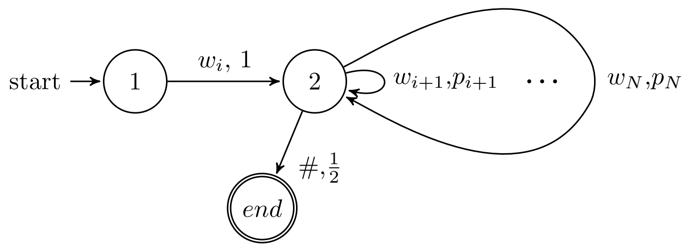


Fig. 3. Typical meta-tree building automaton

probability 1. The third state is in between the two previous ones. It is connected to the final state by means of an arc labeled with probability 0.5 and label $\#$. Moreover, it has $n-i$ loops with labels $w_{i+1}\dots w_n$ and probabilities p_j defined as: $p_j = \frac{\frac{1}{d_{ij}^s}}{2*\sum_{d=1}^{n-1}\frac{1}{d^s}}$, where d_{ij} is the distance between w_i and w_j, and the exponent s is a parameter in our model that modifies the mass of probability that are assigned to long dependencies. The three initialization, namely init-1, init-2 and init-3, we report in Section 4 are obtained by setting s to 1, 2 and 3 respectively. All p_i are such that their sum is not equal to 1, and in order to correctly define a probabilistic automaton, a forth non-final and absorbing state is required to normalize the probability. The probability of going to this state is $1-(0.5+0.5*\sum_{l=i+1}^{m}p_l)$. This state is not shown in the picture. Intuitively, the probability of having many dependents and of having long distance dependents diminishes with an exponential factor s. The bigger the s the less likely are these two situations. The $2*m$ automata, 2 per terminal symbol in the sentence, plus one automaton for the *root* symbol, are used to define a PBG G. Following the general schema, G is used to produce the k-best parsers of sentence $w_0,\dots,w_m$. These k trees are fed into the TREEBANK GENERATOR, and it produces the treebank of meta-trees.

3 Automata Learning Algorithms

We use two different algorithms for learning probabilistic automata. First, we propose the Minimum Discrimination Information (MDI) algorithm [24], which infers automata in a fully unsupervised fashion. It takes a bag of strings and returns a probabilistic deterministic automaton. Briefly, the MDI algorithm produces an automaton by first building a prefix tree that accepts only the training material. Moreover, the prefix tree contains one path for each string and it contains as many final states as there are different strings. All arcs in the prefix tree are marked with the number of times each arc is traversed while mapping strings into paths. These numbers are then transformed into probabilities which results in a probabilistic automaton that recognize exactly the training material. The algorithm proceeds to look for pairs of states that can be merged into one single state. Two states can be merged if the probability distribution defined by the automata that results from the merge is not too far away[3] from the distribution defined by the prefix tree. The algorithm proceeds greedily until no further merges can be done. MDI has only one parameter, α, that can be used to control the maximum allowed value of distance between two distributions before a merge is performed. α is a real number between 0 and 1; when it is equal to 0, no merges are allowed while when equal to 1 all merges are allowed. The MDI algorithm receives a bag of strings together with a value for the parameter α, and it outputs a probabilistic automaton.

Second, we contribute an *ad hoc* algorithm that only learns the transitions probabilities of a given automaton backbone structure. A backbone structure is a deterministic finite automaton without probabilities. It receives a bag of strings and a automaton backbone structure and returns the automaton backbone structure plus their probabilities. The backbones we use are general enough to warranty that they accept all strings in the training material. In contrast, our second algorithm is not fully unsupervised because it receives along with a bag of strings, the backbone of the automaton it should produce. The backbone consists of the states and the arcs of the automaton, but it is missing the transition probabilities. It is the task of our Given Structure (GS) algorithm to find them.

As we see in Section 5, DMV and EVG define a particular skeleton to their automata and as is the GS, the skeleton is information that is given to the algorithm as prior knowledge. In this sense, MDI is fairer learner than GS given that it works with less information. In our experiments we show that even with less information, MDI works better than GS. We currently experiment with different skeletons, but all of them have similar structure: They have a unique absorbing final state, and N intermediate states $S_1 \dots S_N$. The skeleton has one arc between states S_i and S_{i+1} for each possible label, and one arc between states S_i and the final state, labeled with the end of production symbol $\#$. The GS algorithm uses the training material to estimate all arcs probabilities. Since the skeleton is both deterministic and expressive enough, there is a path in the automaton for each sequence in the training material. The GS algorithm maps each sentence to a path in the automaton, it records the number of times each arc has been used, and, finally, it transforms those counters into probabilities. GS-N refers to the GS algorithm when it uses a backbone with N states as describe above. We use these skeletons because they are the easiest structure that can be manually described without making strong assumptions about the underlying language. We experiment with two different skeletons both having 3 and 4 states respectively. The skeleton with 3 states pays special attention to the first dependent while the one with 4 to the first two dependents. Moreover, GS-3 automata generate dependents independently of their siblings. GS-4 automata can recall if a dependent is the first one or not. In general the GS-i can recall if there has been less that $i-1$ dependents. Our GS algorithm is also a generalization over n-grams which can be seen as instances of GS-n where the skeleton has a particular shape. Moreover, Section 5 shows that they can be seen as instances of a

[3] The difference between the two probability distributions is computed by means of the Kullback-Leibler divergence

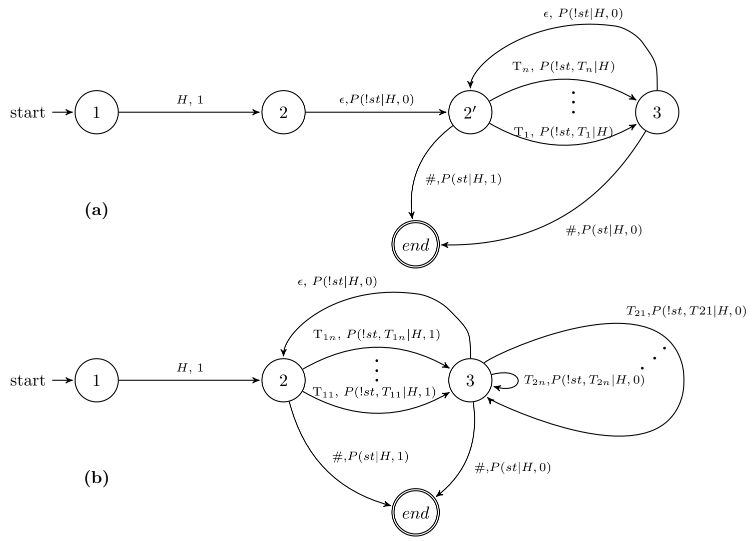


Fig. 4. DMV (a) and EVG (b) automata

Table 1. Size of the training corpus for each language and the number of differents POS tags

	English	German	Turkish	Swedish	Bulgarian	Spanish	Portuguese	Danish
#sen.	6007	12089	3203	3255	5713	595	2409	1757
#POS	36	51	28	40	40	23	19	24

version of GS that allows non-deterministic backbones.

4 Experimental Results

Our model was tested on sentences with a restricted length up to 10 POS. Due to computational costs, the current version of our parser cannot deal with sentences of any length. However, we have some experiments which use sentences with up to 20 POS with promising results.

We report results for English, German, Swedish, Turkish, Bulgarian, Spanish, Portuguese and Danish. We tested 3 different initializations in the TREEBANK GENERATOR module, and two different algorithms for learning automata in the PBGS INDUCTOR module. This showcases the flexibility of our framework. All of our experiments used syntactic categories —POS tags— instead of words. The English model was induced using sentences in the Penn treebank (PTB) [14] with at most ten words (usually called WSJ10 [11, 12, 17]). Sections 2 to 21, that is, 6,007 sentences in total, were used for training. Testing was done using the 398 sentences of Section 23. The metrics used for

Table 2. Directed accuracies on Section 23 of WSJ10 for several baselines and recent systems

model	accuracy
Attach-Right [13]	33.4
DMV-standard [13]	45.8
DMV-babysteps (@15) [18]	55.5
DMV-babysteps (@45) [18]	55.1
DMV-diriclet [3]	45.9
Best of [15]	53.5
Log-Normal Families [3]	59.4
Shared Log-Normals (tie-verb-noun) [4]	61.3
Bilingual Log-Normals (tie-verb-noun) [4]	62.0
EVG-Smoothed (skip-head) [11]	65.0
EVG-Smoothed (skip-val) [11]	62.1
Viterbi EM [23]	**65.3**
EVG-Smoothed (skip-head), Lexicalized [11]	68.8
Hypertext Markup [20]	**69.3**
LexTSG-DMV ($P_{lcfg}, P_{cfg}, P_{sh}$) [1]	67.7

model	accuracy
MDI, $\alpha = 0$, init-1	67.5
MDI, $\alpha = 0$, init-2	**69.1**
MDI, $\alpha = 0$, init-3	67.2
GS-3, init-1	50.9
GS-3, init-2	66.6
GS-3, init-3	67.1
GS-4, init-1	55.7
GS-4, init-2	66.7
GS-4, init-3	**67.6**

evaluation are the usual ones, directed and undirected accuracy[4]. Table 2 compares our English results with others in the literature. In this case we report only directed accuracy. From the table, it can be seen that our results are comparable with the state-of-the-art ones, even for lexicalized instances. Our best result is what we call MDI-0: MDI with $\alpha = 0$ using init-2.

Our best performing models are those that can model the language of dependents as finite languages. Moreover, an inspection on the resulting automata shows that all models tend to create very short sequence of dependents, mostly of length up to one. To better understand our results, it is important to think our learning algorithm as a two-step process. First, the parser and the rewriter define the training material that is going to be given to the automata learning algorithms. In a second phase, all the automata are learned. It is interesting to note that both the MDI and the GS-n algorithms generalize less over the training material as their respective parameters α and n go to zero and ∞, respectively. The MDI algorithm ends up building a tree like automata that recall all and only those strings in the training material. The probability assigned to each string is proportional to the number of times it occurs in the training material. In contrast, a GS-n automaton recalls the number of times each tag occurs as the i-th dependent. In this sense, MDI-0 generalizes less than any GS-n. In both cases, the resulting automaton accepts only finite languages.

From the experiments, it is clear that GS-4 improves over GS-3, and both EVG and DMV. It is surprising that our models obtain good results without resorting to smoothing which is usually applied in other models. As the experiments show, good results are obtained with initializations that penalize long distance dependencies and a high number of dependents. In other words, the models that work better are those that use initialization where words have fewer dependents and where dependents are close to

[4] Directed accuracy is the ratio of correctly predicted dependencies (including direction) over total amount of predicted dependencies. Undirected accuracy is much the same, but also considers a predicted dependency correct if the direction of the dependency is reversed [12].

their heads. If we compare the automata that results at the end of our algorithm, when init-2 and init-3 is used, the most noticeable feature is that, even for GS-3 and GS-4, the probabilities associated to cycling arcs are zero or very close to zero. When init-1 is used, cycles occur in the final automata of GS-3 but only a few in GS-4. Note that MDI-0 is stable across different initializations. If we look at the resulting automata, they all accept finite languages and moreover, all elements in the languages contain only a few symbols per string.

Since there are many parameters in our setup, it might be the case that our models are optimized for English. To validate our model, and unsupervised models in general, it is important to test their performance also in languages other than English. Table 3 compares our results for other languages. We compare them against standard baselines like right and left attachment, DMV model results, and the best results reported in [9]. According to [9], German and Turkish best results are obtained by one model while the score for English and Swedish by two other different models. The fourth row displays the highest score independently of the model used to obtain it. All corpora were part of the ConNLL-X special task on parsing [2]. We show results using treebanks for Swedish, German, Turkish, Bulgarian, Spanish, Portuguese and Danish. Trees that were non-projective or that had more than one root were discarded as well as all trees whose sentences were longer that 10 words. Except Turkish, where the best performing model is the right attach baseline, all instances of our algorithm improve over DMV and the baselines.

Figure 5 shows the evolution of the directed accuracy and logarithmic likelihood. Figure 5 (left) shows, for each language, the directed accuracy in the 30 first iterations measured against the gold trees from the training material. More specifically, while the X axe varies in the number of iteration, the Y axe plots, the directed accuracy of the trees that consists only of the most probably trees returned by the k-BEST PBG PARSER for each sentence in the training material [5]. For iteration number 0 we use the treebank returned by TREEBANK GENERATOR instead of k-BEST PBG PARSER.

Similarly, in Figure 5 (right) we plot the logarithmic likelihood for each treebank in the first 30 iterations. It is important to remark that the gold trees are used only for analysis purposes, and they are not used inside the algorithm.

Two variables must be taken into account to decide which parameterization of our system should be used for a given language: the number of sentences available for training and the number of POS tags.[6] Table 1 shows the variables for the languages used in this work. Table 3 shows that MDI-0 is very robust for those languages that have available a corpus with a significant number of sentences. This is the case of languages such as English, German and Bulgarian. For languages with a reduced number of sentences or a big number of POS, we should use GS-3 or GS-4. Intuitively, if we have

[5] This treebank is the same that produce as a result the 1-BEST PBG PARSER

[6] More tags means a larger number of automata to be induced and thus less training material for each automaton.

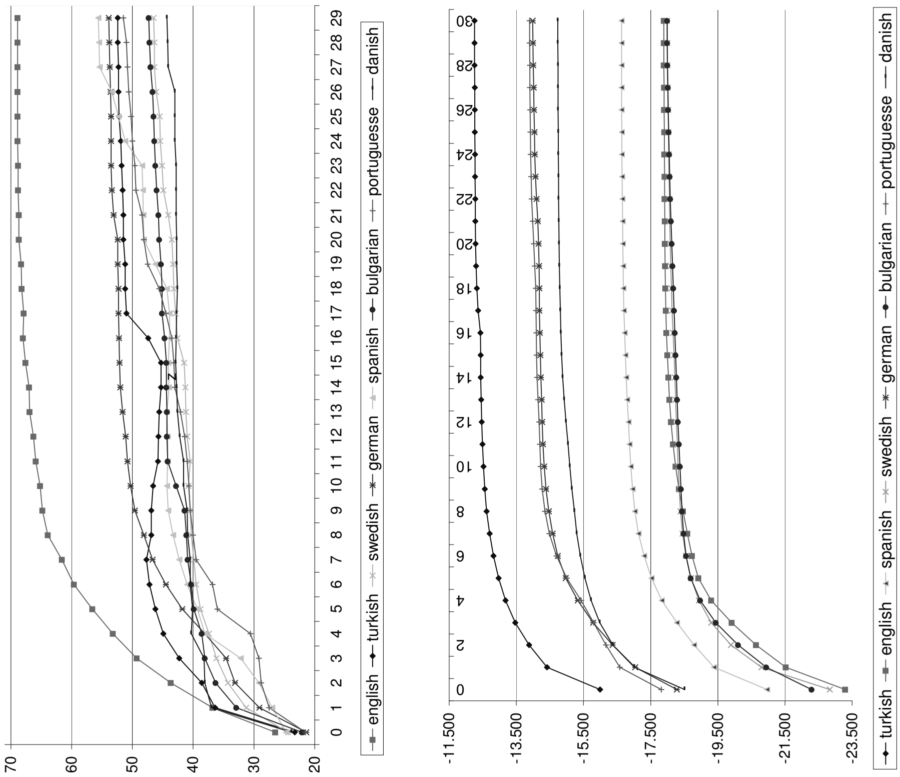


Fig. 5. (left):Directed Accuracy evaluated for each language over the first 30 iterations of the training phase of our framework (right):Evolution of logarithmic likelihood for each language

less training material, models like GS-3 or GS-4 perform better because they generalize over the training material better than MDI-0.

Table 3. Our results for a variety of languages compared with the baselines: right attach, left attach and standard DMV results. We also report the state-of-the-art results for these languages.

model	English	German	Turkish	Swedish	Bulgarian	Spanish	Portuguese	danish
Gillenwater et al. [9] DMV	45.8 / -	35.7 / -	46.8 / -	39.4 / -	37.8 / -	40.3 / -	35.7 / -	47.2 / -
left attach	24.1 / 54.6	25.9 / 53.2	5.1 / 54.8	28.5 / 56.3	40.5 / 59.9	29.0 / 55.2	34.1 / 61.7	43.7 / 60.1
right attach	33.4 / 56.3	29.0 / 52.1	**63.8 / 68.5**	28.5 / 55.5	20.2 / 56.3	29.4 / 55.2	27.9 / 55.5	17.2 / 57.5
Gillenwater et al. [9] best result	64.4 / -	47.4 / -	56.9 / -	**48.6 / -**	**59.8 / -**	**62.4 / -**	54.3 / -	46.6 / -
GS-3 Init-1	50.9 / 65.3	48.6 / 60.7	52.8 / 65.3	46.0 / 60.2	48.7 / 64.1	57.4 / 68.6	55.6 / 66.4	36.8 / 59.7
GS-3 Init-2	66.6 / 72.4	49.1 / 60.7	20.4 / 54.4	47.5 / 61.5	48.6 / 63.9	45.6 / 63.7	39.7 / 63.3	**47.9 / 66.4**
GS-3 Init-3	67.1 / 71.5	46.7 / 59.6	20.4 / 53.6	41.6 / 58.6	34.1 / 55.0	38.3 / 59.0	38.0 / 62.1	44.3 / 62.9
GS-4 Init-1	55.7 / 66.9	48.5 / 60.8	53.5 / 65.2	46.7 / 60.2	34.6 / 55.8	55.3 / 66.9	**55.7 / 66.8**	38.9 / 59.9
GS-4 Init-2	66.7 / 72.4	48.7 / 60.4	43.3 / 60.2	47.6 / 61.5	47.7 / 63.0	45.1 / 63.5	39.5 / 63.2	41.6 / 60.6
GS-4 Init-3	67.6 / 71.9	47.9 / 60.1	25.6 / 53.9	42.3 / 59.2	48.6 / 64.1	38.2 / 58.9	38.1 / 61.9	43.1 / 63.9
MDI-0 Init-1	67.5 / 72.5	47.7 / 60.1	52.5 / 64.9	45.4 / 59.5	35.9 / 55.6	51.1 / 62.6	49.5 / 63.6	35.5 / 58.1
MDI-0 Init-2	**69.1 / 73.3**	**54.1 / 63.4**	38.5 / 58.2	**48.1 / 61.4**	55.1 / 68.9	48.8 / 64.7	30.6 / 55.5	44.1 / 64.6
MDI-0 Init-3	67.2 / 72.6	53.9 / 63.5	24.6 / 53.0	46.2 / 60.7	38.1 / 56.5	46.0 / 64.0	30.8 / 55.8	44.7 / 65.0

5 Related Work

Most unsupervised approaches to unsupervised parsing are based on Dependency Model with Valence (DMV). DMV implements an EM algorithm that maximizes the likelihood of a particular grammar. This grammar can be seen as PBG where all its automata are like the one in Figure 4 (a). The probability between states 1 and 2 is the probability of generating a particular head. The one between states 2 and $2'$ is the probability of generating any dependent using a ϵ movement; the one between states 2 and end is the probability of not generating any dependent; the one between $2'$ and 3 is the probability to generate a particular dependent with its corresponding probability, the one between 3 and 2 is the probability of generating a new dependent, modeled again with an ϵ move, and finally, the probability between 3 and end is the probability of stop generating.

In general, is not possible to transform a non-deterministic automaton to a deterministic one [7]. But for this particular case, the automata can be transformed to one without ϵ moves, but having in mind that some of the its arc probabilities are correlated and consequently can not be learned independently. Cohen et al. [3] derive a Variational Bayes EM for the DMV model. results were 59.3 of directed accuracy.

Spitkovsky et. al. [18] use the DMV model, but they introduce two interesting techniques. First, they use an incremental initialization that starts with sentences of length 1, and only later uses longer sentences. Second, they analyze the trade-off between complexity and quality in the training phase. They found that training with sentences up to length 15 performs better than training with longer sentences when testing in section 23 of WSJ10, WSJ20, WSJ30, WSJ100 and WSJ∞, among others.

Headden et al. [11] extend DMV by adding a parameter that distinguishes the probabilities of the first dependent from the probabilities of the subsequent ones. As in the DMV, even with the automata not being explicitly defined, they can be rebuilt from the definition of the model. Figure 4 (b) shows such an automaton. The probability between states 1 and 2 is the probability of generating a particular head, between 2 and 3, are the probabilities of generating a particular dependent as the first one, between 3 and 3 are the probabilities of generating a dependent that is not the first one anymore, the probability between 2 and end is the probability of not having any dependents, and finally the probability between 3 and end is the probability of stopping generating dependents. To maximize the likelihood of their model they use a Variational Bayes EM algorithm with a Dirichlet prior (similar to [3]) and they used a linearly smoothed model to deal with the data sparseness. As their initialization, they randomly sample some sets of trees and choose the best ones using some iterations of a Variational Bayes EM algorithm. They show that, by including smoothing, they improve over DMV obtaining the best result that is known for unsupervised parsing. The unsmoothed version of the EVG model corresponds exactly to our GS-3 model. The DMV model without the one-side-first parameter, is in between GS-2 and GS-3. It does not distinguish the probabilities for the dependent generation, as in GS-2, but the probability of stopping is distinguished like in GS-3. Gillenwater et al. [9] used a posterior regularization (PR) framework [10] instead of the traditional EM algorithm. They model their dependencies as in DMV, and as variants of the EVG model. They argue that the main problem with unsupervised parsing is data sparseness and their model deals with such problem adding constraint that control for long dependencies. They report substantial gains over the standard EM algorithm, but they are not stable across languages. Finally, our soft-EM corresponds to a hard EM algorithm [16] when a k-BEST PBG PARSER with $k = \infty$ is used. Hard EM is not feasible in our setup because a ∞-best parsing requires an exponential amount of space and time.

Spitkovsky et. al. [23] is in one sense, the work most similar to ours, as we are also estimating the probabilities of a model given the previous model, albeit using k-best parse trees. They obtain good scores 44.8% for English, in long sentences (all sentences in section 23 of PTB) .

Blunsom and Cohn [1] replace the simple underlying grammar commonly used by a probabilistic tree substitution grammar. This formalism is capable of better representing complex linguistic structures because they can learn long dependencies. To limit the model's complexity they used a Bayesian non-parametric prior. They obtained state-of-the-art results for English in long sentences 55.7%.

Using standard DMV, Spitkovsky et al. [20] use Web mark-up for improving parsing up to 50.4% of directed accuracy.

6 Future Work

One of the most important aspects to continue our work is to extend our experiments to longer sentences. To do so, we should optimize our implementation of the parser to make it parallel. We performed some experiments with wsj15 and we obtained promising results, about 49% of directed accuracy, which is close to the state-of-the-art ones, about 53%.

Another experiment that we will perform is to use ideas explored in [6] to split the POS tags in a refined set. Fully lexicalized models may be costly and too sensitive to data sparseness, nevertheless we think unsupervised parsers can benefit of a more appropiate granularity of POS tag sets. This idea may be implemented by selecting a POS tag and splitting the words with this POS by using a chosen feature function. For example, by selecting the POS `VB`, and splitting it using the distance of the word to the root node. We think of applying the split of POS tags starting with the initial tree-bank, which is obtained as in section 2. This split should be recalculated in each step of our learning architecture, after the new set of dependency trees is calculated by using the k-best parser. We hope that this idea may help to obtain more accurate dependency trees, specially with longer sentences because it could distinguish more complex dependency relationships by using more automata specialized according to the dependency languages of each POS tag considered.

Finally, our model allows us to choose different kinds of automata structures. Another interesting idea to improve this parsing model is to benefit from this flexibility of our framework. An interesting experiment is to choose the automaton structure to be associated with a given POS tag according with the size of its training set. As the results obtained for different languages suggest, the automata structure[7] can be adapted to the size of the dependency tree-bank; our idea is to investigate potential relationships between the size of the training set associated with each POS tag.

7 Discussion and Conclusions

Over the last years NLP research has been focused in unsupervised dependency parsing, specially after the DMV parser [13]. Most of the recent parsers like [1, 3, 4, 9, 11, 15,

[7] Recall that we use the same automata structure for all POS tags.

21, 22] are essentially the DMV model which uses a different bias function in each step of its EM algorithm definition. This bias function penalizes long dependencies in utterances. This penalization is needed, as it is explained in [19], because DMV reserves too much probability mass for what might be unlikely productions at the beginning, and the classic EM is not good enough for redistributing such probability across all parse trees.

We chose to take a different approach. Our algorithm implements a soft-EM algorithm that instead of computing the probability distribution over the whole forest of trees, uses a tree-replicator module that builds tree-banks resembling the most likely part of the probability distribution. The resulting algorithm allows us to test different ways to model dependents and different ways to induce automata.

Instead of using a penalization in each iteration of the EM algorithm, in our model we use different biased tree-banks which perform the penalization of long dependencies. We show experimental results using three different initializations, two automata learning algorithms and eight different languages.

Our experiments showed that, for a given language, we have to choose a parameterization of our system that generalizes across different training sets depending on the size of training material available for this language. We show training size influences parameterization in a predictable manner.

References

1. Blunsom, P., Cohn, T.: Unsupervised induction of tree substitution grammars for dependency parsing. In: In Proceedings of EMNLP 2010 (2010)
2. Buchholz, S., Marsi, E.: Shared task on multilingual dependency parsing. In: CoNLL-X. SIGNLL (2006)
3. Cohen, S.B., Gimpel, K., Smith, N.A.: Logistic normal priors for unsupervised probabilistic grammar induction. In: Proceedings of Advances in Neural Information Processing Systems (NIPS) (2008)
4. Cohen, S.B., Smith., N.A.: Shared logistic normal distributions for soft parameter tying in unsupervised grammar induction. In: Proceedings of NAACL-HLT. (2009)
5. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B 39(1), 1–38 (1977)
6. Domínguez, M.A., Infante-Lopez, G.: Searching for part of speech tags that improve parsing models. In: Proceedings of the 6th international conference on Advances in Natural Language Processing, GoTAL,Gotemburgo, Suecia. (2008)
7. Dupont, P., Denis, F., Esposito, Y.: Links between probabilistic automata and hidden Markov models: probability distributions, learning models and induction algorithms. Pattern Recognition 38(9), 1349–1371 (September 2005)
8. Eisner, J.: Three new probabilistic models for dependency parsing: An exploration. In: Proceedings of COLING-96. Copenhagen ("1996")

9. Gillenwater, J., Ganchev, K., Graça, J.a., Pereira, F., Taskar, B.: Sparsity in dependency grammar induction. In: Proceedings of the ACL 2010 Conference Short Papers. pp. 194–199. Morristown, NJ, USA (2010)
10. Graça, K.G., Taskar, B.: Expectation maximization and posterior constraints. In: Proceedings of Advances in Neural Information Processing Systems (NIPS) (2007)
11. Headden, W.P., Johnson, M., McClosky, D.: Improving unsupervised dependency parsing with richer contexts and smoothing. In: Proceedings of NAACL-HLT. (2009)
12. Klein, D.: The Unsupervised Learning of Natural Language Structure. Ph.D. thesis, Stanford University (2005)
13. Klein, D., Manning, C.: Corpus-based induction of syntactic structure: Models of dependency and constituency. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics (2004)
14. Marcus, M., Santorini, B.: Building a large annotated corpus of English: The Penn treebank. Computational Linguistics 19, 313–330 (1993)
15. Pate, J.K., Goldwater, S.: Unsupervised syntactic chunking with acoustic cues: computational models for prosodic bootstrapping. In: Proceedings of CMCL '11. Association for Computational Linguistics, Stroudsburg, PA, USA (2011)
16. Prescher, D.: Inside-outside estimation meets dynamic EM. In: Proceedings of the 7th International Workshop on Parsing Technologies (IWPT) (2001)
17. Smith, N., Eisner, J.: Guiding unsupervised grammar induction using contrastive estimation. In: IJCAI, Workshop on Grammatical Inference Applications. pp. 73–82. Edinburgh (July 2005)
18. Spitkovsky, V., Alshawi, H., Jurafsky, D.: From baby steps to leapfrog: How less is more in unsupervised dependency parsing. In: Proceedings of NAACL-HLT. (2010)
19. Spitkovsky, V.I., Alshawi, H., Jurafsky, D.: Baby Steps: How "Less is More" in unsupervised dependency parsing. In: NIPS: Grammar Induction, Representation of Language and Language Learning (2009)
20. Spitkovsky, V.I., Alshawi, H., Jurafsky, D.: Profiting from mark-up: Hyper-text annotations for guided parsing. In: Proceedings of ACL-2010 (2010)
21. Spitkovsky, V.I., Alshawi, H., Jurafsky, D.: Punctuation: making a point in unsupervised dependency parsing. In: Proceedings of CoNLL '11. pp. 19–28. ACL (2011)
22. Spitkovsky, V.I., Alshawi, H., Jurafsky, D.: Bootstrapping dependency grammar inducers from incomplete sentence fragments via austere models. Journal of Machine Learning Research - Proceedings Track 21 (2012)
23. Spitkovsky, V.I., Alshawi, H., Jurafsky, D., Manning, C.D.: Viterbi training improves unsupervised dependency parsing. In: Proceedings of CoNLL '10g (2010)
24. Thollard, F., Dupont, P., de la Higuera, C.: Probabilistic DFA inference using Kullback-Leibler divergence and minimality. In: Proceedings of ICML (2000)

A Novel Reordering Model for Statistical Machine Translation

Saeed Farzi,[1] Heshaam Faili,[1] Shahram Khadivi,[2] Jalal Maleki[3]

[1] NLP research Lab,ECE Department University of Tehran, Tehran, Iran
[2] CE/IT Dept, Amirkabir University of Technology, Tehran, Iran
[3] Dept. of Computer and Information Science Linköping University SE-581 83 Linköping, Sweden

{saeedfarzi,hfaili}@ut.ac.ir, jalal.maleki@liu.se, khadivi@aut.ac.ir

Abstract. Word reordering is one of the fundamental problems of machine translation, and an important factor of its quality and efficiency. In this paper, we introduce a novel reordering model based on an innovative structure, named, phrasal dependency tree including syntactical and statistical information in context of a log-linear model. The phrasal dependency tree is a new modern syntactic structure based on dependency relations between contiguous non-syntactic phrases. In comparison with well-known and popular reordering models such as the distortion, lexicalized and hierarchical models, the experimental study demonstrates the superiority of our model regarding to the different evaluation measures. We evaluated the proposed model on a Persian→English SMT system. On average our model retrieved a significant impact on precision with comparable recall value respect to the lexicalized and distortion models, and is found to be effective for medium and long-distance reordering.

Keywords: Reordering, phrase-based SMT, syntactical reordering model, long distance reordering.

1 Introduction

The machine translation task is made of two sub-tasks: collecting the list of words in a translation, which is called the lexical choice, and determining the order of the translated words, which is called reordering [3]. In comparison with the word-based systems, the phrase-based systems can readily address local reorderings whilst the reordering is still a computationally expensive problem at the phrase level. The inability of handling the long-distance reordering problems is known as a pitfall of the Phrase-based SMT, which generally two well-known mechanisms have been introduced so far for it [13, 19]. (1) The distortion penalties, and (2) the lexicalized reordering models. The lexicalized reordering models demonstrate superiority regarding to the distortion models in term of handling the long-distance reorderings because of using phrase content information. The distortion penalty not only forces translation systems not to prefer long-distance reorderings, but also has not considered phrase content information. Thus, it is difficult to obtain a satisfactory translation performance. The lexicalized reordering models suffer from data sparseness problem as well as they are restricted to reorder adjacent phrases, phrases with no gap, whereas the long-distance reorderings especially for syntactically divergent language pairs require much more robust solution in order to predicate the orientations of non-adjacent phrases.

In present research, a new way of integrating the phrase-based and syntactically-informed models is proposed as the form of a new model that supplements the Phrase-based SMT [13, 19]. The crystal clear suggestion is to exploit the syntactically-informed reordering elements (reordering rules) based on novel dependency structure, named, the phrasal dependency tree solely for dealing with the medium- and long-distance reorderings. The phrasal dependency tree is a sort of modern syntactic structure based on dependency relations between contiguous non-syntactic phrases. In addition, rather than standard dependency trees in which words are vertices, our trees have phrases as vertices. In order to handle the short-distance reordering problem, we leverage the achievement of the phrase-based approaches providing a series of target words appropriately ordered as a phrase. In general, the lexicalized reordering model not only learns just the orientation of adjacent phrases but also suffers from the data sparseness problem whereas the proposed model tries to overcome these problems.

Two groups of evaluations have been performed on the proposed reordering model as follows. 1) We follow several sorts of scenarios with different goals to verify the performance and make the experimental work stronger. Two Persian→English translation tasks with different sizes have been employed to imply the accuracy and efficiency of our model. The performance of our model also has been compared with the distortion, lexicalized reordering and hierarchical-based models in term of BLEU [20], TER [24] and LRscore [2] measures. The results illustrate the superiority of our approach. 2) The ability of the proposed model to predicate the medium- and long-distance reorderings has been evaluated in more details. On average our model retrieved a significant impact on precision with comparable recall value respect to the lexicalized and distortion models.

The paper is organized as follows. Related works are reviewed in Section 2. In Section 3 and Section 4, the phrasal dependency tree and the phrase reordering model are explained in more details, respectively. In Section 5, the experimental studies are presented. Finally, we draw some conclusions in Section 6.

2 Related Works

In order to vanquish the long-distance reordering problem; some simple models have been introduced. First one is the distortion model [13, 19], which penalizes translations respect to their jumping distance. Second one is the flat reordering model [14, 29, 31], which is not content dependent either. Last one is the lexicalized reorderings model introduced by several researchers [11, 14, 19, 28]. It is a content dependent approach unlike two previous models. The local orientations of each bilingual phrase pair is learned by the lexicalized reordering model. Performance gains have been observed for systems using the lexicalized reordering model. A hierarchical orientation model, which deals with some global phrase reorderings by a shift reduce algorithm has been proposed by [7, 8]. Due to the heavy use of lexical elements, the last two models tend to suffer from data sparseness problems. Another restriction is that the lexicalized models are limited to reordering phrases with no gaps (adjacent phrases). In comparison to [7, 8], our model uses a systematic approach to fight with the data sparseness problems. Utilizing head words instead of phrases in the phrasal dependency relations reduces side effects of using lexical information. This method benefits from its simplicity, but it suffers from purveying at most a one best guess at syntactic movement. Search-space constraints restrict the decoding search space using syntactic intuitions [1].

There have been many attempts to employ dependency parse in SMT. Quirk et al. [21] integrated a source-side dependency parse with word alignments in order to model dependency relations on biphrases. In contrast to [21], our model employs target-side dependency parser. Shen et al. [23] and Gao et al. [9] introduced an extended version of Hiero [5] in which a dependency parse has been employed in order to inject dependency relations into the non-contiguous phrases. In contrast to the model of [9, 22, 23], our model works on non-syntactic contiguous phrases with left-to-right decoder whereas Shen et al. [23] have to design a string-to-tree machine translation system. Galley and Manning [7] relaxed standard assumptions about dependency parsing because the efficient left-to-right decoding algorithm of phrase based translation could be retained while a dependency parse is included.

3 Phrasal Dependency Tree

Dependency grammar (DG) is a sort of syntactic theories introduced by Lucien Tesniere [27], which is based on a dependency relation between a governor (a word) and its dependents. Because of benefits of DG such as extracting long-distance relations. Wu et al. [30] endeavored to expand the dependency tree node with syntactic phrases. Term "Phrase" usually is used as a syntactic unit in natural language processing tasks. However, a contiguous non-syntactic phrase, which consists of some contiguous words without any syntactic constraints, is another phenomenon which plays important role in NLP applications such as Phrase-based SMT.

3.1 Phrasal Dependency Tree

In order to construct the phrasal dependency tree, we introduce an algorithm which utilizes a lexical word level dependency parser and a segmentation of the sentence. The segmentation provides the non-syntactic contiguous phrases which cover all words of input sentence. A phrasal dependency tree is defined as follows.

Definition 3.1. Let R= $\{r_1,r_2,\dots, r_m\}$ be a limited set of possible dependency relation type that could hold between any two phrases of a sentence. A phrase dependency graph $G = (V, E)$ is a directed graph consists of nodes, V, and arcs, E, such that for sentence $S=p_0p_1p_2....p_n$ (p_i is a phrase or segment and p_0 is a dummy phrase as a root). V consists of $p_0p_1p_2....p_n$ and E is set of triple $<p_i,r,p_j>$. There is no edge between two phrases with the same relation type.

Definition 3.2. *A phrasal dependency tree* T=(V, E) for an input sentence S and dependency relation set R is a spanning tree rooted by node p_0 which is derived from a dependency graph.

The main idea is to replace a word by a contiguous non-syntactic phrase in a sentence. Thus a dependency relation holds between two phrases. One phrase is a governor and other is a dependent. In order to make relationship between phrases regarding to the word dependency relations, we must distinguish a word as head of the phrase. The dependency relations of two head words of phrases play important role in making relations of phrases.

Definition 3.3. The head word of the phrase P_i is the closest word to the root of the word level dependency tree T in comparison with other P_i words. On the other words, the shallowest word of phrase P_i in the word level dependency tree T.

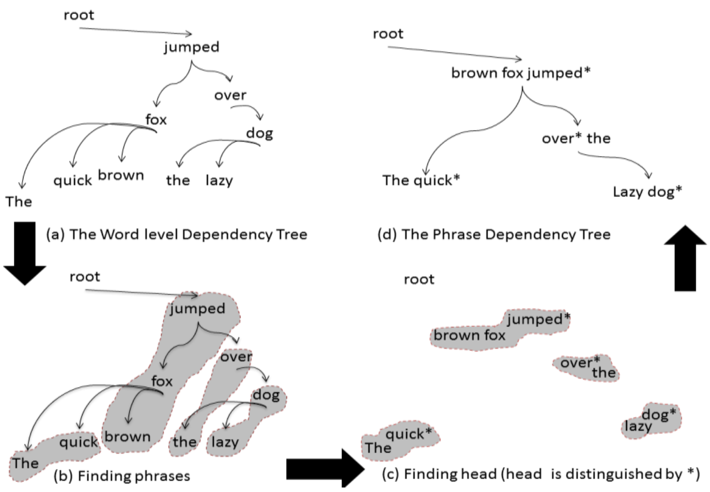


Fig. 1. The procedure of the phrase dependency parsing

The phrasal dependency parsing is conducted in this way: first find a head of each phrase and next travers word level dependency tree in preorder fashion. By visiting a non-head node of a phrase, compact it with the head node of the phrase and remove all its dependency relations. At the end, connect each head node to its nearest ancestor. Consider the following example:

S: “The quick brown fox jumped over the lazy dog”
P[1]: [The quick][brown fox jumped] [over the] [lazy dog]

Fig.1 illustrates the algorithm for the example. Fig.1 (a) demonstrates the word level dependency tree of the sentence S. Fig.1 (b) shows the phrases on the tree. At

[1] P shows a segmentation of S which distinguishes all phrases

the end, each phrase is linked to make a tree. The phrasal dependency tree of sentence S according to the segmentation P is shown in Fig.1 (d).

Note that the output of the algorithm is still a tree because we solely cut edges between neighbor words and generate new edge between a head and its nearest ancestor as well. Additionally, the algorithm guarantees that the output graph's connectivity is maintained and that the graph contains no cycles.

4 Phrasal Reordering Model

As mentioned before, one of the complicated problems in Phrase-based SMT is phrase reordering. We introduce a novel contiguous phrasal reordering model by integrating the phrase dependencies into Phrase-based SMT. The phrase movements are predicated by phrase dependency relations learned from a phrasal dependency corpus. The model depends on calculating the probabilities of the reordering elements, which are estimated via the maximum likelihood estimation from frequencies in a sufficiently-large set of phrasal dependency trees.

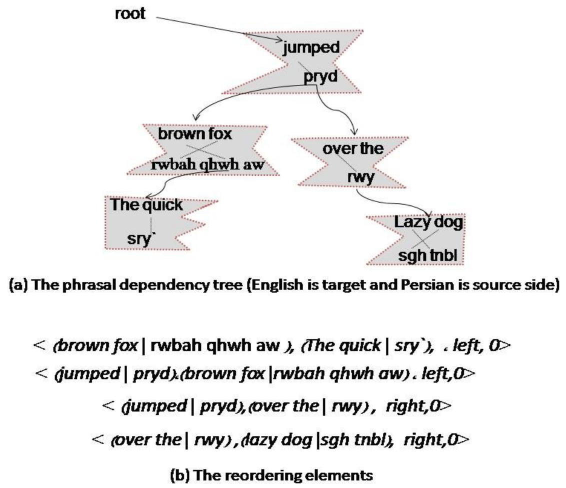


Fig. 2. The phrasal dependency tree and extracted reordering elements (rules)

The reordering element is a branch of the phrasal dependency tree, which depicts the dependency relation between one phrase as a governor (parent) and another phrase as a dependent (child). Fig.2 (a) shows a phrasal dependency tree for two word aligned sentences. The tree has been derived from target order of sentences (English side), and the nodes are constructed by the source and target words plus its word alignments. According to the phrasal dependency tree topology, the dependent node can be settled in the right or left side of the governor node. This direction helps to determine the translation orientation of phrases in decoding phase. Hence, the reordering element has been equipped by the direction of dependent node respect to its governor. All information provided by the reordering elements helps the decoder to score the translation hypothesis more precisely.

The reordering element is shown by:

<governor(target|source), dependent(target|source),direction >

Fig. 2(b) shows the reordering elements, which their probabilities should be computed from a phrasal dependency trees corpus as training data by Eq. 2.

4.1 Training Phase

Given that a reordering element consists of 3elements <g, d, dir >, a total probabilistic model p(<g,d,dir>) is split into 3 partial models as follows.

$$P_{total}(< g, d, dir >) = p(g) * p(d \mid g) * p(dir \mid g, d) \qquad (1)$$

P_{total} *(...)* is the probability of the reordering elements called total model. *P(...)* is the probability of features called partial model. All partial models are learned by the maximum likelihood estimation method and smoothed by the modified Kneser-Ney.

4.2 Decoding Phase

During the decoding phase of the left-to-right decoder, the source sentence is segmented into a series of phrases as in a standard phrase-based model. All standard Phrase-based SMT models with the proposed reordering model are incorporated into a log linear fashion to score the partial translations (hypotheses). In order to score the hypothesis, we use Eq.2 to calculate the reordering probability of H.

$$score(H) = -\log\Big(\prod_{p_i, p_j \in phrase(H)} P_{dep}(p_i, p_j)\Big) \qquad (2)$$

where *H* is a hypothesis and p_i and p_j are non-syntactic contiguous phrases of *H*. p_i is a governor and p_j is a dependent. *Phrase (H)* returns a list of all phrases of *H*. P_{dep} *(...)* is the probability of the dependency reordering elements calculated by Eq. 3.

$$P_{dep}(p_i, p_j) = P_{total}(< p_i, p_j, dir >) \qquad (3)$$

where *dir* is a direction of pi respect to p_j, respectively.

4.3 The data sparseness problem

In order to dominate the data sparseness problem, the head word of a phrase can be utilized by reordering elements instead of incorporating all words of a phrase. Hence the reordering element changes as follows.

<Head (governor), Head (dependent), dir >

Head (governor) and Head (dependent) are head words of governor and dependent phrases, respectively. The big challenge is finding the head words of the governor and dependent during the decoding phase. Due to the lack of word level dependency tree during the decoding phase, determining the head is not straightforward. Here we propose a heuristic approaches to detect head word of phrases during decoding phase. To recognize the head of phrase, POS tags of word has been employed. According to the definition 3.3, the head word is shallowest word respect to the word level dependency tree. In other words, the head plays more luminous syntactically roles than other words of the phrase. For example for given phrase "brown fox jumped" with POS tag sequence "adj[2] noun verb", "verb" is head because of the verb plays more important role than other words.

5 Experimental studies

In order to compare the performance of our reordering model with various reordering models such as the distortion, lexicalized and syntax-based reordering models, some experiments have been carried out by training a Persian→English SMT system. One of the important reasons for choosing Persian and English language pair is lots of differences between English and Persian sentences in the word order.

Persian sentences use SOV (Subject (S), Object (O) and Verb (V)) word order whereas English sentences use SVO structure. Also in Persian language the modifier appears before the modified word whereas English is vice versa. Two validation scenarios have been designed in order to validate the proposed model.

Scenario 1: the aim of the scenario is to validate our model on the small-scale translation task. We intend to understand the impact of our model on translation quality when using the low resource language pairs.

Scenario 2: the performance of our model has been evaluated on large scale training datasets. We intend to show the ability of the model when using several the large-scale translation tasks.

5.1 Data

Table.1 reports translation tasks characteristics. TPC3 [15] with about 400k parallel sentences from novel books has been employed by tsFaEn4-small.

tsFaEn-large utilizes a parallel corpus including about 1 million Persian-English sentences extracted from novel books. PCTS [15] is employed as development and test sets. TMC which is also free Persian monolingual corpus is used to build the target 3-gram language model using the SRILM toolkit with modified Kneser-Ney smoothing [26].

[2] adjective
[3] Tehran Parallel Corpus
[4] Translation task of Persian→English

Table 1. Basic statistics about the novel domain translation tasks (Persian→English)

			Sentence	Token	Unique Token	ASL
tsFaEn-small	Train	source	399K	6M	80K	15.1
		target	399K	6.5M	65K	16.2
	Dev	source	100	1.3K	0.68K	12.4
		target	100	1.4K	0.6K	13.8
	Test	source	300	3.4K	1.4K	11.6
		target	300	3.6K	1.2K	12.1
tsFaEn-large	Train	source	1M	15.6M	250K	15.6
		target	1M	15.7M	210K	15.7
	Dev	source	200	2.1K	1K	10.8
		target	200	2.2K	1K	11.3
	Test	source	200	2.6K	1.1K	13
		target	200	2.2K	0.97K	11.3

5.2 Baseline System Setup

Moses has been employed as a baseline Phrase-based SMT [10, 12] and Hierarchical Phrase-based SMT [4]. Phrase-based SMT utilizes multiple stacks to generate translation hypothesis and SRILM toolkit [25] with interpolated modified Kneyser-Ney smoothing to compute 3-gram language model.

The parameters used for the experiments are: stack size of 100 and the number of target phrases limit of 20. Alignments have been extracted by utilizing the GIZA++ toolkit in words level [17, 18]. Distortion limit equals -1 for the SMT systems equipped by the proposed reordering model.

The hierarchical Phrase-based SMT system utilizes the standard default Moses configuration and relative_thereshold[5]=10 and max_n_item[6]=30.

In order to evaluate the translations, BLEU [20],TER [24] and LRscore [2] measures are used. All model weights have been tuned on development sets via minimum-error rate training (MERT) [16]. The word level dependency tree is generated by the Stanford dependency parser [6].

5.3 Experiments on Small-scale Training Data

In order to illustrate the performance of the reordering models in term of BLEU, TER and LRScore, 4 translation systems with different reordering models have been built on the same conditions. The results on the small-scale translation tasks, tsFaEn-small, have been reported in Table.2. Phrase-based SMT with the distortion, lexicalized and proposed reordering model are denoted by ***pbSMT+d, pbSMT+l*** and ***pbSMT+p***, respectively. ***hpbSMT*** also points to hierarchical Phrase-based SMT.

[5] *Relative_threshold* prunes items in a cell which is worse than the best item in that cell

[6] The maximum number of items which a cell can maintain

Table 2. BLEU, TER and LRscore scores of all systems with different reordering models

		Metrics			
		DEV	TEST		
		BLEU	BLEU	TER	LRSCORE
tsFaEn-small	*pbSMT+d*	26.71	22.96	62.01	0.32
	pbSMT+l	29.10	25.69	59.75	0.34
	hpbSMT	29.19	26.03	59.23	---
	pbSMT+p	**30.31**	**27.01**	**58.39**	**0.36**

As shown in Table 2, the best BLEU/TER/LRscore scores of translation systems (tsFaEn-small) are about 27/58.3/0.36. The proposed model achieves +1.32/+4.05/+0.98, -1.4/-3.69/-0.84 and +0.02/+0.04/0 point improvements in BLEU, TER and LRscore compared with the lexicalized /distortion/hierarchical models, respectively. Therefore, we can observe that adding our reordering model to Phrase-based SMT brings an illustrious improvement on the small-scale translation tasks with different domains and average sentence length.

In order to investigate the reordering predictive capabilities of models, the minimum number of shifts needed to change a system output so that it exactly matches a given references have been computed.

The shift moves a sequence of words within the translation (for more details [24]), and also shift distance indicates to the number of reordering required to move a word to its right place respect to the given references. Fig. 3 presents the amount of the shifts needed by ***pbSMT+d***, ***pbSMT+l*** and ***pbSMT+p*** on PCTS test set.

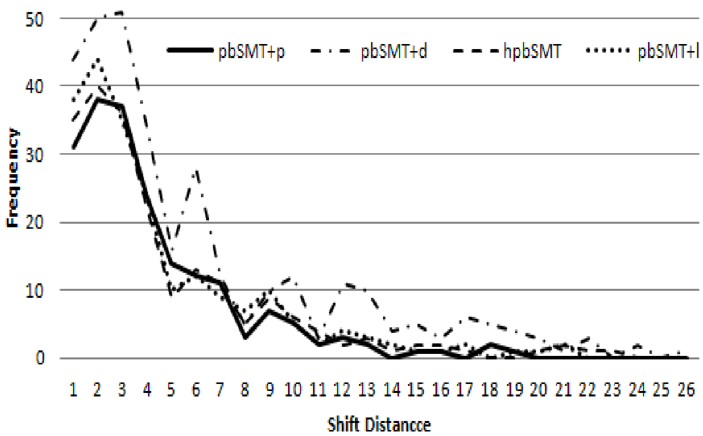


Fig. 3. The number of the reorderings needed by pbSMT+d, pbSMT+l, pbSMT+p to match with the targeted references on PCTS test set

Fig.3 indicates that our model predicates a lot more reordering needed particularly medium- and long- distance reorderings to the translation than the other reordering models. For more analysis, we calculate precision and recall of reorderings [9]. Table

3 reports the total precision and recall which are computed test set and aligned manually.

Table 3. Total Precision and recall

Translation System	Total Precision	Total Recall
pbSMT+d	0.29	0.31
***pbSMT*+l**	0.31	**0.32**
***pbSMT*+p**	**0.33**	**0.32**

From Table 3, we can observe that our model improves precision about +0.02 and +0.04 absolute points respect to the lexicalized and distortion models, respectively. In order to explore the question which word ranges are affected more by the reordering models, Fig.4 shows precision per the reordering distance, respectively. It is figure out that our model has the most positive impact on precision over the most word ranges. The results demonstrate superiority of our model whereas the lexicalized model overtakes in 6, 7, 10 and 11. The error analysis reveals that in the most cases, the proposed model has been predicated the direction of translated phrases correctly. Nevertheless, because of the existence of untranslated phrases, the length of the translations generated by ***pbSMT+p*** is less than others. Consequently, unexpected results observed in 6, 7, 10 and 11. In general, we can observe the momentous improvement on the short-, medium- and even long- distance reorderings. When recall is concerned, our model achieves a comparable recall value respect to the lexicalized reordering model.

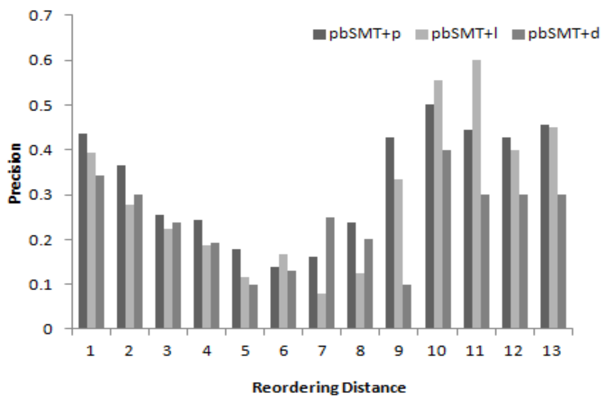


Fig. 4. Precision/reordering distance

5.4 Experiments on Large-scale Training Data

In previous section, we study sparse training data scenarios, in which the reordering and translation models have been learned on two sparse bilingual data sets. In this section we scale the method to a large training set and illustrate that the improvement in terms of translation quality is maintained. Table 4 presents the results of our model in comparison with the other reordering models.

Table 4. BLEU, TER and LRscore scores of all systems with different reordering models

		Metrics			
		DEV	TEST		
		BLEU	BLEU	TER	LRSCORE
	pbSMT+d	30.11	26.96	59.17	0.35
tsFaEn-large	*pbSMT+l*	33.13	29.03	56.57	0.37
	hpbSMT	**33.45**	29.0	56.56	---
	pbSMT+p	33.33	**30.07**	**55.63**	**0.38**

The best BLEU/TER/LRscore scores of translation systems are about 30/55.63/0.38. The proposed model achieves +1.02/+3.11/+1.07 and -0.94/-3.54/-0.94 point improvements in BLEU and TER compared with the lexicalized /distortion/hierarchical models, respectively.

Similar to the small-scale experiments, the reordering predictive capabilities of the models on the large-scale translation tasks have been considered by the number of needed shifts, reordering precision and recall. Fig. 5 depicts the amount of shifts needed by ***pbSMT+d, pbSMT+l*** and ***pbSMT+p***.

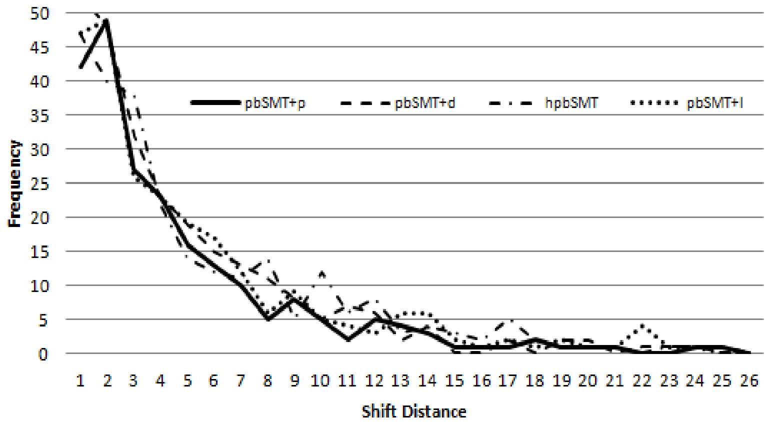


Fig. 5. The number of the reorderings needed by pbSMT+d, pbSMT+l, pbSMT+p

As Fig. 5 shows, pbSMT+p has been more successful than other models in the prediction of the reordering needed. It indicates that our model predicates a lot more reordering needed particularly medium- and long- distance reorderings than the other reordering models. The experiments on the large-scale translation tasks also implies that the proposed model not only obtains better results over the well-known and popular reordering models but also can predicate the medium- and long- distance needed reorderings more than others.

6 Conclusion

In this paper, we introduce a new phrasal reordering model of integrating the phrase dependencies as syntactical structure to the Phrase-base SMT. We exploit the syntactically-informed reordering elements which are included by the translation direction feature in order to deal with the medium- and long- distance reordering problems. The proposed model has been discussed from the theoretical and experimental points of view, and its advantages, disadvantages and constraints in comparison of well-known and popular reordering models have been analyzed. In order to compare the performance of our reordering model with the distortion, lexicalized and hierarchical reordering models, lots of experiments have been carried out by training Persian→English SMT systems. We evaluated the proposed model on two translation tasks in different size. The evaluations illustrate significant improvements in BLEU, TER and LRscore scores comparing to the lexicalized /distortion/hierarchical models. Furthermore, the reordering predictive capabilities of models have been compared by calculating the minimum number of shifts needed to change a system output so that it exactly matches a given references. The results imply that our model predicates a lot more reordering needed particularly medium- and long- distance reorderings than the other reordering models. For a more detailed analysis and answering the question which word ranges are affected more by the reordering models, total precision/recall and precision/recall per distance have been calculated. The proposed model retrieved a significant impact on precision with comparable recall value respect to the lexical reordering model.

References

1. Bach, N., Vogel, S., and Cherry, C.: Cohesive constraints in a beam search phrase-based decoder. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers, pp. 1–4 (2009)
2. Birch, A. and Osborne, M.: LRscore for evaluating lexical and reordering quality in MT. In: Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR, pp. 327–332 (2010)
3. Chang, P. C. and Toutanova, K.: A Discriminative SyntacticWord Order Model for Machine Translation. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, Prague, Czech Republic, pp. 9–16 (2007)
4. Chiang, D.: A hierarchical phrase-based model for statistical machine translation. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pp. 263–270 (2005)
5. Chiang, D.: A hierarchical phrase-based model for statistical machine translation. In: Proc. of ACL (2005)
6. De Marneffe, M.-C., MacCartney, B., Manning, C.D.: Generating typed dependency parses from phrase structure parses. In: Proceedings of LREC, pp. 449–454 (2006)
7. Galley, M., Graehl, J., Knight, J., Marcu, D., DeNeefe, S., Wang, W., and Thayer, I.: Scalable Inference and Training of Context-Rich Syntactic Translation Models. In: Proceedings of the joint conference of the International Committee on Computational Linguistics and the Association for Computational Linguistics, Sydney, Australia (2006)
8. Galley, M. and Manning, C.D.: A Simple and Effective Hierarchical Phrase Reordering Model. In: Proceedings of the EMNLP 2008 (2008)

9. Gao, Y., Koehn, P., and Birch, A.: Soft dependency constraints for reordering in hierarchical phrase-based translation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 857–868 (2011)
10. Koehn, P., Hieu Hoang, Birch, A.: Moses: Open source toolkit for statistical machine translation. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (2007)
11. Koehn, P., Axelrod, A., Mayne, A., Callison-Burch, C., Osborne, M., and Talbot, D.: Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation. In: International Workshop on Spoken Language Translation (2005)
12. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, R., Bojar, O., Constantin, A., Herbst, A.: Moses: Open source toolkit for statistical machine translation. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, 2007.
13. Koehn, P., Och, J., and Marcu, D.: Statistical Phrase-Based Translation. In: Proceedings of HLT/NAACL (2003)
14. Kumar, S. and Byrne, W.: Local phrase reordering models for statistical machine translation. In: Proceedings of HLT-EMNLP (2005)
15. Mansoori, A. and Faili, H.: State-of-the-art English to Persian Statistical Machine Translation System. In: Proceeding of 16th CSI International Symposiums on Artificial Intelligence and Signal Processing (AISP 2012) to appear, Shiraz, Iran (2012)
16. Och, F. J.: Minimum error rate training in statistical machine translation. In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1, pp. 160–167 (2003)
17. Och, F. J. and Ney, H.: Improved statistical alignment models. In: Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics, pp. 440–447 (2003)
18. Och, F. J., Ney, H.: Improved statistical alignment models, In: Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics, pp. 440–447 (2003)
19. Och, J. and Ney, H.: The alignment template approach to statistical machine translation. Computational Linguistics, vol. 30, pp. 417–449 (2004)
20. Papineni, K., Roukos, S., Ward, T., and Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: Proc. of 40th Annual meeting of the Association for Computational Linguistics, pp. 311–318 (2002)
21. Quirk, C., Menezes, A., and Cherry, C.: Dependency treelet translation: Syntactically informed phrasal SMT. In: proceedings of the 43th Meeting of the Association for Computational Linguistics, pp. 271–279 (2005)
22. Shen, L., Xu, J., and Weischedel, R.: A new string-to-dependency machine translation algorithm with a target dependency language model. In: Proceedings of ACL-08: HLT, pp. 577–585 (2008)
23. Shen, L., Xu, J., and Weischedel, R.: A new string to-dependency machine translation algorithm with a target dependency language model. In: Proc. of ACL (2008)
24. Snover, M., Dorr, B., Schwartz, R., Micciula, L., and Makhoul, J.: A Study of Translation Edit Rate with targeted Human Annotation. In: AMTA 2006, 7th Conference of the Association for Machine Translation in the Americas, Cambridge, pp. 223–231 (2006)
25. Stolcke, A.: SRILM – an extensible language modeling toolkit. In: Proceedings of the international conference on spoken language processing, pp. 901–904 (2002)
26. Stolcke, A.: SRILM – an extensible language modeling toolkit. In: Proceedings of the International Conference on Spoken Language Processing (ICSLP 2002) (2002)
27. Tesniere, L.: El´ements de syntaxe structurale. Editions Klincksieck (1959)
28. Tillmann, C.: A block orientation model for statistical machine translation. In: HLTNAACL, Boston, MA, USA (2004)

29. Wu, D.: Stochastic inversion transduction grammars, with application to segmentation, bracketing, and alignment of parallel corpora. In: Proceeding of IJCAL 1995, Montreal, pp. 1328–1334 (1995)
30. Wu, Y., Zhang, Q., Huang, X., and Wu, L.: Phrase Dependency Parsing for Opinion Mining. In: Proceedings of the 2009 Conference on Empirical Methods in natural Language Processing, pp. 1533–1541 (2009)
31. Zens, R., Ney, H., Watanabe, T., and Sumita, E.: Reordering Constraints for Phrase-Based Statistical Machine Translation. In: Proceedings of CoLing 2004, Geneva, Switzerland, pp. 205–211 (2004)

An Approach to Semantic Natural Language Processing of Russian Texts

Valery Solovyev[1], Vladimir Polyakov[2], Vladimir Ivanov[1],
Ivan Anisimov[3], Andrey Ponomarev[3]

[1] Kazan Federal University, Kazan, Russia
[2] Institute of Linguistics, Russian Academy of Sciences, Moscow, Russia
[3] National Research Technologic University "MISiS", Moscow, Russia

{maki.solovyev, nomemm}@mail.ru, pvn-65@mail.ru,
{valeter, genhoo}@mail.ru

Abstract. The article contains results of the first stage of a research and development project aimed at creating a new generation of intellectual systems for semantic text analysis. Described are the main principles, system architecture, and task list. The features cloud and cluster architecture realization are regarded as well.

Keywords: Paradigmatics, syntagmatics, parallel computing, cloud computing, natural language processing.

1 Introduction

The history of the development of natural-language text processing began over half a century ago; it has its ups and downs, however, as an industrial technology within the framework of informational technologies, this direction has been forming during the past twenty years. Nowadays, there are several program libraries of Natural Language Text Processing (NLTP) in Russia and abroad, and they can be regarded as positive examples of the complex solution of the problem (AOT [26], RCO [27], DICTUM [28], GATE [29], UIMA [30], OpenNLP [31]). However, the overall progress in the sphere of IT collected in the field of NLTP systems creation, particularly, and intellectual systems in general, allows looking at the problem of NLTP from another angle.

The topicality of parallel and distributed computing use in the sphere of NLTP is defined by the following factors:

There is need. Today there are millions of terabytes of texts in the Internet and in the corporative medium that are of great interest from the point of view of machine learning, creating of more powerful search systems, and new cutting-edge applications in the sphere of NLTP.

There is possibility. The task of NLTP suits for both parallel and distributed computing, due to the well-known autonomy of text units on each of its levels (tokens, morphoforms, chunks, sentences). This autonomy allows organizing mass parallel computation of numerous NLTP tasks.

Technical preconditions in the sphere of IT. There has been achieved a technological limit of increasing the processing power of computers by increasing the clock rate. The world leading processors producer, Intel company, stopped producing mononuclear processors. The growth of processing power is more and more conditioned by the architecture of the computation system. Computation resources have become much cheaper and more available. The carrying capacity of the channels of digital communication has reached the level when it has stopped being an obstacle for the organization of distributed computing in the net.

2 Principles for Developing of the New Generation Technologies for Natural Language Processing

Our group of researchers has formulated the principles, which must underlie the creation of program libraries and new generation technologies for natural language processing.

Division of algorithms and data. As a rule, in modern systems of NLTP, algorithms and data are so deeply intertwined that it is almost impossible to use them separately.

We suggest to follow the principle of division, thus, it is possible to use ready sets of data and change the algorithms to a more powerful one, and vice versa, by only changing the algorithms, which makes the researcher's work considerably easier.

Similarly, third-party developers can specialize in ready sets of data for linguistic support of standard algorithms and program libraries.

This principle is widely used by the developers of foreign systems of natural language processing [29-31].

Open algorithm standards and data formats. A consequence of the first principle, which allows comparing achievements in NLP sphere on a healthy competitive basis.

This does not mean open program code or other objects of intellectual property.

Pipeline architecture. At present the market has a positive experience of creating program platforms on the base of Java language, which serve for the aim of integration of packets for NLP. Nowadays, the most popular projects are GATE, UIMA and OpenNLP. But there is a clear lack of libraries for Russian language processing compatible with the mentioned platforms. Developing business oriented applications for Russian language based on the stated platforms is laborious both in terms of time and technology.

The solution for overcoming this obstacle is to simplify interaction with the platform on the user level, where software can be used as a service within the framework of strictly regulated scenarios and for solution of user's certain tasks.

Iterativity. Due to the ambiguity of language on all of its levels, it is impossible to get a 100% precise result in the process of NLTP. Well known causes of this ambiguity are homonymy, lexical polysemy and syntactical polysemy.

The suggested solution is to reiterate different levels of analysis, repeatedly running tasks after some ambiguity was resolved.

For example, preliminary morphological analysis + chunking [1] + secondary morphological analysis (resolution of polysemy), etc.

Frequency, F1-measure, of the mini-corpus. Improvement of each type of analysis should be based on the frequency of the occurring phenomena. For example, before coping with homonymy, one should conduct its frequency analysis and work with its most frequent cases.

F1-measure is a reliable way to check the quality of the analysis. For every text processing task a reference mini-corpus (a set of texts with a reference marking) should be created to test the developed methods.

Practice shows that such mini-corpora hand-marked for 10-100 uses of the phenomena in question (1000-10000 thousand words) are sufficient at the current stage of NLTP development.

Orientation on the technologies of data extraction from the text (Information Extraction). It is already evident that Information Extraction technology is becoming the most real alternative for complete NLP, which is, probably, unattainable in the nearest decade.

The authors of the project developed a method to extract relations, which considerably lowers the work content of application creation, as it does not require a big set of teaching examples. It is planned to integrate this method in the library as a separate application for open information extraction.

Orientation on ontologies. Ontologies are the most standardized format of paradigmatic data representation, this allows to build on the existing technological systems. Languages of ontologies presentation (RDFS, OWL) are now applied for program components description (OWL-S) and their data sets (SKOS).

Machine learning. The renunciation of manual language specification. One should try to create products with minimal hand labor of linguists. Orientation on technologies that do not require great volume of manual work, such as open information extraction, will allow to efficiently adapt the created applications for the platform of cloud computing.

Multilanguage. The architecture and design of separate modules must not obstruct the creation of multilingual systems (search engines, machine translation, etc.).

Multifunction. If the functions of NLTP are well standardized, the applications of the whole system or its parts can be diverse.

Cloud computing. Technologies of cloud computing [20] seem to be the most suitable means of computation needed for large scale natural language processing. Cloud computing eliminates a broad range of issues connected with the producing capacity of the machinery, availability of services due to high technological and price barriers that companies, novices and research groups creating applications for intellectual analysis in any subject field have to overcome.

[1] Chunking is, in Russian NLTP tradition, breaking a sentence into minimal semantically meaningful parts (chunks).

Outside of Russia, besides widely spread Amazon Web Services [32], Windows Azure [33] and Google App Engine [34], there are such popular projects as OpenNebula [35] and Ubuntu Enterprise Cloud [36].

Open instruments. Application of standard platform like GATE and UIMA, openness on the level of branch standards for algorithms and formats, allows incessantly developing new instruments for NLTP systems. It is known that most companies and research groups cannot work on this problem because of a lack of access to effective text processing libraries, and the fact that development of such libraries takes a lot of time and costs. Even if there are free libraries, there arises the question of their installation, launch and integration. Creation of a complex of language processing tools on a unified technological platform will simplify the solution of this problem as well.

Commercialization of temporary results of scientific research work. There appear preconditions for integration of partial developments in the sphere of NLP into a united system, the process of program products maintenance becomes easier.

On the base of above mentioned principles, we formed a structure of program components of the future program complex (fig. 1).

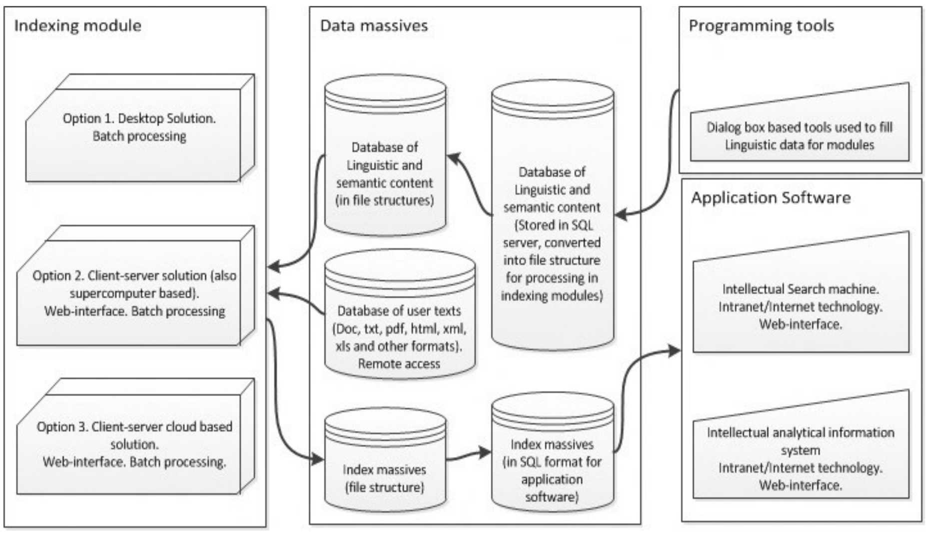


Fig. 1. Future program complex architecture

3 NLTP Tasks Included in the Library

The question of the content of the tasks to be included in the program library was deduced from the balance of desirable and maximally possible current development of science, technology and NLTP practice.

Classically [4, 8], it is believed that the process of Russian text processing is divided in to three stages:

- morphologic analysis;
- syntactical analysis;
- semantic analysis.

However, detailed study of the state of the art in the sphere of NLTP shows:

1) This statement does not take into consideration such important stages as preliminary processing of the text (conversion of formats, codes, clearance of control characters), tokenization (exposure of symbol chains, processing of extralinguistic data), pragmatic text processing (exposure of the information value with respect to the task at hand, etc., associating to the context);

2) The level of development of the tasks at each of the three macro-stages can be different. Thus, one can say that:

- The stage of morphological analysis is well elaborated from the scientific, as well as engineer point of view6;
- The stage of syntactic analysis is elaborated from the scientific point of view, but from the point of view of program engineering there is still considerable work to be done on optimal solutions in computation performance;
- The stage of semantic analysis still has a lot of gaps, and it is far from the completion of research. However some elements of semantic analysis (ontologies, semantic roles) are elaborated enough to provide the possibility of their engineer realization.

Considering these circumstances, our work group developed a list and a sequence of NLTP tasks, which we decided to include into the basic software library of NLTP[2]:

1. Preprocessing [20, 21, 24];
2. Tokenization-1 [22];
3. Morphological analysis;
4. Stemming[3] [1, 2];
5. Prediction of morphological characteristics[4];
6. Segmentation[5]-1 [7, 13];
7. Tokenization-2;
8. Set phrases and idioms identification [9, 10];
9. Building of broadened grammar vector[6];

---

[2] Working project title is “NLP@Cloud”.

[3] In our case stemming includes not only finding the stem of a word, but also resolving the morphologic structure of a lemma.

[4] Used for words not found in the dictionary.

[5] In our case the task of segmentation is to identify syntactic constructions which are separated by punctuation marks in complex sentences.

[6] Broadened grammar vector – a special notation, used to transit from shallow to deep syntactic analysis

10. Chunking-1 [3, 6, 11, 12, 21];
11. NER (Named Entity Recognition) [14, 17];
12. IER (Identified Entity Recognition)[7][23];
13. NPR (Noun Phrase Recognition);
14. Morphological analysis-2. Homonymy resolution-1 [15, 16];
15. Segmentation-2;
16. Thematic classification of the text [18, 19];
17. Identifying communicative meaning of inquiries[8];
18. Binding to ontology. Homonymy resolution-2. Polysemy resolution-1;
19. Text nucleus detection[9];
20. Chunking-2;
21. Syntactic tree analysis[10]-1;
22. Referential analysis[11]-1;
23. Detection of actant semantic roles;
24. Binding to ontology-2. Homonymy resolution -3. Polysemy resolution -2;
25. Syntactic analysis-tree-2;
26. Referential analysis-2;
27. Connotative classification[12];
28. Identification of new concepts and their binding to ontology[13].

We can say that most tasks connected with morphology and syntax processing are included in the basic set of library functions.

It also includes tasks from the semantic level, which are mainly based on paradigmatic structures of knowledge (ontologies). Semantic roles, referential links and meanings belong to the syntagmatic sphere. Everything concerning other syntagmatic relations (temporal, spatial, causal and other links and relations) are currently left out of the basic library of NLTP. Due to incomplete distinctness in standards of semantic processing and high computation loads necessary for this kind of analysis, tasks related to syntagmatics will be included in user-end applications.

We can speak of a new standard of NLTP, which is introduced in this work. This standard set of NLTP instruments can be called paradigm-oriented text processing, or POTP.

---

[7] Identifying entities which have a numeric, digital or temporal quality.

[8] Identifying communicative meaning of inquiries is finding a relevant area of human usage for piece of text. This is usually different from thematic classification.

[9] By text nucleus we imply the most frequent meaningful word or entity in a text. A nucleus can be graded (i.e. include multiple words and entities ranged by frequency).

[10] Dependency grammar trees are used.

[11] Anaphora and cataphora links are marked, the denotative status of concepts is identified, referential ambiguity is resolved.

[12] Connotative classification is based on entities rather than texts as a whole.

[13] Based on WordNet [25] lexical ontology.

4 Conclusion

Let us enumerate the main innovations suggested in this work, which were not applied in industrial developments of NLTP before or were applied on a more limited scale, including:

- Iterativity;
- Chunking;
- Broadened grammar vector;
- Communicative classification;
- Connotative classification be objects;
- Text nucleus (graduated);
- Automatic building of ontology;
- Improved resolution of ambiguity (morphological, lexical, syntactic);
- Use of mechanism of semantic roles;
- Technology of text classification Rubryx [19];
- Lexical and syntactic portraits for resolution of lexical polysemy;
- Orientation to high-capacity computing.

An important feature of the suggested solution is its flexibility, that allows setting the complex of program libraries for different applied tasks. Openness (in the functional) secures the possibility of the system development in future. The suggested technologies are based on the latest achievements in Semantic Web sphere and ontological systems.

Acknowledgments. The work is supported by grants of Foundation for Assistance to Small Innovative Enterprises, Russia (program Soft-2012, contract №10151p/17593 (28.04.2012), program UMNIK, contract №16925/15-12 (21.05.2012)). The project is performed under a state assignment to universities by Ministry of Education and Sciences of Russia (№ 8.3358.2011). It is a joint project of National University of Sciences and Technology “MISiS” and Kazan Federal University.

References

1. Alvares, R. V., Mondaini, R.: Evaluation of Stemming Errors: Towards a Qualitative Analysis. In: XXXI CNMAC Conference proceedings (1999)
2. Alvares, R. V.; Garcia, A. C. B.; Ferraz, I. N.: STEMBR: A Stemming Algorithm for the Brazilian Portuguese Language. In: 12th Portuguese Conference on Artificial Intelligence (EPIA 2005), Covilhã, Portugal. Lecture Notes in Artificial Intelligence. v. 3808, pp. 693–701 (2005)
3. Attardi, G., Dell'Orletta, F.: Chunking and Dependency Parsing. In: Proceedings of LREC 2008 Workshop on Partial Parsing, Marrakech (2008)
4. Handbook of Natural Language Processing. Second Edition. Edited by Nitin Indurkhya, Fred J. Damerau. CRC Press, 666 p. (2010)

5. Antonopoulos, N., Gillam, L.: Cloud Computing: Principles, Systems and Applications. Springer, 379 p. (2010)
6. Hacioglu, K.: A lightweight semantic chunking model based on tagging / HLT-NAACL-Short '04 Proceedings of HLT-NAACL 2004: Short Papers (2004)
7. Huang, X., Peng, F., Schuurmans, D., Cercone, N., Robertson, S. E.: Applying Machine Learning to Text Segmentation for Information Retrieval. Information Retrieval (2003)
8. Jurafsky, D., Martin J. H.: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice Hall, 988 p. (2009)
9. Khokhlova, M.: Applying Word Sketches to Russian. In: Proceedings of Raslan 2009. Recent Advances in Slavonic Natural Language Processing. Brno: Masaryk University, pp. 91–99 (2009)
10. Khokhlova, M., Zakharov, V.: Statistical collocability of Russian verbs. In: After Half a Century of Slavonic Natural Language Processing.Brno, pp. 105–112 (2009)
11. Koeling, R.: Chunking with maximum entropy models. In: ConLL '00 Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning , volume 7, pp. 139–141 (2000)
12. Kudo, T.: Japanese dependency analysis using cascaded chunking. In: Proc. of COLING-02, proceedings of the 6th conference on Natural language learning. Volume 20, pp. 1–7 (2002)
13. Lobanov, B., Tsirulnik, L.: Statistical study of speaker's peculiarities of utterances into phrases segmentation. In: Speech Prosody: proceedings of the 3rd International conference, Dresden, Germany, May 2-5, 2006. V. 2, pp. 557–560 (2006)
14. Masayuki, A., Matsumoto, Y.: Japanese Named Entity Extraction with Redundant Morphological Analysis. In: Proc. Human Language Technology conference, North American chapter of the Association for Computational Linguistics. (2003)
15. Mihalcea, R.: Using Wikipedia for Automatic Word Sense Disambiguation. In: Proc. of the North American Chapter of the Association for Computational Linguistics (NAACL 2007), Rochester, April 2007 (2007)
16. Old, L.J.: Homograph disambiguation using formal concept analysis. In: Fourth International Conference on Formal Concept Analysis, 13th-17th February 2006, Dresden, Germany (2006)
17. Poibeau, Th. and Kosseim, L.: Proper Name Extraction from Non-Journalistic Texts. In: Proc. Computational Linguistics in the Netherlands (2001)
18. Polyakov, V. and Sinitsyn, V.: Method for automatic classification of web-resource by patterns in text processing and cognitive technologies. Text Collection, No.6, Publ. House Otechestvo, pp. 120–126 (2001)
19. Polyakov, V. and Sinitsyn, V.: RUBRYX: technology of text classification using lexical meaning based approach. In: Proc. of Intern. Conf. Speech and Computing (SPECOM-2003), Moscow, MSLU p. 137–143 (2003)
20. Prabhu, C. S. R.: Grid and Cluster Computing. PHI Learning (2013)
21. Erik, F., Kim Sang, T.: Introduction to the CoNLL-2000 shared task: chunking. In: ConLL '00 Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning, Volume 7, pp. 127–132 (2000)
22. Schmidt, H.: Tokenizing. In Corpus Linguistics: An International Handbook. Walter de Gruyter, Berlin (2007)
23. Soraluze, A., Alegria, I., Ansa, O., Arregi, O., and Arregi, X.: Recognition and Classification of Numerical Entities in Basque., In: Proceeding of Recent Advances in Natural Language Processing, RANLP 2011, 12-14 September, Hissar, Bulgaria (2011)

24. Zhu, X.: Common Preprocessing Steps. CS769 Spring 2010 Advanced Natural Language Processing (2010)
25. WordNet, lexical database for English, http://wordnet.princeton.edu
26. Automatic Russian Text Processing project, http://aot.ru
27. RCO, search and analytic systems developer, http://www.rco.ru
28. DICTUM, text analysis tools developer, http://www.dictum.ru
29. General Architecture for Text Engineering, http://gate.ac.uk
30. Unstructured Information Management Architecture, http://uima.apache.org
31. OpenNLP, http://opennlp.apache.org
32. Amazon Web Services, http://aws.amazon.com
33. Windows Azure, http://www.windowsazure.com
34. Google App Engine, https://appengine.google.com
35. Lutz, S., Keith, J., Burkhard, N.: The Future of Cloud Computing. Opportunities for European Cloud Computing beyond 2010. European Commission Expert Group Report (2010)
36. Ubuntu Cloud, http://help.ubuntu.com/community/UEC

Analysis of Asymmetric Measures for Performance Estimation of a Sentiment Classifier

Diego Uribe, Arturo Urquizo, Enrique Cuan

Instituto Tecnológico de la Laguna,
División de Estudios de Posgrado e Investigación,
Revolución y Cuauhtémoc, Torreón, Coah., Mexico
{diego,aurquizo,kcuan}@itlalaguna.edu.mx

Abstract. The development of a sentiment classifier experiences two problems to cope with: the demand of large amounts of labelled training data and a decrease in performance when the classifier is applied to a different domain. In this paper, we attempt to address this problem by exploring a number of metrics that try to predict the cross-domain performance of a sentiment classifier through the analysis of divergence between several probability distributions. In particular, we apply similarity measures to compare different domains and investigate the implications of using non-symmetric measures for contrasting feature distributions. We find that quantifying the difference between domains is useful to predict which domain has a feature distribution most similar to the target domain.

Keywords: Sentiment classifier, performance estimation, asymmetric measures.

1 Introduction

Domain adaptation is a common problem in several computational linguistic tasks. Information extraction is a task that takes unseen texts as input and produces structured-unambiguous data as output. However, it is a domain dependent task since when we need to extract information from a new domain, a new ad-hoc system is demanded. But building an information extraction system is difficult and time consuming [7], [5]. Similar challenges are also addressed by an open domain question answering system, a system to obtain concise answers to questions stated in natural language, that needs to be adaptable to play a crucial role in business intelligence applications [17].

Opinion mining is another very interesting computational linguistic task concerned with the classification of the reviews posted by the users, as well as the identification of the aspects of an object that people like or dislike [10], [6]. Since the object might be a product, a service, an organization, etc., opinion mining is also a domain dependent computational linguistic task. As reviews in different domains may be expressed in very different ways, training a classifier using data from one domain may fail when testing against data from another one. In other words, we have to cope with a harder problem when the available training instances are dissimilar to the target domain. Aue and Gamon illustrate how the accuracy of a classifier trained on a different domain drops significantly compared to the performance of a classifier trained on its own native domain [2]. Thus, to determine which subset of outside domains has a feature distribution most similar to the target domain is of paramount importance.

In this study, we focus our attention in the analysis of the distributions corresponding to different domains in order to look for similarities that allow us to optimize the use of the available data. Said in another way, our aim is to look for differences between domains by using divergence measures. We show in this paper how two unannotated and different datasets are used to measure the contrast between their corresponding feature distributions. Once the contrast is determined, we can estimate the performance on the target domain B of an opinion classifier trained on the domain A. Since by using a non-symmetric measure we can obtain two similarity scores (AB and BA), it is also possible to estimate the performance on the target domain A of an opinion classifier trained on the domain B. As we analyse several domains, we may decide to implement a generic classifier depending on the similarity between one domain and other distinct domains.

We evaluate our approach with a data collection of several domains [15]. The results of the experimentation conducted show how the quantification of the divergence among domains is worthwhile to predict the domain with a feature distribution similar to a new target data.

The description of our work is organized as follows. The next section 2 makes a brief review of previous work on the domain adaptation problem in sentiment analysis. Section 3 describes in detail the divergence measures used in our analysis. Section 4 defines the dataset used in our experimentation as well as the pre-processing task for the extraction of the linguistic features to which we submitted our data collection. Then, the results of the experimentation are exhibited and discussed in section 5. Finally, conclusions are given in section 6.

2 Related Work

In this section we briefly describe some of the substantial works dealing with the problem of domain adaptation in sentiment classification. One of the first works in this specific topic was carried out by Aue and Gamon [2]. Their work is based on multiple ways of fusion of labeled data from other domains and unlabeled data from the target domain. The best results were obtained with an approach based on bootstrapping techniques. Shou-Shan et al. [14] propose an interesting algorithm for multi-domain sentiment classification based on the combination of learners for specific domains called *member classifiers*. The combination of these member classifiers is done according to two types of rules: fixed and trained rules. The purpose of the combination process is to obtain and to make available global information to the final classifier. Likewise, Blitzer et al. [3] cope with the domain adaptation problem by extending an algorithm for sentiment classifier by making use of *pivot features* that relate the source and target domains. This relationship is defined in terms of frequency and mutual information estimation.

Also, there are significant works about comparing corpora to explore how corpus properties can affect the performance of natural language processing (NLP) tools. Sekine studied the effect of the use of different corpora on parsing tools [13]. Another interesting work to predict the cross-domain performance of an NLP tool was carried out by Asch and Daelemans [1]. Their work makes use of six similarity metrics to measure the difference between two corpora. Once the similarity is calculated, they investigate the correlation between similarity and the performance of an NLP tool such as a part-of-speech (POS) tagger. Other investigation concerned with the adaptation problem was carried out by Mansour et al. [11]. In their work, they make use of the Rényi divergence measure [12] to estimate the distance between diverse distributions.

3 Approach

In this section, we describe in detail our approach to estimate the subset of different domains with a feature distribution similar to the target domain. Unlike traditional supervised learning, adaptive learning entails the necessity to extract and exploit metaknowledge to assist the user in the task of selecting a suitable predictive model while taking into account the domain of application. One form of metaknowledge is to obtain insight about the data distribution [4]. We analyze in this work two ways to look for differences between domains by using non-symmetric divergence measures such as: Kullback-Leibler divergence [8] and cross-entropy.

3.1 Kullback-Leibler (KL) Divergence

The KL divergence (also known as relative entropy) is a measure of how different two probability distributions are. To be more specific, the KL divergence of q from p, denoted by $D(p \parallel q)$, is a measure of the information lost when q is used to approximate p.

$$D(p \parallel q) = \sum_{x} p(x) \log \frac{p(x)}{q(x)} \tag{1}$$

As the relative entropy of a target dataset, given a source dataset, is the data required to reconstruct the target, our interest in this divergence measure consists in to observe the behavior of a learning tool for the domain B that has been trained in terms of the domain A. In other words, we are interested in the performance estimation of a learning tool for B when using a feature set corresponding to A, rather than to B. And since KL is a non-symmetric measure, we can also observe the behavior of a learning tool in the opposite direction: the performance for A when using a feature set corresponding to B.

3.2 Cross Entropy (CE)

Cross entropy is also a measure to compare different probability distributions that bears close relation to the KL divergence (relative entropy). However, the purpose of the cross entropy between a random variable X with probability distribution p and another probability distribution q, denoted by $H(X, q)$, is to observe the role of q as model of the real distribution p.

$$H(X, q) = H(X) + D(p \parallel q) = -\sum_{x} p(x) \log q(x) \tag{2}$$

Our interest in this similarity measure consists in to observe a model q of the real distribution p. According to the expression 2, it is noticed that we want to minimize $D(p \parallel q)$: an optimum probabilistic model q is obtained as long as the divergence between p and q can be minimized. In this way, we are concerned in a learning tool for B trained on a model dataset A, in order to assess how accurate the model is in predicting B. Moreover, cross entropy is also a non-symmetric measure, so we can observe the behavior of a learning tool by measuring the cross entropy between A and B, and vice versa.

4 Experimental Setup and Results

We use for the experimentation conducted in this work a collection of Epinions reviews developed by Taboada et al. [15]. Such dataset consists of eight different categories: books, cars computers, cookware, hotels, movies, music and phones. There are 50 opinions per category, giving a total of 400 reviews in the collection, which contains a grand total of 279,761 words. And since within each category there are 25 reviews per polarity, the baseline accuracy for each domain is 50%.

The set of sentences corresponding to each review in the dataset used in our experimentation was submitted to a tagger based on a broad use of lexical features: The Stanford Tagger with a remarkable degree of accuracy [16]. We model each review as a feature vector. The granularity of the feature sets used in our experiments consisted of unigrams and bigrams. As the frequency of the ngram is required by the similarity measures, frequency features rather than binary features represent our content vector.

To be able to observe reliable regularities in the information provided by our divergence measures, we have only considered those domains with at least 4,000 features. Thus, we have discarded two domains: Cookware and Phones. These domains have been discarded because we can consider them as *outliers*: their number of features is clearly separated from the rest of the domains.

Once the data representation model (datasets and the content vector) has been defined, we estimate the subset of domains with a feature distribution similar to the target domain by making use of the similarity measures previously mentioned. Thus, from the feature sets corresponding to each domain we produce the matrix of the KL divergence (relative entropy) of the unigrams across domains shown in Table 1. It can be noticed how the entries on the main diagonal are zero, that is, when $p = q$, the KL divergence is 0.

Table 1. KL scores across domains

	Books	**Cars**	**Compu**	**Hotels**	**Movies**	**Music**
Books		0.64	0.54	0.56	0.42	0.70
Cars	0.42		0.46	0.50	0.50	0.67
Compu	0.24	0.49		0.44	0.46	0.59
Hotels	0.44	0.63	0.51		0.50	0.82
Movies	0.25	0.49	0.52	0.39		0.64
Music	0.21	0.51	0.42	0.42	0.40	

In the same way, we generate the cross entropy of the unigrams across domains shown in Table 2. In this case, the entries on the main diagonal represent the entropy of p, that is, when $p = q$, the cross entropy is H(p).

Table 2. CE scores across domains

	Books	**Cars**	**Compu**	**Hotels**	**Movies**	**Music**
Books		6.81	6.51	6.56	7.32	7.42
Cars	6.00		7.12	6.83	6.30	6.81
Compu	5.66	7.41		6.56	6.36	6.64
Hotels	6.17	7.24	6.70		6.56	6.96
Movies	6.67	6.64	6.60	6.35		7.50
Music	5.63	6.36	6.00	5.80	6.37	

5 Analysis and Discussion

Taking into account that the higher the cross entropy, the more similar the two domains, cross entropy is a guide as to how well a classifier trained on one domain will work when tested on another target domain.In the case of the KL divergence: the lower the relative entropy, the more similar the two domains, we can also make use of the KL divergence to estimate the performance of a classifier that has been trained on a foreign domain. For example, when the target domain is Books, Table 1 suggests the Movies domain as the best option to train a classifier. However, Table 2 proposes the Music domain as the best option.

Thus, once we obtained the similarity distributions for each target domain, we want to corroborate such distributions. In other words, we want to observe if, for example, the feature set of Movies represents a better option to classify Books reviews rather than the Music's features. In order to carry out this corroboration, we evaluate the performance for each target domain using a classifier based on the common features between the unseen reviews of the target domain and each of the foreign domains.

The method to evaluate the performance is based on 5-fold cross-validation and the use of support vector machines (SVM). As we know, SVM is a hyperplane classifier that has proved to be a useful approach to cope with natural text affairs [12]. Table 3 shows the SVM classifier accuracy for each target domain. Taking into account that the baseline accuracy for each domain is 50%, Table 3 exhibits how the use of the linguistic features corresponding to Music (77%) represent a better option to classify Books rather than the Movies' features (73%).

Table 3. SVM scores across domains

	Books	**Cars**	**Compu**	**Hotels**	**Movies**	**Music**
Books		73%	75%	69%	73%	77%
Cars	74%		82%	78%	76%	78%
Compu	80%	74%		78%	80%	78%
Hotels	76%	76%	72%		76%	74%
Movies	82%	76%	78%	84%		84%
Music	80%	74%	74%	68%	74%	

Additionally, we make use of the TP rate and FP rate values to show an alternative perspective of the performance estimation. Since the TP rate and FP rate values of different classifiers on the same test dataset are often represented diagrammatically by a ROC graph, Figure 1 shows the ROC graph corresponding to different classifiers tested on Books and trained on each of the foreign domains. As Figure 1 shows, the use of Music as model for Books has outperformed the rest of the domains. Therefore, the information provided by cross entropy has been more useful to identify a feature distribution most similar to the target domain.

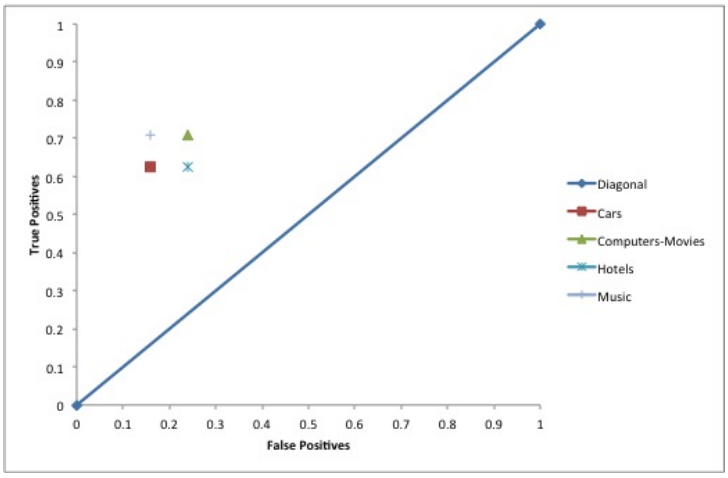


Fig. 1. ROC graph corresponding to Books

The use of non-symmetric measures is also exhibited by the obtained results. As we can see in Tables 1 and 2, the divergence value obtained between any two domains, A and B, is not the same as the one obtained between B and A. For example, both tables show how the divergence value obtained between Books and Movies is not the same discrepancy value obtained between Movies and Books. More specifically, the KL divergence in Table 1 shows how Books diverge less from Movies than the opposite case. On the other hand, the cross entropy in Table 2 shows how Movies is a more useful model to predict Books than Books to classify Movies. Taking into account the results shown in Table 3, the use of the linguistic features corresponding to Books represent a better option to classify Movies (82%) rather than the use of Movies to classify Books (73%).

Also, as an alternative perspective of the performance estimation, Figure 2 shows the ROC graph corresponding to different classifiers tested on Movies and trained on each of the foreign domains. Now, by comparing Figure 1 and Figure 2, we can see how the use of Books as model for Movies has outperformed the rest of the domains. Thus, the information provided in this case by relative entropy has been more useful to identify a feature distribution most similar to the target domain.

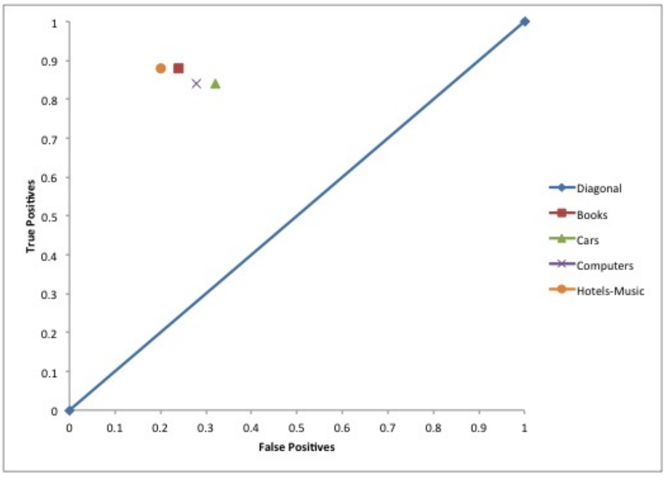


Fig. 2. ROC graph corresponding to Movies

By analyzing the information provided by our divergence measures (Tables 1 and 2), the most important point is to observe regularities that allow us to make reliable predictions when training a classifier for a domain for which no annotated data is available. According to the obtained results in our experimentation, we are able to observe the relationship between the performance (Table 3) and a divergence measure (Tables 1 and 2): in most of the cases, when the difference between the cross entropy values is close (less than one), KL is the best guide to predict the domain with a feature distribution similar to the target domain (i.e. see the divergence values between Books and Movies and vice versa). Otherwise, CE is an alternative to make such predictions (i.e. see the divergence values between Movies and Music and vice versa).

6 Conclusions and Future Work

In this paper, we focus our attention in the analysis of the distributions corresponding to different domains in order to determining the subset of domains with a feature distribution similar to the unlabeled target domain. By making use of non-symmetric divergence measures, we estimate the performance on the unlabeled target domain B of an opinion classifier trained on the domain A and vice versa: the performance on the unlabeled target domain A of an opinion classifier trained on the domain B. We find that quantifying the difference between domains is useful not only to predict which domain has a feature distribution most similar to the target domain but also to optimize the use of the available data.

As part of our future work, we intend to extend our distributional analysis by including measures that allow us to cope with feature-distribution vectors that are quite sparse. In particular, we intend to explore the implications of the use of the Jenson-Shannon divergence [9].

Also, we are interested in the analysis of more datasets collections. For example, the dataset collected by Blitzer et al. [3] is an interesting collection of product reviews from four domains: books, DVDs, electronics, and kitchen appliances. We think this collection is worth our attention to improve and optimize our analysis.

References

1. Asch, V., Daelemans, W.: Using domain similarity for performance estimation. In: Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing. pp. 31–36 (2010)
2. Aue, A., Gamon, M.: Customizing sentiment classifiers to new domains: a case study. In: Proceedings of Recent Advances in Natural Language Processing (RANLP) (2005)
3. Blitzer, J., Dredze, M., Pereira, F.: Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL) (2007)
4. Brazdil, P., Giraud-Carrier, C., Soares, C., Vilalta, R.: Metalearning Applications to Data Mining. Springer-Verlag (2009)
5. Cardie, C.: Empirical methods in information extraction. AI Magazine 39(1), 65–79 (1997)
6. Feldman, R.: Techniques and applications for sentiment analysis. Communications of the ACM 56(4), 82–89 (2013)

7. Glickman, O., Jones, R.: Examining machine learning for adaptable end-to-end information extraction systems. In: AAAI 1999 Workshop on Machine Learning for Information Extraction (1999)
8. Kullback, S., Leibler, R.A.: On information and sufficiency. Annals of Mathematical Statistics 22, 79–86 (1951)
9. Lee, L.: Measures of distributional similarity. pp. 25–32 (1999)
10. Liu, B.: Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data. Springer-Verlag (2007)
11. Mansour, Y., Mohri, M., Rostamizadeh, A.: Multiple source adaptation and the rényi divergence. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence. pp. 367–374 (2009)
12. Rényi, A.: On measures of information and entropy. In: Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics and Probability. vol. 1, pp. 547–561 (1961)
13. Sekine, S.: The domain independence of parsing. In: Proceedings of the 5th Conference on Applied Natural Language Processing (1997)
14. Shou-Shan, L., Chu-Ren, H., Cheng-Qing, Z.: Multi-domain sentiment classification with classifier combination. Journal of Computer Science and Technology 26(1), 25–33 (2011)
15. Taboada, M., Anthony, C., Voll, K.: Creating semantic orientation dictionaries. In: Proceedings of Fifth International Conference on Language Resources and Evaluation (LREC). pp. 427–432 (2006)
16. Toutanova, K.and Klein, D., Manning, C., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL). pp. 252–259 (2003)
17. Vila, K., Ferrández, A.: Model-driven restricted-domain adaptation of question answering systems for business intelligence. In: Proceedings of the 2nd International Workshop on Business Intelligence and the WEB. pp. 36–43 (2011)

Intelligent Learning Environments

Guest editors:
Ramón Zatarain,
María Lucía Barrón,
Yasmín Hernández

Learning Styles and Emotion Recognition in a Fuzzy Expert System

Ramón Zatarain-Cabada, M. Lucía Barrón-Estrada, Rosalío Zatarain-Cabada

Instituto Tecnológico de Culiacán, Juan de Dios Bátiz s/n, Col. Guadalupe, Culiacán Sinaloa, 80220, Mexico

{rzatarain, lbarron}@itculiacan.edu.mx

Abstract. This paper presents a fuzzy system that recognizes learning styles and emotions using two different neural networks. The first neural network (a Kohonen neural network) recognizes the student cognitive style. The second neural network (a back-propagation neural network) was used to recognize the student emotion. Both neural networks are being part of a fuzzy system used into an intelligent tutoring system. The fuzzy system evaluates both cognitive and affective states in the student whenever he/she answers math exercises.

Keywords: Intelligent tutoring systems, affective computing, learning technologies, artificial neural networks, education.

1 Introduction

In last years, Intelligent Tutoring Systems (ITS) have integrated the ability to recognize the student's affective state, in addition to traditional cognitive state recognition. Research on affective computing includes detecting and responding to affect. Affect detection systems identify frustration, interest, boredom, and other emotions [1, 2,]. On the other hand, affect response systems transform negative emotional states (frustration, boredom, fear, etc.) to positive ones [3, 4]. Ekman's work on face analysis [5] describes a subset of emotions including joy, anger, surprise, fear, disgust/contempt and interest, which have been used in new ITS, which include the recognition and treatment of emotions and/or feelings [1, 3, 6, 7].

In this work, we present a system that combines affective computing and learning styles into a fuzzy system which is part of a ITS. We have integrated two methods for selecting the learning style and emotional state of a student and to consider them in the ITS response. For recognizing the learning style and the affective state, we implemented two neural networks. During a training session, the first network (a SOM or Kohonen network) used for detecting the learning styles, receives a number of different input patterns (the student learning style obtained from an Inventory Learning Style Questionnaire (ILSQ), the learning style of three defined courses, and the student's grade in each course), discovers significant features in these patterns and learns how to classify the input patterns. The second network (a back-propagation neural network), which is used to detect affective or emotional states, is trained with a corpus of faces representing different emotional states. The affective and learning style recognizers are used into a fuzzy system, which is part of an ITS.

2 Recognizing Learning Styles and Emotional States

By using the neural networks, the learning style and the emotional state of a student can be dynamically calculated according to evaluations and face recognition applied to the student while he/she is using the ITS.

A course can be seen as a discipline-specific knowledge space (a particular tree diagram) containing chapters, which in turn are made by subjects. The total of nodes in the tree represents the domain or expert knowledge. Figure 1 shows the knowledge domain of the math course with subjects related to arithmetic operations such as multiplication and division and topics like fractions. For each topic, different learning instances teaching the same theme under different learning styles can be observed. In this way, many combinations or paths are provided for learning the same topic. Dashed lines represent those paths. The domain module is stored as a XML-based document.

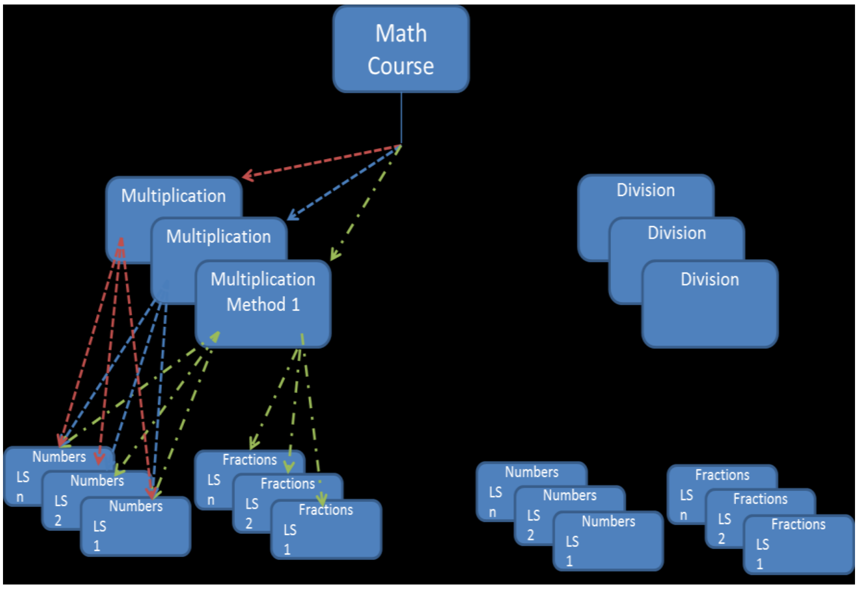


Fig. 1. Knowledge domain of a math course under different learning styles (LS)

2.1 The Kohonen Neural Network for Recognizing Emotional States

The method used for the detection of visual emotions is based on Ekman's theory (Ekman and Friesen, 1975). The recognition system was built in three stages: the first one was an implementation to extract features from face images in a corpus used to train the neural network. The second one consisted of the implementation of the neural network. The third stage integrated extraction and recognition into the fuzzy system. For training and using the neural network we used the corpus RAFD (Radboud Faces Database) [8] which is a database with 8040 different facial expressions, which contains a set of 67 models including men and women. Once the emotion state is extracted from the student the state is sent to the fuzzy system (see figure 2).

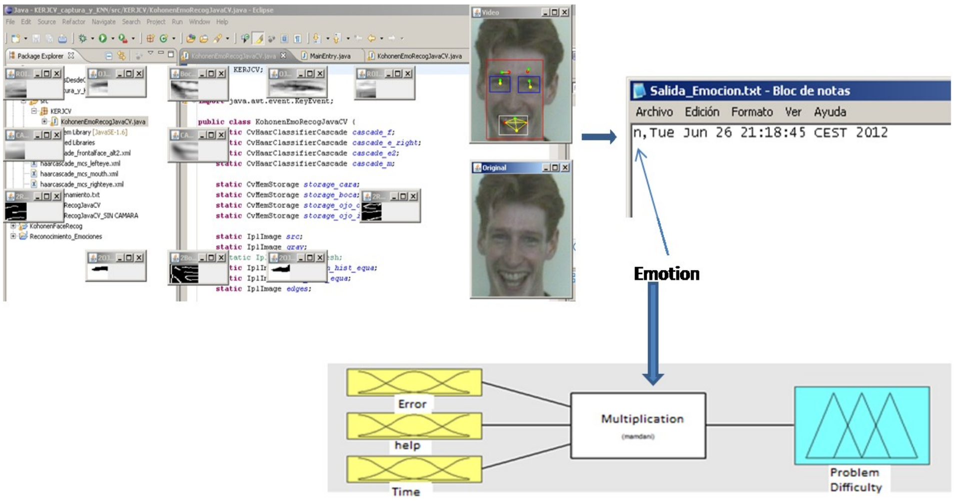


Fig. 2. Emotion extraction in the student

2.2 A Kohonen Neural Network for Learning Styles

The input layer of the neural network for learning styles has 7 neurons. The Kohonen layer has 1600 neurons, organized in a lattice of hexagonal cells with dimensions of 40x40 neurons. The signals are part of the training data space and they are vectors composed of three elements: two vectors and a scalar. The first vector is the student's learning style identified by using the ILSQ questionnaire. The second vector is the learning style of the learning material read by the student (three courses about Computer Science, Photography, and Eolithic Energy). The last element is the student’s performance. The neural network provides the student's learning style as an output value. The number of iterations used for training the neural network was 5000 iterations. Figure 3 shows part of the training of the Kohonen or SOM neural network.

3 The Fuzzy Expert System

The student module provides information about student knowledge and learning aptitudes. The module identifies what the student's knowledge is through a diagnostic test. The student knowledge can be seen as a subset (sub-tree implemented) of all knowledge possessed by the expert in the domain (module) and this is stored in a student profile.

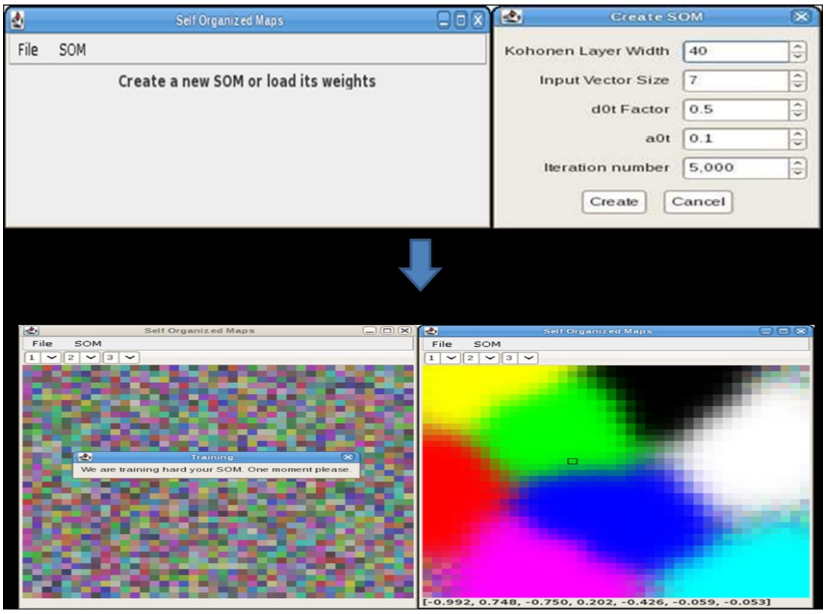


Fig. 3. Training of the Kohonen network for recognizing learning styles

For every student there are a static and a dynamic profile, which store particular and academic information like affective states, learning styles, and scoring results. In the ITS, a fuzzy expert system was implemented with a new knowledge tracing algorithm, which is used to track student's pedagogical states, applying a set of rules. The benefit of using fuzzy rules is that they allow inferences even when the conditions are only partially satisfied. The fuzzy system uses input linguistic variables such as *error, help, time*, *Emotion*, and *Learning Style* (Figure 4). These variables are loaded when the student solves an exercise. The output variable of the fuzzy system is the difficulty and type of the next exercise. The type is defined according the learning style assigned to the student.

3.1 The Fuzzy Sets

The proposed fuzzy sets, for each linguistic variable are:

- Error = {low, normal, many}
- Help = {little, normal, helpful}
- Time = {very fast, fast, slow, very slow}
- Emotion = { anger, disgust, fear, happiness, sadness, surprise, and neutral }
- Learning Style = { Visual, Verbal, Sequential, Global, Sensitive, Intuitive }
- Difficulty = {very easy, easy, basic, hard, very hard}

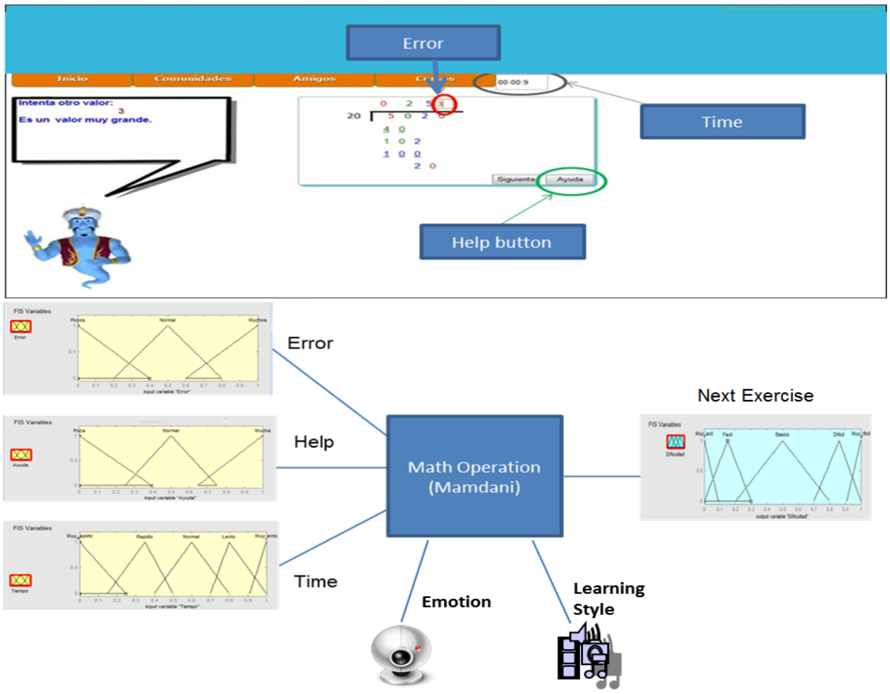


Fig. 4. Input and Output of fuzzy variables

3.2 Rule Evaluation

One important step of our fuzzy expert system is to evaluate the fuzzy values of the input variables. Table 1 shows a sample of some of the fuzzy rules that are used in the system.

Table 1. A sample of fuzzy rules of the expert system

Rule #	Rule
Rule 1	If(**Error** is low) and (**Help** is **little**) and (**Time** is **very fast**) and (Emotion is **neutral**) then (**Difficulty** is **very_hard**)
Rule 2	If(**Error** is low) and (**Help** is **little**) and (**Time** is **fast**) and (Emotion is **neutral**) then (**Difficulty** is **very_hard**)
Rule 3	If(**Error** is low) and (**Help** is **little**) and (**Time** is **normal**) then (**Difficulty** is **very_hard**)
Rule 4	If(**Error** is low) and (**Help** is **little**) and (**Time** is **slow**) then (**Difficulty** is **hard**)
Rule 5	If(**Error** is low) and (**Help** is **little**) and (**Time** is **very-slow**) then (**Difficulty** is **hard**)
Rule 6	If(**Error** is low) and (**Help** is **normal**) and (**Time** is **slow**) and (Emotion is **sadness**) then (**Difficulty** is **basic**)
…	…
Rule 41	If(**Error** is **many**) and (**Help** is **helpful**) and (**Time** is **very-slow**) and (Emotion is **fear**) then (**Difficulty** is **very_easy**)

In order to evaluate the conjunction of the rule antecedent, we applied the following equation:

$$\mu_{A\cap B\cap C\ldots\cap Z}(x) = \min[\mu_A(x), \mu_B(x), \mu_c(x), \ldots, \mu_Z(x)] \quad (1)$$

To evaluate disjunction, we applied equation:

$$\mu_{A\cup B\cup C\ldots\cup Z}(x) = \max\,[\mu_A(x), \mu_B(x), \mu_c(x), \ldots, \mu_Z(x)] \quad (2)$$

For instance, to evaluate the next fuzzy rule:

IF Error is low (0.3)
AND Help is little (0.2)
AND Time is very-fast (0.1)
AND Emotion is neutral (0.2)
THEN Difficulty is very-hard (0.1)

Equation 1 is applied:

$\mu_{\text{very-hard}}(\mathit{Difficulty}) = \min[\mu_{\text{low}}(\mathit{Error}), \mu_{\text{little}}(\mathit{Help}), \mu_{\text{very-fast}}(\mathit{time}),$
$\mu_{\text{neutral}}(\mathit{emotion})] = \min[0.3,0.2,0.1,0.1] = 0.1$

3.3 The Implementation of an Exercise

Next, we present the structure of the XML-based file of six exercises about integer divisions and how the student solves the exercises with the help and support of the intelligent tutoring system.


```
Division ([
    {"divisor":9,"dividend":[1,0,8],"quotient":[0,1,2],"reminder":[1,0],"mul":[9,18]},
    {"divisor":2,"dividend": [4,2],"quotient":[2,1],"reminder":[0,0],"mul":[4,2]},
    {"divisor":11,"dividend":[1,0,0],"quotient":[0,0,9],"reminder":[1],"mul":[99]},
    {"divisor":10,"dividend":[5,0,0],"quotient":[0,5,0],"reminder":[0,0],"mul":[50,0]},
    {"divisor":20,"dividend":[5,0,2,0],"quotient":[0,2,5,1],"reminder":[10,2,0],"mul":[40,100,20]},
    {"divisor":14,"dividend":[1,3,2],"quotient":[0,0,9],"reminder":[6],"mul":[126]}
    ]);
```


The basic structure of the XML-based file consists of an array of objects which contains the "divisor" and "dividend" attributes which are shown to the student. The "quotient", "reminder" and "mul" attributes contain the correct answers.

An initial exercise is presented to the student through the interface; students can enter answers they think are correct, while the intelligent tutor dynamically check the corresponding XML file to verify the answer and to provide responses to them. The initial exercise has a difficulty level that was set for each student profile according the result in the diagnostic test completed by the own student. The difficulty level of the next exercises can be modified depending on the student's performance in solving each math exercise. The functionality of how responses are evaluated and the path considered by the solution process are shown in Figure 5. In this context, the premise is simple. The ITS waits for an entry value *t*, and verifies that the value is correct. When a correct value is entered, the ITS moves to the next box; then it will wait for the next input value. Otherwise, the ITS sends a message through a pedagogical agent about the type of error found in the answer and then it waits for a student response. This action is repeated until the division is complete. During this process the student can make use of two buttons located below the division operation. The "Help" button sends tips or advices to the student through the pedagogical agent. The "Next" button moves to the next exercise.

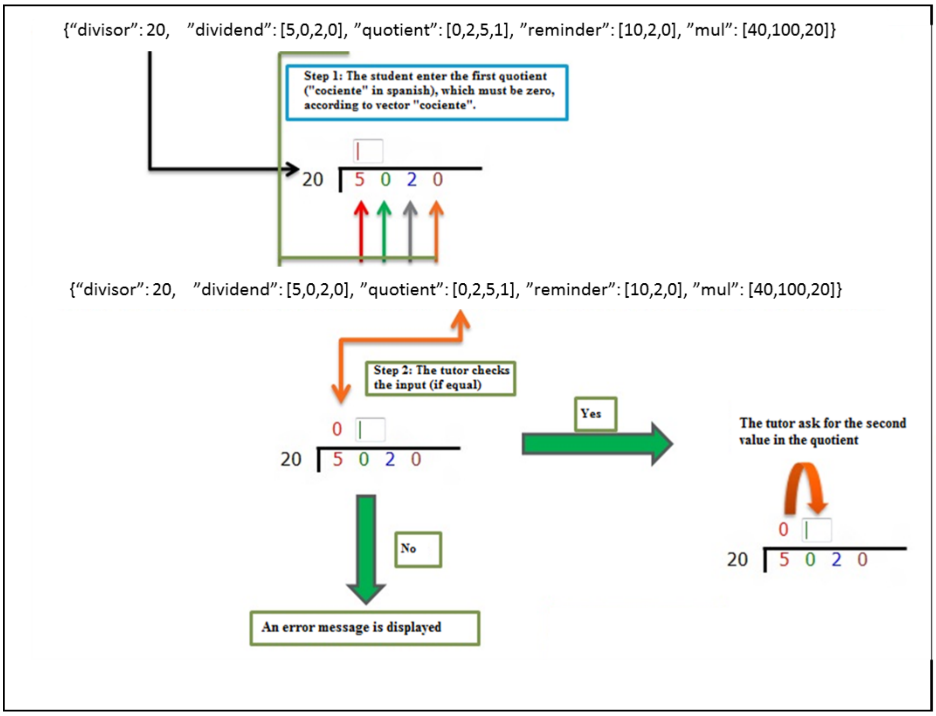


Fig. 5. Evaluation of an example of the Division Arithmetic Operation

4 Results and Conclusions

We still are working with the integration of the neural networks to the ITS. We have evaluated the ITS (with no emotion and learning style recognition) with a group of children from third grade. There were 72 children from public and private schools who tested the tool and the tutoring system. We evaluated the subject of multiplication. We applied a test before and after the students used the software tool. We obtained from the results a good improvement in most students (more in students with lower initial grades) using one of the two teaching methods for multiplication: traditional and lattice.

The results up to now are encouraging. The next step is to integrate the neural networks with the ITS. We also are integrating as an application, the affective ITS into the Facebook social network. The intelligent tutoring system and recognizers were implemented by using different software tools and programming languages.

References

1. Arroyo, I., Woolf, B., Cooper, D., Burleson, W., Muldner, K., Christopherson, R.: Emotions sensors go to school. In: Proceedings of the 14th international conference on artificial intelligence in education. pp. 17−24. IOS press, Amsterdam (2009)
2. Conati C., Maclaren, H.: Empirically Building and Evaluating a Probabilistic Model of User Affect. User Modeling and User-Adapted Interaction, 19, 267−303 (2009)
3. 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)
4. Du Boulay, B.: Towards a motivationally intelligence pedagogy: how should an intelligent tutor respond to the unmotivated or the demotivated? In: R. Calvo & S. D'Mello (Eds.) New perspectives on affect and learning technologies, New York: Springer, pp. 41−52 (2011)
5. Ekman, P., Friesen, W.: Unmasking the face: a guide to recognizing emotions from facial clues. Englewood Cliffs, NJ: Prentice-Hall (1975)
6. Forbes-Riley, K. & Litman, D.: Adapting to Student Uncertainty Improves Tutoring Dialogues. In: Proceedings of the 14th International Conference on Artificial Intelligence in Education (AIED), pp. 33−40, Brighton, UK (2009)
7. D'Mello, S.K., Graesser, A.C.: AutoTutor and Affective AutoTutor: Learning by Talking with Cognitively and Emotionally Intelligent Computers that Talk Back. ACM Transactions on Interactive Intelligent Systems, 2(4), 23:2-23:39 (2012)
8. Langner, O., Dotsch, R., Bijlstra, G., Wigboldus, D., Hawk, S., van Knippenberg, A.: Presentation and validation of the Radboud Faces Database. Cognition & Emotion, 24(8), 1377−1388, DOI: 10.1080/02699930903485076 (2010)

Overview of UI Patterns on Mobile Platform for Educational Applications

Raquel Vázquez-Ramírez, Humberto Marín-Vega, Giner Alor-Hernández

Division of Research and Postgraduate Studies, Instituto Tecnológico de Orizaba, Mexico

vz.rmz.raquel@gmail.com, hmarin@acm.org, galor@itorizaba.edu.mx

Abstract. Software development is currently experiencing significant growth, and the need to improve user interfaces is an important factor. Given this situation, interface design patterns have emerged as a new means of improving the user experience. However, their use has not been adequately studied for use in specific domains such as education. For this reason, it is necessary to analyze design patterns for mobile applications in order to determine the best design patterns for improving the quality and usefulness of applications, more specifically in educational environments. Having this into account, this paper presents an analysis and classification of mobile UI patterns in order to understand their applicability for the educational domain. We present a comparative table of mobile UI patterns implemented on several platforms.

Keywords: User interface, mobile platforms, educational applications.

1 Introduction

New technologies and programming styles are currently promoting the development of adaptive user interfaces for different types of software systems. However, if the concepts established by the Human-Computer Interaction (HCI), which are based on the concepts of utility and usability, are not taken into account, these interfaces do not ensure user satisfaction. HCI is intended to increase the productivity and safety of both the system and the user [1].

User interfaces are a crucial factor in ensuring the success of an application [2], [3], not only in a commercial and promotional point of view, but also in terms of acceptance by the end user, as well as ensuring the software achieves its objective of meeting a need or solving a problem. In educational applications, the aforementioned concepts represent an exceedingly important means of ensuring better teaching-learning processes through the use of technology. In this sense, HCI allows the successful development of graphic user interfaces. Sometimes the applications are developed for specific users, and in this case it is necessary to determine the type of interface to develop.

Mobile applications have grown considerably and have been faced with the challenge of adapting the user interface design for different devices such as iPads, Microsoft's Surface Tablet and Android tablets, to mention but a few, taking into account the mobile device constraints such as screen size, keyboard type, screen resolution, processing power, storage space and communication capabilities. Mobile devices offer higher independence in terms of location and time when compared to web-based education processes accessed via computers. In this sense, there is a necessity to define interface standards for these kinds of applications. However, interface design patterns for mobile devices have not yet been sufficiently studied in terms of their use for specific domains such as education. For this reason, it is necessary to analyze existing mobile design patterns in order to get more out of mobile devices.

This paper is organized as follows: Section 2 presents recent advances in state-of-the-art mobile UI design patterns. Section 3 describes the analysis of the interface design patterns for mobile devices. Section 4 presents the mobile platforms that were selected for this work. Section 5 presents a case study that implements the design patterns identified on mobile platforms, as well as shows the results of the evaluation using frameworks multi-device and the discussion of the evaluation results. Section 6 describes the future directions to be taken. Finally, the conclusions are presented in Section 7.

2 State of the art

In recent years, several studies have been proposed with the aim of improving the development of user interfaces. Most of these have been focused on the use of UI patterns in a variety of contexts. However, the use of design patterns in educational contexts has not yet been reported. In this section, we present a set of related works focused on the use of UI design patterns. These works have been grouped according to the kind of application to be developed: mobile and cross-platform applications.

2.1 UI Patterns for Mobile Applications

The UI design patterns for mobile applications have been recently studied due to the increase in the development of applications for mobile devices such as smartphones and tablets. Unlike UI patters for Web and desktop applications, the UI design patterns for mobile applications overcomes restrictions such as screen size, mobile device type and processing power [4].

Raj & Komaragiri [5] presented an analysis to identify UI design patterns in mobile devices, specifically interaction patterns. This analysis shows that a prototyping tool is useful to solve the constraints of usability and consistency, as well as reducing the time taken to develop a mobile application.

Nilsson [6] presented a structured collection of user interface design patterns for mobile applications. This paper described the use of these patterns to solve six identified problems in mobile application development: 1) screen space in general, 2) flexible user interfaces, 3) handling input, 4) not using the stylus, 5) guidelines and 6) difficult to understand. The collection of user interface design patterns has different levels of abstraction and shows how patterns may be used to present problems and solutions in different levels of detail.

Serhani, Benharref, Dssouli & Mizouni [7] discussed the main issues involved in developing mobile applications. A framework that helps developers to build efficient, high quality and secure mobile applications was proposed. This framework was used to develop the Eivom Cinema guide. This system is a mobile application that integrates data from different cinemas information systems through Web services technology to provide users with a variety of services. Viana & Andrade [8] presented a number of challenges, among them the need for developing multi-device interfaces in different contexts. This research presented an AMB-UID (model-based UI development) environment called *XMobile* for the automatic generation of user interfaces for mobile devices. The main goal of *XMobile* was to cut down on the prototyping time of an application and to provide various adaptation levels of the user interface.

2.2 UI Patterns for Cross-platform

Several approaches are oriented to study UI patterns that can be applied in the development of desktop, Web and mobile applications.

Korozi, Leonidis, Margetis & Stephanidis [9] presented a new design framework called *MAID* that helps designers to easily create user interfaces. This design framework is complemented by a widgets library that could be used in different application panels or in entirely different applications. The *MAID* tool development process could be decomposed in the following four phases: 1) UI Definition, 2) Application Data Integration, 3) UI Adaptation and 4) Deployment.

Seffah, Forbrig & Javahery [10] highlighted the problems in developing user interfaces for multiple devices such as computers, laptops or mobile telephones. The authors presented an investigation about the Multiple User Interface (MUI) and the most important problems surrounding MUI development models. Finally, a set of HCI patterns was presented and the types of cross-platforms that are recommended for each pattern were discussed.

Märtin, Engel, Kaelber & Werner [11] explained that user experience and usability aspects have never been the main focus when developing knowledge management systems. This research described a pattern-based approach for designing highly-usable individualized multi-media interfaces for enterprise knowledge identification, structuring and communication.

Tidwell [12] presented a set of patterns according to different facets of UI design; categories include Content Organization, Navigation, Page Layout, and Actions/Commands. These patterns have been used to develop more effective UI for desktop applications, websites, web applications and mobile devices.

3 Mobile UI Design Patterns for Educational Applications

A mobile operating system, also referred as mobile OS, can operate on Smartphones, tablets or other digital mobile devices. Modern mobile OS combine the features of a personal computer Operating System with other features [13]. Currently, the most popular mobile OS are Android, iOS, BlackBerry and Windows Phone.

– Android is a Linux-based Operating System developed by Google. Android is open source and Google releases the code under the Apache License [14].
– iOS is a mobile OS developed and distributed by Apple Inc. It is closed source and proprietary and built on open source Darwin core OS, which is derived from Mac OS X [15].
– BlackBerry OS is a proprietary mobile Operating System developed by BlackBerry Ltd for its BlackBerry line of smartphones handheld devices [16].
– Windows Phone is a series of proprietary smartphones Operating Systems developed by Microsoft. Windows Phone is based on the Windows NT kernel [17].

In recent years, mobile applications have successfully supported a lot of processes but sometimes the applications are developed for specific mobile OS, and in this case it is necessary to determine the type of interface to develop for several mobile OS. In this case, interface design patterns play an important role in providing general solutions that have been tested and have shown their effectiveness in solving recurring problems [18].

Table 1. Mobile UI design patterns for educational applications

Pattern		**UI Pattern**
Content Navigation	(1) List Menu	Menu of topics
	(2) Tabs	Switch between content types
	(3) Page Carousel	Navigation between content
	(4) Expanding List	Menu of topics
	(5) Bottom Navigation	Navigation control between topics
	(6) Thumbnail and Text List	Topic with icon thumbnail
	(7) Springboard	Menu for content selection (Type of content in a topic)
	(8) Header less Table	List questionnaires
	(9) Grouped Rows	Menu of topics with sections
Search, Sort and Filter	(10) Dynamic Search	Topic search
	(11) Scoped Search	Topic search by content type
Help, Hints and Cheats	(12) Tip	Contextual help and tool tips
	(13) Walkthroughs	Help for UI Interaction
Confirm Interactions	(14) Dialog	Confirm dialog for user actions
Display Specialized Content	(15) Video	Show videos and animations
Task Progress	(16) Loading indicators	Show progress while content is loading
Group Activities	(17) Tour	Group questionnaires

The UI Patterns considered for this work is the proposal of [19] Mobile UI Patterns for Educational Applications. Table 1 present the results of the analysis of existing patterns, highlighting the following information: 1) patterns that are adapted to the educational context; 2) the pattern type, based on the classifications proposed by various authors; 3) the name of the design pattern that best fits the educational context; 4) its use in an educational context; and 5) an image of an educational application that implements this pattern. For the elaboration of this table, we have used a system called *Athena* that enables the generation of educational applications for multiple mobile devices [13].

Table 1 show some of the most representative patterns presented in [13]. These patterns have been selected based on the functionality offered in deploying educational content such as topics, images, text, animation and videos, to name but a few.

4 UI Design Patterns Support on Mobile Platforms

In order to validate the aforementioned analysis, we have selected five frameworks for RIAs multi-device. A multi-device framework is a software framework that is designed to support and facilitate the development and maintenance of mobile applications. The multi-device frameworks that were selected for these analyses are PhoneGap, Rhomobile Rhodes, MoSync, IUI and Marmalade. Evaluation was a mathematics course based on mobile UI patterns as educational software to create multi-device native applications for Android, iOS, BlackBerry and Windows Phone.

Let us suppose that we need to develop educational software for mobile devices; in this case mathematics course for grade school. The application to be developed should be visually attractive and easy and intuitive to use, in such a way that the user feels comfortable using it. It should also support the teaching-learning process through the use of educational resources. To achieve this we have two options: 1) using an HTML5-based Web application optimized for mobile devices; or, 2) developing a native application based on a mobile operating system. In order to take advantage of hardware features, it is feasible to develop native applications. The evaluation process of UI design patterns is not an easy task. The following shows the set of UI patterns available for each mobile platform.

At the end of the evaluation process, we have determined that importance of the design patterns for the educational context and the availability of each UI pattern on a framework multi-device according to the mobile platform that were be selected. Nonetheless, it is possible that, according to the type of educational application and the academic level to which this educational software is aimed, it may require some other pattern that may have been omitted from this evaluation. Table 2 shows the available UI patterns and mobile OS of each framework multi-device. The notation on the table for mobile platforms is as follows: "A" is used for Android, "BB" is used for BlackBerry and "WP" for Windows Phone. The notation on the table for the UI Patterns is 1 when UI pattern is supported on the mobile OS by using multi-device framework and 0 when the UI pattern is not supported on mobile OS.

Table 2. UI patterns support on mobile platforms using a multi-device framework

Framework	Platform	UI Pattern																
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Phone Gap 2.5	A 4.0.3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
	iOS 4.0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
	BB 7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
	WP 7.5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Rho-mobile Rhodes 3.3.3	A 2.2	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	0	1
	iOS 4.0	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	0	1
	BB 6	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	0	1
	WP 7.5	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	0	1
MoSync 3.1	A 2.3.3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
	iOS 4.0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
	BB 6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
	WP 7.5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
IUI	A 2.3.4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	iOS 4.0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	BB 6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	WP 7.5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Marmalade	A 2.3.3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	iOS 4.0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	BB 6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	WP 7.5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

5 Future Directions

We are considering identifying and adapting design patterns for the development of applications for digital TV.

We also considered more frameworks multi-device for the analysis and UI Patterns.

On the other hand, the results of the analysis of design patterns provide the opportunity to observe their use in specific contexts such as education, so it would be interesting to explore other contexts and applications of design patterns in order to clarify and obtain a more detailed classification in order to facilitate and promote their use.

6 Conclusions

This paper has presented an evaluation of UI design patterns for the development of educational software for mobile devices. This paper provides a guide for anyone involved in the development of educational applications for mobile devices that must include patterns according to their relevance and contribution in terms of usability.

We have also presented the basis for evaluating and classifying design patterns in a specific context, in order to provide guidance for software developers in choosing the appropriate design patterns for the context in which they are developing.

Acknowledgments. This work was supported by the General Council of Superior Technological Education of Mexico (DGEST). Additionally, this work was sponsored by the National Council of Science and Technology (CONACYT) and the Public Education Secretary (SEP) through PROMEP.

References

1. Helader, M., Thomas, L., Prabhu, P.: Handbook of Human-Computer Interaction. Amsterdam: North-Holland: Elsevier Science Publishers (1977)
2. Wasserman, A.: Software Engineering Issues for Mobile Application Development. Proceedings of the FSE/SDP workshop on Future of software engineering research, Santa Fe, New Mexico, USA, pp. 397–400 (2010)
3. Charland, A., Leroux, B.: Mobile application development: web vs. native. (ACM, Ed.) Communications. ACM, 54(5) 49-53 (2011)
4. Lee, Y. E., Benbasat, I.: Interface design for mobile commerce. Commun. ACM, 46(12) 48–52 (2003)
5. Raj, A., Komaragiri, V.: RUCID : Rapid Usable Consistent Interaction Design Patterns-Based Mobile Phone UI Design Library. Human Computer Interaction. New Trends, 5610, pp. 677–686 (2009)
6. Nilsson, E. G.: Design patterns for user interface for mobile applications. Advances in Engineering Software, 14(2) 1318–1328 (2009)
7. Serhani, M., Benharref, A., Dssouli, R., Mizouni, R.: Toward an Efficient Framework for Designing, Developing, and Using Secure Mobile. International Journal of Human and Social Sciences (IJHSS), 5(4) 272–278 (2010)
8. Viana, W., Andrade, R. M. XMobile: A MB-UID environment for semi-automatic generation of adaptive applications for mobile devices. Journal of Systems and Software, 81(3) 382–394 (2008)
9. Korozi, M., Leonidis, S., Margetis, G., & Stephanidis, C.: MAID: a Multi-platform Accessible Interface Design Framework. Universal Access in Human-Computer Interaction. Applications and Services, 5616, pp. 725–734 (2009)
10. Seffah, A., Forbrig, P., Javahery, H.: Multi-devices "Multiple" user interfaces: development models and research opportunities. Journal of Systems and Software, 73(2) 287–300 (2004)
11. Märtin, C., Engel, J., Kaelber, C., Werner, I.: Using HCI-Patterns for Modeling and Design of Knowledge Sharing Systems. Perspectives in Business Informatics Research, 64, pp. 1–13 (2010)
12. Tidwell, J.: Designing Interfaces (Second ed.). O'Reilly Media. California (2011)
13. Arroyo, N.: Información en el móvil. Barcelona: UOC (2011)
14. Gargenta, M.: Learning Android. O'Reilly. Ucrania, pp. 1–2 (2011)
15. Lee, W.: Beginning iOS 4 Application Development. Wiley Publishing. Indianapolis (2010)
16. King, C.: Advanced BlackBerry Development. Apress. New York (2009)
17. Thurrott P.: Windows Phone 7 Secrets. John Wiley & Sons (2010)

18. ZadahmadJafarlou, M., Arasteh, B., & YousefzadehFard, P.: A pattern-oriented and web-based architecture to support mobile learning software development. Procedia-Social and Behavioral Science Journal, 28, pp.194–199 (2011)
19. Alor-Hernández, G., Vasquez-Ramirez, R., Rodriguez-Gonzalez, A.: Athena: a hybrid management system for multi-device educational content. Computer Applications in Engineering Education (2012)

Results of a Case Study of an Intelligent Tutoring System for Analyzing Student Projects Presented as Research Papers

Jesús Miguel García Gorrostieta[1], Samuel González López[2], Aurelio López-López[2]

[1] Universidad de la Sierra, Moctezuma, Sonora, Mexico
[2] Instituto Nacional de Astrofísica, Optica y Electronica, Mexico

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

Abstract. At "Universidad de la Sierra", projects of the end of a course are conducted. These projects are requested to have the structure of a research paper. In order to improve the quality of the assignment and guide the students in developing such work, an intelligent tutoring system was used. In this paper, we show 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 present the empirical evaluation results indicate that students found this tool useful and improve their writing.

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

1 Introduction

Performing well structured research work is a complex process which requires the support of the teacher. For this task technologies such as tutoring systems can be the solution. An intelligent tutoring system (ITS) is a system that provides personalized instruction or feedback to students without much involvement of instructors. Advances in ITS includes the use of natural language technologies to perform automated writing evaluation and provide feedback as presented in the article by Crossley [1]. Writing Pal (WPal) is an ITS that offers strategy instruction and game-based practice in the writing process 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.
At “Universidad de la Sierra”, projects of end of course are conducted, this projects are requested to have the structure of a research paper, in order to improve the quality of the assignment and guide the students in developing such work, an intelligent tutoring system was used to assist students. 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 certain methods to evaluate the use of vocabulary in a document. One of them is to measure the sophistication using a list of 3000 easy words in spanish [5]. For Spanish, some studies use the list provided by the SRA (Spanish Royal Academy) of 1000, 5000 and 15000 most frequent words. Others works have used Yule's K to measure the richness in texts [6], where this kind of measures focuses on the word repetitions and it’s considered a measure of lexical variety.

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. We also apply an empirical evaluation with students to verify the effectiveness of the proposed system and present its results.

2 Writing Evaluation Model

The intelligent tutor presents material concerning the different elements of the project, such as the problem statement, hypothesis, objectives and justification in the Domain Module. 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.

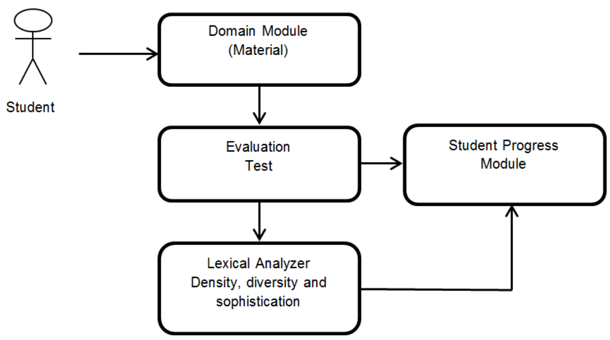


Fig. 1. Intelligent Tutoring System

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].

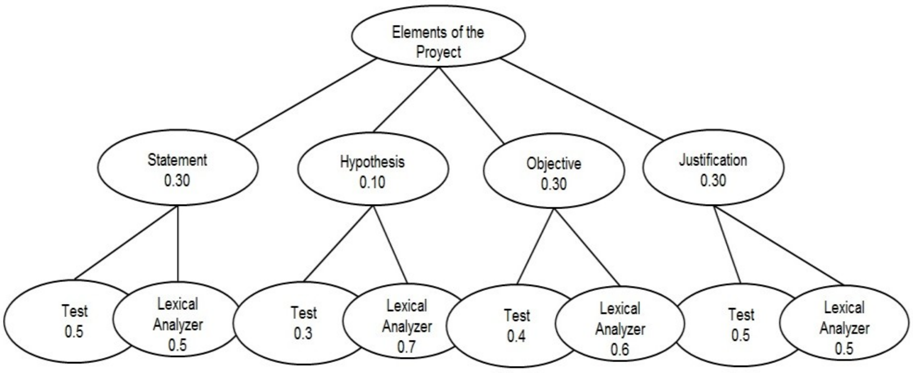


Fig. 2. Network used in Student Progress Module

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.

Figure 2 illustrates the weights assigned to each node according to the experience of the teacher. For instance, in the Test node of the Objective, a weight of 40% of the parent node Objective is assigned, which includes 5 questions to verify that the student has read the material. Once the student has correctly answered questions, this will enable him to use the lexical analyzer to perform three exercises which have a combined weight of 60% of the parent node, which is distributed as follows: 20% to lexical density, 20% to lexical diversity, and finally 20% for lexical sophistication.

In Figure 3 we show the model of lexical Analyzer, the 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.

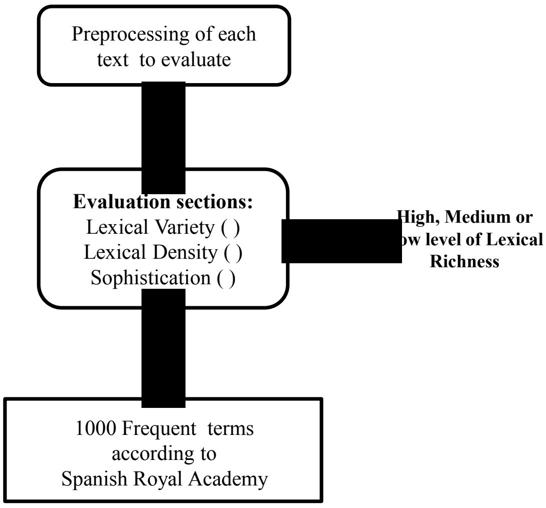


Fig. 3. Model of Lexical Analyzer

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 thesis of graduate and undergraduate students.

3 The Intelligent Tutoring System

The system was developed in PHP and MySQL with XAMPP package to have a web access, the lexical analyzer is developed in Python because of the ease access to processing tools of natural language. The analyzer uses the open source tool FreeLing[1] 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, hypothesis, 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

[1] This software is available at http://nlp.lsi.upc.edu/.

concept with 70% and 21% of the complete course. As we can observe, to enter the practical evaluation, the student must first successfully complete a test of basic knowledge of the concept.

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

The section of practical evaluation is also depicted in figure 4, where the student writes his problem statement to be analyzed, first density analysis measures the balance between content words and stop words; if the text has too many stop words it will have a very low density. Then 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*) as we can see in figure 4. 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.

At the end 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 Results of the Empirical Evaluation

An empirical evaluation was applied to verify the effectiveness and acceptance of the system at ABC University in a Computer Science Career. Two groups were formed of 14 students, both groups were requested to do the same product, write the preliminary draft of their final project which consisted in the statement of the problem, hypothesis, justification and objective. The first group named control group, they received the printed material concerning to the elaboration of the preliminary draft, they were told not to use a lot of empty words and avoid to repeat the same word several times; they can consult with the teacher about the doubts concerning the work. On the other hand the second group called Experimental group, they were requested to use the intelligent tutoring system and the teacher explained how to use it. The time for the experiment was one week.

At the end of the experiment the results were analyzed, it was observed that the control group did not consult the teacher to review the work before final delivery, since they were not obligated to make revisions with the teacher, the students skipped this opportunity and they handed the document at reach the deadline. On the other hand, the experimental group used the intelligent tutor; they consulted more the material and the teacher to improve their drafting and get a higher score in the system and finish all the project elements. As we can see Figure 5 shows average results of lexical analysis for density, variety and sophistication of the problem statement of the experimental group and the control group. We can see that the experimental group had higher scores on all three lexical aspects.

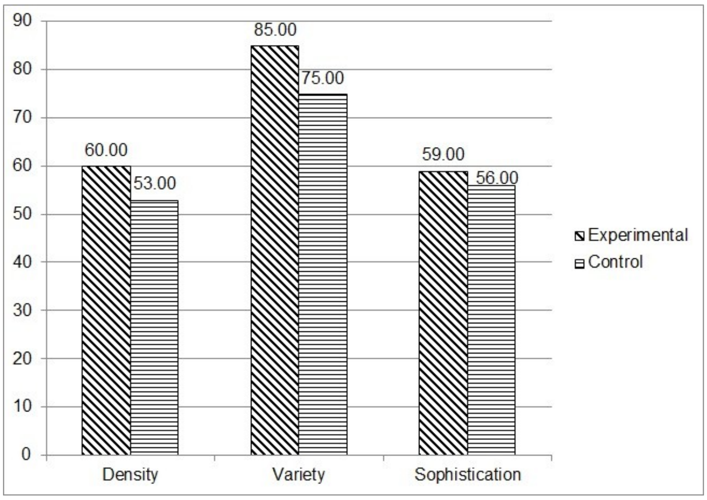


Fig. 5. Lexical analysis of the problem statement

The ranges established in the system to provide a textual feedback to the student in the density lexical analysis to the problem statement are: for a "low" value less than or equal to 52%, to a "medium" value greater than 52% and less than 59%, finally for a "high" value greater than or equal to 59%. According to these ranges we see that the control group obtained an average of 53% corresponding to a "medium" value and the experimental group with 60% which corresponds to a "high" value in the lexical density of the problem statement.

We also apply a satisfaction survey based on TAM model [10] (Technology Acceptance Model) to know the opinion of the experimental group in using the intelligent tutoring system in the aspects of system usefulness, system ease of use, system adaptability and intention to use the system.

Students answered based on a scale of five-point Likert scale ranging from 1 as "strongly disagree" to 5 as "strongly agree". We can see in Figure 6 the averages results by aspect of the satisfaction survey in which it is observed that the preference of the students is over 4 point equal to "agree" to all aspects, so we can conclude that the system is useful, easy to use, adapted to their level and they have the intention to keep using it.

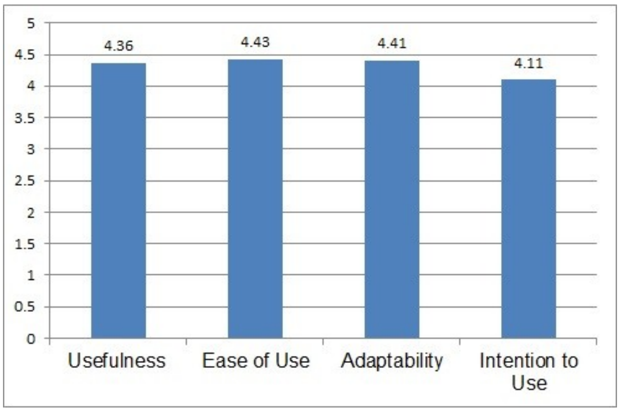


Fig. 6. Satisfaction survey results

5 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. As we can see the use the ITS improved the three lexical aspect: density, variety and sophistication, in the experimental group and according to the satisfaction survey it has a good acceptance among the students.

In future work, we intend to implement in the ITS the use of SCORM Learning Objects with RELOAD editor software and supported by an open source LMS to improve the portability of digital resource and improve the content assimilation in students. Expecting enhance the structuring of the student research projects.

References

1. Crossley, S.A., Varne, L.K., Roscoe, R.D. and McNamara, D.S.: Using Automated Indices of Cohesion to Evaluate an Intelligent Tutoring System and

an Automated Writing Evaluation System. In: Proceedings 16th International Conference, AIED 2013, Memphis, TN, USA. Springer, pp. 269−278 (2013)
2. Rospide, C.G. & Puente, C.: Virtual Agent Oriented to e-learning Processes. In: Proc. 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., and 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, no. 2, pp. 293−312 (2008)
5. Schwarm, S. and Ostendorf, M.: Reading level assessment using support vector machines and statistical language models. In: Proc. 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) 287−294 (2005)
7. Sucar, L.E., 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, No. 48, pp. 97−104 (2012)
9. González López, 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, NY, USA, pp. 215−220 (2012)
10. Tobing, V., Hamzah, M., Sura, S. and Amin, H.: Assessing the Acceptability of Adaptive E-Learning System. In: Proc. of Fifth International Conference on Learning for Knowledge-Based Society, December 11−12, 2008, Bangkok, Thailand, 10 p. (2008)

SCORM Sequencing and Navigation Model

Liliana Argotte[1], Gustavo Arroyo[1], Julieta Noguez[2]

[1] Instituto de Investigaciones Eléctricas, Reforma 113, Col. Palmira, 62490, Cuernavaca, Morelos, Mexico
[2] Tecnológico de Monterrey, Campus Ciudad de México, 14380, México, D.F., Mexico

{largotte, garroyo}iie.org.mx, jnoguez@itesm.mx

Abstract. One of the characteristics of e-learning is that students can acquire knowledge in different ways or with different learning paths, also called sequencing models. These sequencing patterns can be predefined by an expert from the type of learner or may be adaptive to the specific situation of each student. This paper analyzes the SCORM sequencing and navigation model for the dynamic presentation of learning content.

Keywords: Adaptive learning, learning objects, sequencing model, SCORM.

1 Introduction

The use of E-learning platforms and Learning Management Systems (LMS), and the design of online courses with their own educational technologies have increased significantly in education and business training. Learning Objects combined with LMS have been used to express knowledge, provide information and guide learning activities in the materials. The activity of an instructor or tutor is essential in guided learning: It is the tutor who selects the problems to show the learner, determines how to analyze the responses, presents the solution of certain problems, or decides to show some examples. The tutor also manages the training materials and is responsible for selecting the most appropriate material depending on the reported situation.

All of the above can be incorporated into a model similar to the offered by the Sharable Content Object Reference Model (SCORM) to display the material in an optimal structure determined by the instructor. SCORM is the most widely standard used for the development of E-learning courses. This standard is made up of independent units of information residing in a repository and viewable on any SCORM compatible LMS platform [1].

SCORM helps to define the technical basis for an online learning environment. This model has a Sequencing and Navigation structure for dynamic presentation of learning content based on the learner's needs. SCORM sequencing provides developers with E-learning courses with necessary tools to create complex designs that can even be adapted to individual learning needs of students, consistently applying sequencing capabilities that offer the following models [2]:

– Tracking Model,

– Activity State Model,
– Sequencing Definition Model,
– Navigation Model.

The SCORM Sequencing and Navigation structure is based on a version of the IMS Simple Sequencing, which defines a method to represent the expected behavior of a learning experience so that any LMS can generate the sequence of learning activities in a consistently manner [3]. In early versions of SCORM, the development of advanced online instructional courses was limited by several factors:

1. Sequencing Code embedded in the code of Learning Objects (LO);
2. Sequencing Behaviors inconsistent across Delivery Systems;
3. Sequencing Owners and Idiosyncratic Models; and
4. Sequencing Models and Activities ill-defined.

For many developers, the sophisticated intelligence structure of the course (for example, using diagnostic tests to derive individual routes or determine requirements for remedial instruction of each learner), is hard-coded in the LO. This technique limited the reusability, sharability and interoperability of SCORM-based training [4].

SCORM version 1.3 has largely solved the above-mentioned problems, and the Advanced Distributed Learning Community (ADL) has been given the task of developing and testing instructional content in accordance with the principles of Intelligent Tutoring Systems (ITS's) where developers can create instruction using more complex branching and sequencing [5].

This paper presents the characteristics of the sequencing and navigation structure of the SCORM model in order to analyze adaptive learning and the dynamic presentation of learning content.

2 Tracking Model

The values of this model are used for tracking or monitoring the sequencing control behavior. For each attempt on activity by a learner, that activity shall have associated tracking status data. Learner's interactions with a content object may affect the tracking data of the activity to which the content object is associated. Tracking data is used during the different sequencing processes to affect their behavior as follows (see Fig. 1):

– Communicative and Non-communicative content: The communicative content may transmit information about the learner’s interactions with the content through the SCORM Run-Time Environment Application Programming Interface (API), while non-communicative content does not use it.
– Suspending and Resuming Activities: An attempt on an activity may be suspended and later resumed. Resuming a suspended activity does not count as a new attempt and you can also try other activities while this is suspended, and there can be more than one activity suspended at any given time. Suspending the attempt on the root activity of the activity tree causes the

LMS to remember the last activity experienced by the learner in the activity tree and end the sequencing session in a suspended state. The learner may later resume the attempt on the root of the activity tree and at that time, it is will also resume the last activity performed by the student.

– Persistence of data: It is not specify how the data will persist between a beginner’s multiple sequencing sessions and a the activities of an particular tree, including, for example, if there are different learning experiences or multiple entries from the same activity. One attempt of this type can comprise one or more sequencing sessions. LMS policies should govern whether such information will be retained after this session.

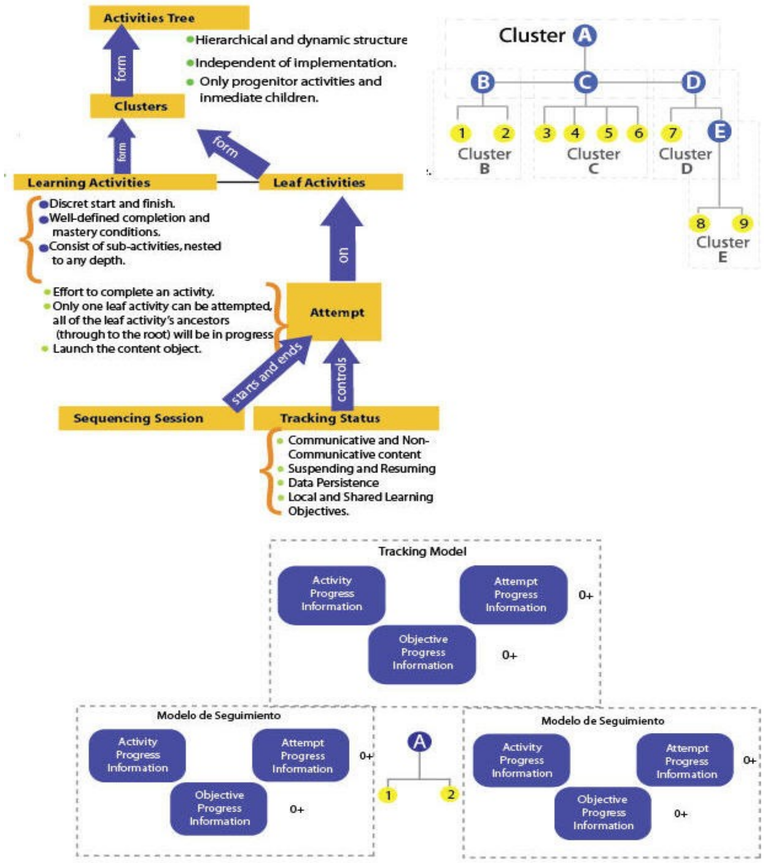


Fig. 1 Tracking Model

– Learning Objectives: Learning Objectives are separate from learning activities. SCORM does not impose any restriction on how learning objectives are associated with learning activities nor does it define how content objects are to use learning objectives. The SCORM Sequencing Behaviors make no assumption as to how to interpret learning objectives (e.g., is it a capacity domain, or is it a shared value, etc.). From a tracking perspective, a set of objective status information (objective satisfaction status and objective satisfaction measure) is maintained for each learning objective associated with a learning activity. Activities may have more than one target associated with them.

The tracking control status attempts in a learning activity. An attempt is defined as an effort to complete an activity, and during that effort, zero or more learning objectives may become satisfied. Attempts on activities always occur within the context of attempts on their parent activity(ies). It is noteworthy to mention that for a given activity tree one and only one leaf activity can be attempted at any given time and all attempts on all of the leaf activity's ancestors (through the root) will be in progress while the leaf activity is being attempted. When a leaf activity is being attempted, it can be assumed that the activity's corresponding content object has been launched.

An attempt begins when the activity is identified for delivery and ends while the LMS's sequencing implementation attempts to identify the next activity for delivery. An attempt on an activity is closely related to learner attempt on the activity's associated content object. A sequencing session is the period between the time learners begin the attempt on a tree root activity of activities and the time that such attempt ends, outside the context of the sequencing session, it is considered that the current activity is undefined.

A learning activity can be described as a useful unit of instruction; it is conceptually something the learner does while progressing through the instruction. A learning activity may provide a learning resource to the learner or it may be composed of several sub-activities. A cluster includes a single parent activity and its immediate children, but not the descendants of its children.

Finally an activity tree is a general term that represents an instance of hierarchical learning activities and the corresponding sequencing information for the interoperable application of specified sequencing behavior. Because the SCORM Sequencing Behaviors are defined in terms of structured learning activities, a functional content structure provides the necessary starting point for deriving an activity tree. In terms of sequencing, a content organization represents an interoperable structure of an activity tree. The content organization (<organization> element) is the root of the activity tree and each of its <item> elements correspond to a learning activity, depending on the granularity.

3 Activity State Model

This model manages sequencing state of each activity in the Activity Tree and the global state of the Activity Tree. This is a dynamic run-time data model utilized by the LMS's sequencing implementation to manage the state of the activity tree during a sequencing session. The overall sequencing process uses the following sequencing behavior (see Fig. 2):

– Navigation Behavior: Describes how a navigation request is validated and translated into requests for completion and sequencing.

– Termination Behavior: Describes how the current attempt on an activity ends, how the state of the activity tree is updated and if some action should be performed due to the attempt ending.
– Roll up Behavior: Describes how tracking information for cluster activities is derived from the tracking information of its child activities.
– Selection and Randomization Behavior: Describes how the activities in a cluster should be considered during processing a sequencing request.
– Sequencing Behavior: Describes how a sequencing request is processed on an activity tree in attempt to identify the next activity to deliver.
– Delivery Behavior: Describes how an activity identified for delivery is validated for delivery and how an LMS should handle delivery of a validated activity.

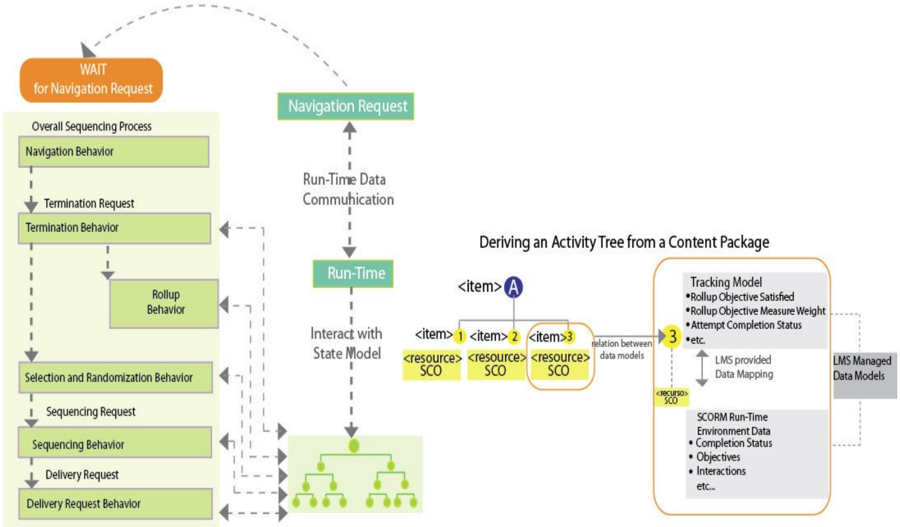


Fig. 2 Activity State Model

4 Sequencing Definition Model

This model defines a set of elements that may be used by content developers to define intended sequencing behavior. The definition model elements are applied to learning activities within the context of an activity tree (see Fig. 3). Each element has a default value that is to be assumed by any sequencing implementation in the absence of an explicitly defined value. SCORM does not require or imply that the values of sequencing definition model elements apply to an activity are, become or remain static for any period. The LMS may alter the element's value as you like, however, some groups of sequencing definition model elements are highly coupled to one another through the SCORM Sequencing Behavior.

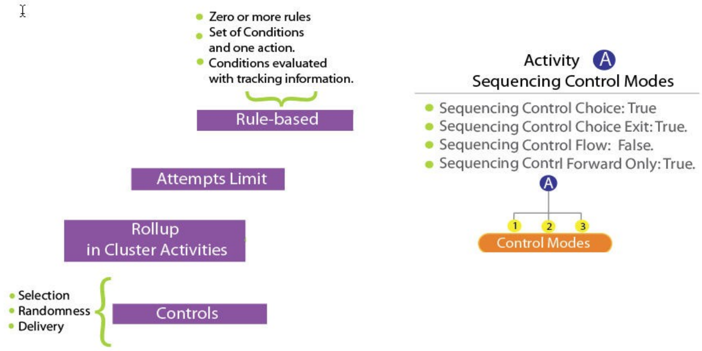


Fig. 3 Sequencing Definition Model

The Sequencing Control Modes allow the content developer to affect how navigation requests are applied to a cluster and how the cluster's activities are considered while processing sequencing requests. Sequencing Control Modes may be applied, as needed, to constrain a desired learning experience. Control option restriction defines a set of controls of restricted options to impose greater conditions and behaviors regarding the manner in which applications are processed by sequencing.

SCORM sequencing employs a ruled-based sequencing model. A set of zero or more sequencing rules can be applied to an activity and the rules are evaluated at specified times during various sequencing behaviors. Each sequencing rule consists of a set of conditions and a corresponding action. The conditions are evaluated using tracking information associated with the activity. The behavior associated with the rule's action is performed if the rule's condition set evaluates to True. The structure of a sequencing rule is (if [condition] then [action]).

A content developer can define limit conditions that describe conditions under which an activity is not allowed to be delivered. Limit conditions can be associated with activities and are conditional based on activity's tracking status information. When a limit condition is met or exceeded, the activity becomes unavailable for delivery. SCORM only requires the support for the Limit Condition Attempt Limit element. Attempt limit does not require the evaluation of any time-based limit conditions. Therefore LMS's are not required to manage data for or honor the evaluation of any of the optional portions of the Limit Conditions Check Process.

Cluster activities are not associated with content objects, therefore is no direct way for learner progress information to be applied to a cluster activity. SCORM sequencing provides a way to define how learner progress for cluster activities is to be evaluated. A set of zero or more Rollup Rules may be applied to a cluster activity and the rules are evaluated during the Overall Rollup Process. Each Rollup Rule consist of a set of child activities to consider, a set of conditions evaluated against the tracking information of the included child activities, and a corresponding action that sets the cluster's tracking status information if all conditions evaluates to True. Rollup Rules have no effect when defined on a leaf activity.

There is a mechanism for learning objectives associated with an activity. An activity may have one or more learning objectives associated with it. SCORM does not describe how a learning objective is defined, used or read, but for sequencing purposes, each learning objective associated with an activity, have a set of tracking status information that allows student progress towards the learning objective, thus enabling conditional sequencing decisions.

There are randomization controls that describe when and what actions will the LMS take to reorder the available children of encountered cluster activities, while performing the various sequencing behaviors. Content developers may apply randomization controls to any cluster in the activity tree.

There are delivery controls describing actions that LMS will take prior to an attempt on an activity beginning and after the attempt ends. Delivery controls shall be used by LMS's to aid in the management of the activity's tracking status information. The elements also indicate whether the LMS can expect the SCO associated with the activity to communicate specific types of tracking information.

5 Navigation Model

The SCORM Navigation Model considers the use of the following concepts or processes:

- Start Navigation Requests. The navigation model only applies to SCORM navigation between learning activities. SCORM currently does not directly address the ability to define the sequencing or navigation within a Sharable Content Object (SCO). However, SCORM does not preclude the ability to navigate between SCO's (this ability is completely controlled by the SCO).
- The SCORM Navigation Model defines a set of navigation events that can be triggered by a student through an LMS and content provided user interface devices or directly by SCO's. SCORM does not define how these events are activated within a SCO or through the LMS. Navigation requests are processed as defined by SCORM Sequencing Behaviors. They provide the learner and content an interoperable means to indicate how the progress through an activity tree is; such as to choose a particular learning activity, continue to the next activity or go back to a previous activity.
  - Processing Navigation Requests. When the learner or content triggers a navigation event through any mechanism, the LMS processes the corresponding navigation request by invoking the sequencing system: The result of processing the navigation request will always be one of the following:
  - If the effect of the navigation request is to end the current attempt on the activity tree, the LMS will process an Exit All navigation request, which ends the attempt and returns control to the LMS.
- After evaluating the current tracking status and the applicable sequencing information on the activity tree, the LMS determines that processing the intended navigation request should not be honored. In that case, the LMS ignores the navigation request. The LMS takes no sequencing action until another navigation request is triggered.
- Completion of Content Objects by Navigation Means.

– From this description it is noted that if a LMS offers user interface devices for navigation events, the learner may indicate their willingness to navigate by triggering one or more of these devices. When learners indicate their desire to navigate, SCORM assumes the learner is implying he just ended with the currently launched content object, if any. If the chosen LMS navigation attends the event initiated by the learner, you must first remove (download) the content object released at that time and then process the request for proper navigation. The content object must be removed before processing the navigation request to ensure that the content object has recorded information on learner progress that may affect sequencing.

The SCO's intentions can communicate directly to LMS through the SCORM Navigation Data Model. A SCO must also indicate to the LMS should act on its intentions; this is done by invoking that the method is complete, which indicates that the SCO has completed communication with the LMS. Therefore, if the SCO has completed communication and indicated a navigation intention, the LMS system must respond. Once the request has finished processing, the LMS process any outstanding navigation event initiated by the learner. If there were no outstanding navigation events initiated by the student, the LMS will process the last navigation request sent by the SCO. If neither the student nor the SCO indicates their navigation intentions, the LMS should wait for the learner to indicate a navigation event.

6 Conclusions and Future Work

SCORM does not address, but does not exclude artificial intelligence-based sequencing schedule-based sequencing, sequencing requiring information from closed systems and external services, collaborative learning, customized learning, or synchronization between multiple parallel learning activities.

In this paper we presented the sequencing model and SCORM navigation; our hope is that this paper will serve as a useful tool in helping readers who are not yet experts in the field. It is considered that the SCORM Sequencing and Navigation Model is not enough to adapt Learning Objectives (LO’s) Sequencing to each student’s specific learning situation because it is a rule-based model where the instructor plays a key role to previously set different learning paths that learner may have. As it is also a deterministic model, it is based on the achievement of certain objectives to determine whether or not a student has been progressed. Then, it is proposed as future work, to present a more comprehensive model (not deterministic) aided by Artificial Intelligence tools managing uncertainty and could be used to add intelligence and adaptation to the already existing SCORM sequencing model.

References

1. ADL-Overview: SCORM 2004 4th Edition Overview Version 1.0. Washington, Advanced Distributed Learning (2009)

2. ADL-SN: SCORM 2004 4th Editon Sequencing and Navigation (SN) Version 1.0. Washington, DC, Advanced Distributed Learning (2009)
3. IMS Global Learning Consortium: IMS Simple Sequencing Behavior and Information Model v1.0 Final Specification (2003). Retrieved 02/07/2009, from http://www.imsglobal.org/
4. Panar, A., and Smith, S.: Sequencing Overview and Demonstration. Presented at Plugfest 7, Orlando Convention Center. Orlando, Florida (2002)
5. Panar, A., & Thropp, S.: What's new in SCORM version 1.3. Presented at Plugfest 8, Carnegie Mellon University. Pittsburg, Pennsylvania (2003)

Virtual Reality and Affective Computing for Improving Learning

Norma J. Ontiveros-Hernández[1], Miguel Pérez-Ramírez[2], Yasmín Hernández[2]

[1] Instituto Tecnológico de Zacatepec
Calzada Tecnológico No 27, Zacatepec, Morelos, 62780, Mexico
[2] Instituto de Investigaciones Eléctricas
Reforma 113, Col. Palmira Cuernavaca Mor., 62490, Mexico

njoh_314@yahoo.com.mx, {mperez, myhp}@iie.org.mx

Abstract. After several years of working independently in development of Virtual Reality (VR) training systems and affective computing, we have decide to undertake an incursion on development of new systems which integrate both fields, so that training can be improved. Here first is presented evidence of importance of emotion in human activity and specifically in training, then we present preliminary insights towards a model for VR inducing emotional states.

Keywords: Virtual reality, student affect, intelligent tutoring systems.

1 Introduction

It has been argued that Virtual Reality is effective for creation of learning contexts within the comprehensive approaches to learning [1]. Here it is claimed that VR is also an effective tool for creation of virtual environments which are able to originate or influence users' emotional states. Although not frequently considered, emotional states of people are present in the whole spectrum of humankind activities. It is only until recent years that computer scientists have started to pay attention to emotions in order to improve human computer interaction [2]. In particular, emotions are also present in any learning process; therefore training systems which understand and deploy emotional states of students might be more effective as training tools. The aim of this paper is to analyze and provide evidence that the emotional factor cannot be omitted in the multidimensional approach to learning, where it is considered that the more influencing dimensions are integrated in each specific learning process; the more efficient is the instruction to reach specified learning goals.

On one hand, VR possesses huge potential for content and learning contexts creation, which integrates different factors favoring knowledge transference. A learning context is conceived as the sum of factors which intervene in a specific learning process. The architecture followed by Virtual Reality Group (VR Group) at IIE, for developing VR training systems, allows observing the benefits of VR in the creation and integration of different influencing factors (dimensions) in the learning process.

On the other hand, in recent years has arisen a new strand of the computer science called affective computing [2, 3], which has shown the importance of considering emotions in human-computer interaction and that might have benefits within different application fields.

Thus, in this paper first we analyze the role played by emotions in learning and provide insights of the features we might be able to integrate to our VR systems so that they provide the emotional support in order to improve learning. Then it is proposed to include the affective dimension (factor) into the comprehensive approach to learning.

The rest of the paper is organized as follows. Section 2 includes a brief description of the comprehensive approach to learning. Section 3 relates VR and Affective Computing. Section 4 shows some insights of the VRG towards a model of VR inducing emotional states. Finally some conclusions are presented followed by a list of references.

2 Multidimensional Approach to Learning

2.1 VR for Training

Virtual Reality is the electronic representation (partial or complete), of a real or fictitious environment. Such representation can include 3D graphics and/or images, has the property of being interactive and might or might not be immersive. [4].

It has been extensively supported in the literature the advantages of VR in a variety of fields [5, 6]. The systems developed for the VRG are mostly devoted to free risk training of highly dangerous maintenance procedures, involving medium tension live lines maintenance and tests to substations.

They operate in three modes namely, learning, practice and evaluation (Fig. 1 a). Before a user enters to any of these modes, the systems allows users to visualize and manipulate catalogs of 3D models of all the tools and equipment needed for maintenance work without being in a company's warehouse (Fig. 1 b).

2.2 Comprehensive Approach to Learning using VR

This approach has been detailed in [1], here is provided a short description. Different approaches and theories have arisen to improve learning, such as behaviorism, constructivism and others might be included. One of the problems here is that instructional design usually does not target groups of students with the same skills; rather they are applied to a heterogeneous audience of learners each one with different skills.

(a) (b)

Fig. 1. (a) Learning mode (b) Tools catalog

From this intuition we can observe that a learning process requires a more comprehensive view so that instruction can impact learning in a broader audience. Following the comprehensive approaches, when dealing with instruction, there is a variety of different dimension or factors which intervene in a learning process (Fig. 2) and that must be considered if we want to accomplish the main goal of any instruction task (knowledge transference). These dimensions can vary on different situations, some are mentioned here:

– *Learner- instructor dimensions:* These dimensions involve, a perhaps just assumed, but decisive demand in order to get a combined effort to get the training goal namely, learners must really want to learn and instructors must really want to teach.
– *Instructional model dimension:* Different instructional model have been proposed (e.g. Behaviorism, Constructivism, etc.) each having strengths and weaknesses. They all provide some truth and some approach for learning improvement (e.g. learning centered on instructors, learning centered in students, learning centered on instructor-student interaction, etc.). Depending on the instruction domain, a model or combination of models must be selected in order to make the instruction efficient.
– *Instructional domain dimension:* It is not the same football training, which is mostly a physical activity than a physics lesson which might be mostly theoretical. It is clear that each domain demands specific abilities from learners, but also determines which instructional method can be better to reach an instructional goal.
– *Learning channels dimension:* Usually three different kinds of learners are identified according to dominant learning channel, namely auditory for those who learn better by hearing, visual for those who learn better through visualization and kinesthetic for those who learn better by manipulating objects. Providing stimuli for these three channels, instruction might reach and be successful in a broader audience.
– *Affective dimension*: Here we are assuming that the affective state of a student and the instructor might also determine whether or not a learning goal is reached successfully. As a simplistic example, a student might have discussed with his girlfriend just before attending a presently classroom lesson. This might originate

an affective state, which derives in distraction and lack of concentration in class. As another example, depending on the exposition method used by the instructor, he might cause an apathy state on students or might motivate them to concentration and reaching effectively the learning goal planed. Emotional states are inherent to students; therefore the affective dimension must be part of the comprehensive approaches to learning.

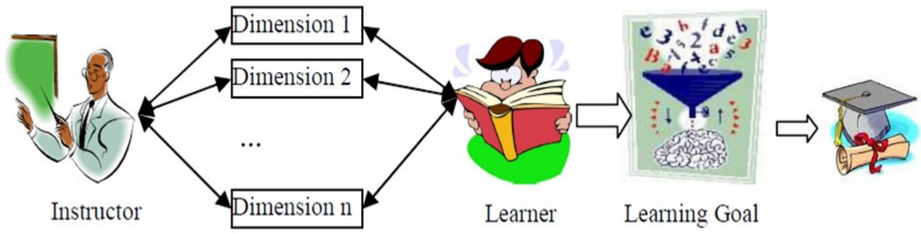


Fig. 2. Different dimension intervene in a knowledge transference task.

Following the comprehensive approaches, intuitively we would expect that identification and integration of influencing factors or dimension (including the affective dimension) into specific learning contexts, would improve the learning process. Since we want to undertake an incursion in affective computing to see how we can integrate the affective dimension in our VR systems, we analyze this technology below.

3 Affective Computing and Virtual Reality

3.1. Affective Computing

The Dictionary of the Spanish Real Academy of Language (RAE, Spanish acronym) [7], defines affective as "Belonging or relative to affect", in turn one of its meanings defines affect as "each of the passions of the mood, such as, anger, love, hate, etc., and specially love and affection." These "passions of the mood" seem to correspond to what within the computational (affective) scope is referred as "basic emotions" where authors provide examples such as fear, anger, sadness, pain, joy, annoyance, etc., but it seems that there is no consensus among different authors who consider different lists of emotions. Even more, it is considered that emotions are not expressed purely, that is to say, only one emotion at a time, rather it is expressed a mixture of the so called basic emotions.

In the evolution of computing, specifically within the field of Human-Computer Interaction (HCI), there are attempts to find more efficient HCI by integrating emotional communication, that is, the affective computing [8]. Picard [2], developed computational methods for human emotion recognition and generate synthetic emotions in order to improve HCI (see Fig. 3).

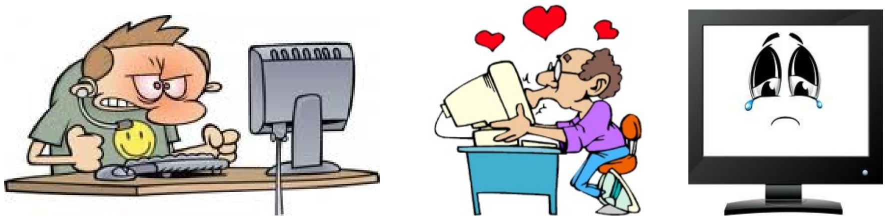


Fig. 3. Affective computing

Usually HCI focuses in the logic aspects of the interaction, in HCI which makes sense. Even in the field of Natural Language Processing, it is pursued a more natural and logic HCI. However, these attempts are simulations which incursion in some functions of only one of the brain hemispheres, the left side. This side of the brain is in charge of the logic reasoning. This effort by the HCI community is necessary but incomplete. Users have other brain hemisphere, the right side, which among other thing, is in charge of the emotional aspect and when a user interacts with a computers he makes it with both hemispheres not only with one of them. This is why integrating the emotional aspect in the HCI, might provide computer systems with more complete and efficient interfaces. On the other hand, affective computing might also widen the application spectrum of computers in general and make it more effective.

3.2 Affective Computing and VR

VR is not only efficient for creation of learning contexts, the vast literature shows that VR is affective also in the creation of environments with different degrees of influence in users' emotive states. Some examples are the following: Riva et al [9] present a study in which is reported the efficacy of VR as an effective mean to induce and influence emotions. They observed that interaction with relaxed and anxiety virtual environments induced relaxation and anxiety states in the users respectively. The sceneries represented a relaxing park (well illuminated and with nice aspect) and the anxiety park (dark and with sullen aspect). The data presented by these authors show a dual feedback between presence and emotion, the feeling of presence increased in the emotional environment and the emotional state in turn was influenced by the level of presence.

VR has also been used in virtual worlds, in which users interact with other users by means of avatars that they can choose as representations of themselves (resembling the use of nicknames in chat channels). Users control their avatars, so they can move freely within the virtual environment, talk, make gestures or represent emotions [10]. All this behavior will be perceived by other users who are represented by their respective avatars and with the same level of control over them, and as in the real world, the behavior of a user might affect the others'. Perhaps one of the most representative examples of environments like these is second live [11].

Another field in which emotions have drawn quite a lot of attention is the game industry [12], in which we can find many of them based on VR. No wonder, in some degree, emotions is what game developers sell.

It can be observed that VR is able to play a role in the creation of emotion influencing contexts, and therefore it can be also a possible creation tool for Affective Computing. We would expect that a VR learning context observed substantial improvement for knowledge transference if it is enriched with Affective Computing technology.

3.3 Affective Computing in Training

Even in the 50's, affect was considered in learning, the so called Bloom's taxonomy, proposed that educational objectives were divided into three "domains": Cognitive, Affective, and Psychomotor [13]. Now, association between Affective Computing and learning is sometimes called affective learning (AL). AL systems try to recognize and deploy emotions during the learning process so that knowledge transfer is more effective. There are intuitions about the relationship between emotional states and learning. For instance Miller [14] points out that affective learning outcome involve attitudes, motivation, and values. Minsky [15] suggests that different emotional states induce different kinds of thinking. Picard et al. [3] mention that a slight positive mood induces a different kind of thinking, characterized by a tendency toward greater creativity and flexibility in problem solving. We can also observe the satisfaction originated by an achieved goal, that is to say, when partial learning objectives are achieved, the students experience a feeling of satisfaction, which in turn motivate them. The opposite is also common, when learning goals cannot be achieved, it produces frustration.

From these few examples we can identify some key terms such as: motivation, positive mood, satisfaction, achieved leaning goal, frustration, and kind of thinking. Others are: encouragement, reflective thinking, active learning, interesting, curiosity, etc. It is clear that there is the need of finding a precise relationship between emotions and learning aspects, but this involves challenges with no answer so far. On one hand, we recognize some emotions but we still do not know how to measure them and usually we do not even know how to control our own emotions. This reminds us that we are dealing with the other side of the brain, the no logical one. Thus, if we want to improve leaning from the affect dimension, we need to influence students' emotions, which demands us some kind of control on others' emotions (Fig. 5).

There are also different works relating the three fields: affect-learning-VR. Some examples are the following: Ho-Shing et al. [17], propose a Smart Ambience for Affective Learning Model, based on VR, which focused on learning of concepts of animal survival. In experiments they conducted, report a high correlation between positive feelings and high learning rate for players. Lee [18] reports a study in which effectiveness of using desktop virtual reality for learning was evaluated; outcome was measured through academic performance whereas affective learning outcomes were measured through perceived learning effectiveness and satisfaction. One of his conclusions was: The results imply that desktop VR technology was effective in boosting the students' affective behavior and the perception of their learning experience. In [19] the relation affect-learning-VR is given through educational games, the OCC model is adapted and affective and cognitive user modeling is achieved by combining evidence from students' errors, this on the basis that achievement and error influence students' affective states.

Fig. 4. Controlling mood? (Taken from [16])

There is in fact a vast and recent literature in which VR is being used as a tool for affective learning.

4 Towards the Model for VR Inducing Emotional States

4.1 Towards the Model of VR

One of the features of VR which might support effectively the induction of emotions, is that it can induce feelings of presence, "the feeling of been there." This has a huge potential, since depending on the kind of emotive states wished to induce, it can be decided what kind of elements to include in specific VR environments. Thus, possibilities are unlimited since creation of VR environments is only limited by imagination of creators. This provides evidence that VR is effective to influence and induce emotions. Fig. 5 shows a simplistic view of what might be a VR model for AC.

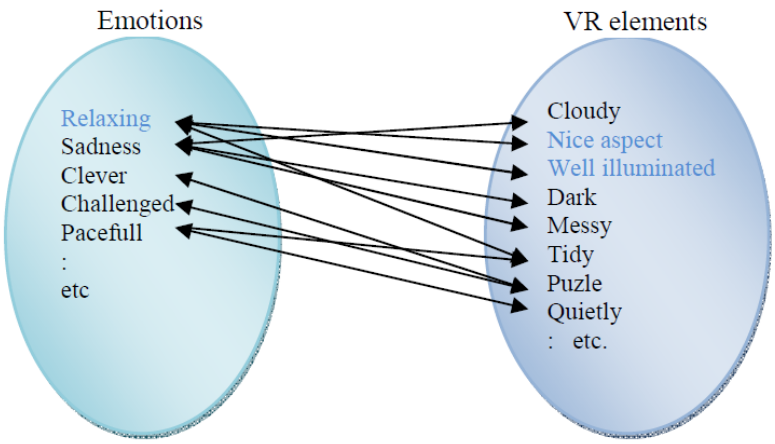


Fig. 5. Toward a model for the integration VR and Affective Computing

The overall idea seems to be quite simple; each emotion must be related with a set of VR elements which induce such emotion. Nevertheless, deeper research is needed to make a mapping between a set of emotions and a set of elements which induce those emotions. Then in order to implement this mapping by using VR, we might map these elements into VR elements, which can be presents within VR learning environments trying to influence students' mood.

An instance of the model is shown in Fig. 6. Based on the work of [9] described above, where the authors relate relaxing park environment with the feeling of relaxation. In turn the relaxing park is associated with a well illuminated place and with nice aspect. Then Fig. 7 relates for instance, a 'relaxed' state with 'nice aspect' and 'well illuminated'. This does not mean that those are the only VR elements needed to trigger a relaxing state; it is only exampling the mapping between feelings and VR elements. This mapping would provide some guidelines for the features that should integrate the virtual reality environment so that it originates this specific emotive state on users.

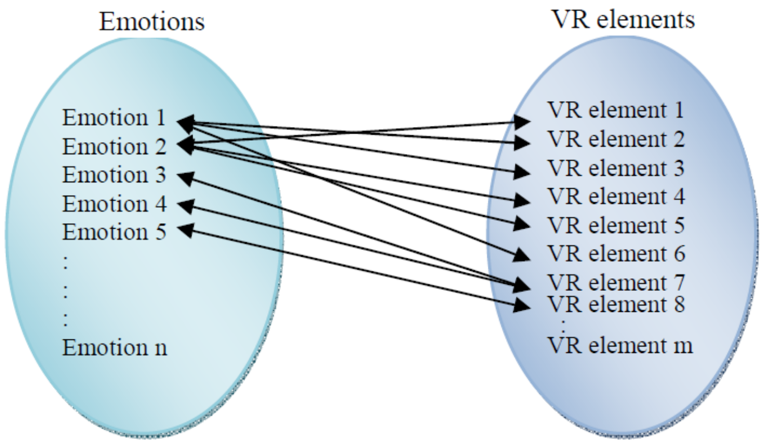


Fig. 6. Example of mapping between emotions induced by possible elements within virtual environments on the basis of presence

VR can be really powerful at inducing emotional states because it possesses different tools which enables it to integrate different VR elements within any virtual environment, examples are: recreation of environments with different levels of realism, different intensities and colors of lights, integration of different kind of audios and sound effects and even music, visual representation of environments and visual effects, avatars able to show different emotional states, integration of voice, and it is even possible to represent unreal environments, among others. All these elements provide VR with a vast arsenal for the creation of a wide variety of environments each addressing some emotive state(s).

Thus, VR reality is able to create different realities and engage users within them through the feeling of being there (presence), inside these realities and unchaining different emotive states.

Nevertheless, this simplistic model must be completed with other features already proposed by other authors.

VR and AC have been integrated within the learning field, for instance Elliot et al [20] believe that animated 3D pedagogical agents would make a good tutor if they understand and deploy emotions. Based on STEVE (Soar Training Expert for Virtual Environments), they also provide intuitions about the features that a PA should deploy in order to improve learning. The PA must appear to care about student and his progress, so that he feels that he and the PA are “in things together”. The PA must be sensitive to student´s emotion in order to encourage in case of frustration, should convey enthusiasm for the subject matter, in order to foster enthusiasm in the student, but at the same time the PA should make learning more fun.

Table 1: Set of possible behaviors for a Virtual Pedagogical Agent

Behaviors	Description
Negation	Shaking head as sign of negation
No understanding	Slight leaning trying to hear
Suggesting	Extend hand in signal of explanation
Sadness	Showing a sad face
Thinking	Looks aside and puts a hand on the chin
Saluting	Nodding head and rising a hand
Congratulating	Applauding
Waiting	Observing the actions of the students
Recognition	Nodding head
Happy	Happy expression
Explaining	Extend both hands and point to elements in the environment

Considering the fast changing nature of context, in [21] is proposed a dynamic model based on the OCC cognitive model for emotions [22]. The model of the student is composed by three elements namely: the profile of the operator, the pedagogical model and the affective model. The emotions of the student are thought of as a function of achieved goals. Here the student emotional state is inferred from indirect sources such as personality and knowledge of the topic.

Once the affective state of the student is inferred, the tutor should decide which the learning activities are most suitable for him. Then it establishes a relation between the affective and pedagogical state of the student with the training actions oriented to improve training. It is also proposed an animated PA with the behaviors listed in Table 1.

To start with, the VR Group might integrate this set of behaviors listed in Table 1, to an animated agent within a virtual environment of the systems developed.

All these intuitions might be another kind of elements in addition to VR elements which can be included in the emotions model based on VR.

In the virtual reality model for emotion, there might be even the possibility of regulating the intensity of the emotion originated by adding of removing VR elements in the virtual environment. For instance, a sullen aspect environment might be added with audio effects or shady animations to increase the filling of anxiety.

4.2 Emotions as a Dimension in the Comprehensive Approaches to Learning

On the basis of the shallow analysis presented above we have evidence of the role played by emotions in the human activity including the learning process. In section 2.2 we have included the affective dimension within the comprehensive approach to learning. In the previous section we have even identified some behaviors that might constitute a departure point to consider the affective dimension in the VR Group developments. Nevertheless, the VR Group still needs to do more research in order to complete a VR model for emotions, to find a precise mapping between emotions and VR elements, inclusion of affective PA, etc.

5 Conclusions

Discussion presented before seems to allow us inferring the wide potential of VR for influencing emotions. We have presented the first insights towards an emotions model based on VR. Deeper research is needed to make a better mapping [VR elements] ↔ [emotions]. VR Group has developed different training systems whose development methodology can be enriched with infrastructure to integrate emotive aspects which surely will improve the learning process.

Affective Computing might be located into an upper level within the evolution of artificial intelligence. Beyond pursuing the simulation of human cognitive and inferential processes, it comes into the emotional aspect which is under control of the other brain hemisphere (right) and somehow considered as a human factor separated from reason (left hemisphere), although they are rather complementary. IIE has been working separately on Affective Computing and VR, but now in order to improve training and offer better systems to our customers, in a very near future we want to integrate both fields within a new generation of training systems based on VR.

References

1. Miguel Pérez-Ramírez, Norma J. Ontiveros-Hernández: Virtual Reality as a Comprehensive Training Tool. WILE-MICAI. Guanajuato, Mexico (2009)
2. Picard, Rosalind W.: Affective Computing. MIT Press, USA (2000)
3. R.W. Picard, S. Papert, W. Bender, B. Blumberg, C. Breazeal, D. Cavallo, T. Machover, M. Resnick, D. Roy and C. Strohecker: Affective learning — a manifesto. BT Technology Journal 22(4) (2004)
4. Pérez Ramírez Miguel, Zabre Borgaro Eric y Islas Pérez Eduardo: Prospectiva y ruta tecnológica para el uso de la tecnología de realidad virtual en los procesos de la CFE. Instituto de Investigaciones Eléctricas. Reporte Interno IIE/GSI/022/2003 (2003)
5. Grigore C. Buerdea and Philippe Coiffet: Virtual Reality Technology. Second Edition, Wiley- Interscience (2003)
6. K.M. Stanney (Ed.) Handbook of Virtual Environments, design implementation and applications. Mahwah, NJ: Lawrence Erlbaum Associates Publishers, 1232 p. (2002)
7. Diccionario de la Real Academia Española de la Lengua. http://rae.es/
8.

9. Emiliano Causa, Andrea Sosa: La Computación Afectiva y el Arte Interactivo. Noviembre de 2007. http://www.biopus.com.ar/txt/textos/Computacion_Afectiva_Y_Arte_Interactivo-Emiliano_Causa-Andrea_Sosa.pdf
10. Giuseppe Riva, Fabriziamantovani, Claret Samantha Capideville, Alessandra Preziosa, Francesca Morganti, Daniela Villani, Andrea Gaggioli, Cristina Botella, and Mariano Alcañiz: Affective Interactions Using Virtual Reality: The Link between Presence and Emotions. CYBERPSYCHOLOGY & BEHAVIOR 10(1) Mary Ann Liebert, Inc. DOI: 10.1089/cpb.2006.9993 (2007)
11. Digital Space Technologies Inc. Digital Space 3D Authoring Kit, 2.0, P/N 2003-00-0, EE. UU (1997)
12. Second Life. http://secondlife.com
13. Eva Hudlicka: Affective Computing for Game Design. In: Proceedings of the 4th Intl. North American Conference on Intelligent Games and Simulation (GAMEON-NA), McGill University, Montreal, Canada, pp. 5−12 (2008)
14. Terri Langan: Bloom's Taxonomy for Affective Learning and Teaching. on line (2012)
15. Mary Miller: Teaching and Learning in Affective Domain. The University of Georgia. http://epltt.coe.uga.edu/index.php?title=Teaching_and_Learning_in_Affective_Domain
16. Minsky M: The Emotional Machine, http://web.media.mit.edu/~minsky/ (2003)
17. Wikicurios. La ptenorofobia, el miedo a las cosquillas. Wikicurios ~ Información y curiosidades en un click. 16 septiembre, 2011 de Wikicurios en Curiosidades, Salud. http://wikicurios.com/2011/09/16/la-ptenorofobia-el-miedo-a-las-cosquillas/ (2011)
18. Horace Ho-Shing Ip, Julia Byrne, Shuk Han Cheng, and Ron Chi-Wai Kwok: The SAMAL Model for Affective Learning: A multidimensional model incorporating the body, mind and emotion in learning. DMS, pp. 216-221. Knowledge Systems Institute (2011)
19. Elinda Ai Lim Lee: An investigation into the effectiveness of virtual reality-based learning. PhD thesis, Murdoch University. http://researchrepository.murdoch.edu.au/4070/ (2011)
20. George Katsionis and Maria Virvou: Personalised e-learning through an Educational Virtual Reality Game using Web Services. Multimedia Tools and Applications. 39(1) 47−71, (2008)
21. Clark Elliott, Jeff Rickel, and James Lester: Integrating affective computing into animated tutoring agents**.** In: Proceedings of the IJCAI-97 Workshop on Animated Interface Agents: Making them Intelligent, pp. 113−121 (1997)
22. Yasmín Hernández, Gustavo Arroyo-Figueroa, L. Enrique Sucar, Miguel Pérez-Ramírez: Comportamiento afectivo en sistemas de capacitación inteligente para operadores de sistemas eléctricos. Komputer Sapiens (2013)
23. A. Ortony, G.L. Clore and A. Collins: The Cognitive Structure of Emotions. Cambridge University Press (1998)

Strategic Learning, towards a Teaching Reengineering

Rafaela Blanca Silva-López[1], Iris Iddaly Méndez-Gurrola[1],
Victor Germán Sánchez Arias[2]

[1] Universidad Autónoma Metropolitana, Unidad Azcapotzalco
Av. San Pablo 180, Col. Reynosa Tamaulipas, Del. Azcapotzalco, México, D., Mexico
[2] Universidad Nacional Autónoma de México
Circuito Escolar Ciudad Universitaria, 04510, México, D.F, Mexico

rbsl@correo.azc.uam.mx, iddalym@yahoo.com.mx,
victor_sanchez@cuaed.unam.mx

Abstract. This article presents the strategic learning meta-model (SLM). We describe the architecture of the SLM, which consists of three layers: reactive layer, intelligent layer and infrastructure layer. The purpose of this paper is to present the reactive layer of the SLM and in particular the model that constitutes: the regulation model. This model was tested with a series of cases which indicate an increase in student performance in a particular course. Furthermore briefly presents the design of intelligent layer of SLM, which consists of a set of ontologies, this paper only presents the design and implementation of 3 of the ontologies that comprise the ontology model. Finally, the strategic learning meta-model proposed integrates the principles of mediator evaluation, customizing of learning route, monitoring and personalized attention, work in learning communities with the aim of providing better learning opportunities, optimizing the physical and human resources an institution, with the aim of reducing desertion rates.

Keywords: Strategic learning meta-model, self-regulated learning, ontological model, diversity and assessment.

1 Introduction

Mexico educational models do not respond to the problem of school failure leading from the courses reprobation to the terminal efficiency. This problem becomes relevant internationally and is manifested in the publication of articles and books. Is imminent the necessity of proposing new alternatives to experiment and find alternative solutions to the problem of reprobation [1].

Since 1995, Baena [2] deals with this problem by performing an analysis of the students' failure and desertion rates in the specialty of Political Science at the Political and Social Sciences Faculty of the UNAM. He focus is on the teaching-learning process and proposes a change in the teaching methodology.

In 1999, Juan Pozo and Carles Monereo [3] constitute a research community in which productive dialogue peer opens an options range and proposals aimed at connecting the disciplinary content learning with learning strategies, they agree on the importance of the dichotomy of disciplinary content and strategies or learning process in order to achieve a strategic learning. Talk about strategic learning involves a commitment to the integration of mechanisms that allow students to learn to learn, considering: a) the student's ability to manage their own learning, b) the adoption of autonomy in their learning, c) the provision of methodologies and tools for continuous learning throughout life[1].

In this paper we present the psychoeducational theoretical framework that supports the architecture of strategic learning meta-model. An analysis of various authors was made and are taken up basic concepts to define the layers that make up the architecture of the strategic learning meta-model constituted by three layers: a) reactive (regulatory model), b) intelligent (ontological model); and c) infrastructure (Virtual Environment of Custom Learning).

2 Psychopedagogic Theoretical Framework

Authors as Weinstein [4], Pintrich [5], Castañeda and Lopez [6] and Monereo [7] have positions for which models have been developed that attempt to encompass aspects comprising learning strategies, strategic knowledge representation referred to teaching and learning contexts, in which self-regulation and motivation are essential. Below we briefly describe some of these models.

Weinstein model: learning strategies are the thoughts and behaviors that students engage in their learning, which influence the cognitive processes associated with the encoding of information, registration memory and learning outcomes. Weinstein classified the strategies into two blocks, which focus on the information that is going to learn and those are supported by meta-cognitive aspect and emotional. These strategies are integral part to the learning regulation [4].

Self-regulated academic learning model of Pintrich: Pintrich 's model, focuses on the integration of motivational and cognitive components. For them, the use of cognitive, meta-cognitive strategies promoted self-regulated learning [5].

Integral assessment model of Castañeda and Lopez: They propose a comprehensive evaluation model encourages the development of cognitive skills, affective, motivational and social, needed to the learner reach efficient learning, motivated, self-regulated and independent. Their model considers four types of learning strategies: 1) Information acquisition strategy, 2) Strategies for the recovery of learning, 3) Learning organizational strategies, critical and creative processing, and 4) Self-regulation strategies. The model incorporates two functions: motivation and strategic learning. Strategic learning is multidimensional, related the discipline learning, skills of self-regulation and mutual reinforcement among equals [6].

---

[1] http://uil.unesco.org/es/portal/areas-de-negocio/politicas-y-estrategias-de-aprendizaje-a-lo-largo-de-toda-la-vida/

Monereo model: according to Monereo, Pozo and Castelló [8], it is necessary to provide to learners of personal strategic resources also influence the curriculum, the organization of counselors and teachers to create contexts that promote the strategic use of knowledge. Monereo [9] presents interaction between contexts in strategic knowledge construction as shown in Figure 1.

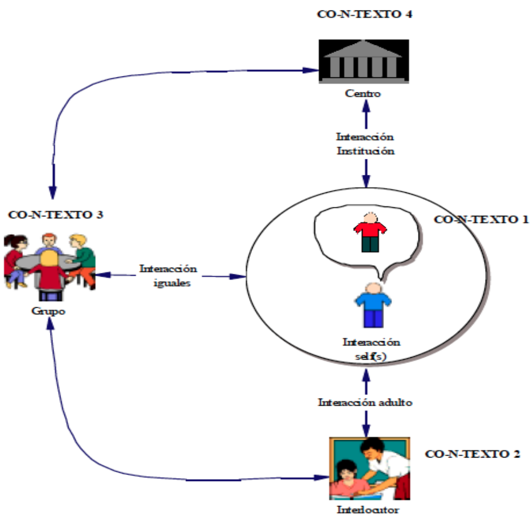


Fig. 1. Interaction between contexts [9]

The models and papers presented by various authors, considered as a key element of strategic learning: meta-cognition, self -regulation and motivation.

However, these models lack of contexts and strategic actions to facilitate the functionality of the learning regulation, do not define the actions to be performed to ensure that the learner reaches a strategic learning. Therefore, in the section 5 we propose a strategic learning meta-model.

3 Strategic Learning and Education Mediated by ICT

Strategic learning is based on the cognitive paradigm to recognize that the learner acquires not only information, but also learn cognitive strategies of two kinds: a) procedural cognitive strategies, to acquire, retrieve and use information; and b) the meta-cognitive strategies, associated with the reflection on their own learning processes [10]. The psychopedagogic proposed of strategic learning, whose principle is "learning to learn" and aims to transform the student into a strategic learner, self-regulated and reflective by Hernandez [11], cannot be excluded from the Information and Communications Technology (ICT). Therefore, the purpose of the pedagogical intervention in distance education is to develop in the learner the ability to perform meaningful learning alone in a new learning environment mediated by technology. (Cited by Rocha) [12].

Therefore, under the framework presented in this research, in Section 5 we propose a meta-model that defines the architecture of a platform to support distance education or part-time attendance under the strategic learning approach.

3.1 Diversity and evaluation, Key Factors in Strategic Learning

The evaluation process includes a set of didactic methods, is subjective and of multidimensional nature, occurs in different times and spaces, interactively involving the persons involved in the educational process, as proposed in concept of mediator evaluation Hoffmann and Anijovich in [13]. The mediator evaluation principles are: a) ethical principle of valuing differences, focusing on the idea that all students learn forever, b) teacher pedagogical principle of action research, which finds that students learn more if they have a better chance of learning, c) the provisional dialectical principle and complementary proposes significant learning for life [13].

In this paper we present a meta-model that implements the principles of mediator evaluation through an excellence monitoring cycle that creates a synergy in which it is possible to observe the student, identify recurring errors, and verifies the harmony in the collaborative environment mediated by Information and Communication Technologies [14].

4 Related Work

In the existing works have been several proposals for self-regulation of learning, as the DIDEPRO model [15] which is defined as last generation model, focused on the study of self-regulated learning, but from an interactive design and interdependent of teaching-learning process by using ICT. In addition there are experiences that incorporate new design schemes for self-regulation on e-learning in particular the work of Lee, Barker and Kumar [16] which describe the research done on the initial development of the e-learning model, instructional design framework, research design as well as issues relating to the implementation of such approach.

There are other works that highlight the importance ubiquitous learning such as the work of Joo and Park [17] in which it is proposed u-SM(Ubiquitous Scaffolding and Mentoring) teaching and learning model which applies the scaffolding and e-mentoring. In addition, it embodies and applies the designed u-SM model. Then it examines it affects in studying achievements and attitudes of students and verifies the application possibility of the u-SM teaching and learning.

There are also proposals ubiquitous learning, some as the work of Barbosa and others, [18] they presents the proposed GlobalEdu content management model, as well as its model of interoperability among repositories of learning objects that are used throughout the educational processes carried in the system.

5 The Proposed Model

The goal of the architecture of Strategic Learning Meta-Model (SLM) is to improve student performance, make strategic learners, self-regulated and self-reflective, encouraging learning through an educational environment that integrates psycho-pedagogical model, an ontological model and emerging technologies that enable ubiquity.

The strategic learning meta-model provides an architecture consisting of three layers: the reactive layer, intelligent layer and the infrastructure layer. The proposed meta-model is based on the principles of mediator evaluation described in section 3.1.

5.1 Architecture of the Meta-model for Strategic Learning

The SLM architecture integrates three layers: 1) the first layer is the reactive layer, consisting of a regulatory model, which aims to maintain the interaction between actors in different contexts which allow the regulation of learning until they self-regulate; 2) The second layer is the intelligent layer, integrated by an ontological model (ontologies set) that personalizes the student's learning activities, and 3) the infrastructure layer, which enables communication through various technologies, applications, devices and media, forming a Virtual Environment of Customized Learning (EVAP for its acronym in Spanish). Figure 2 shows the meta-model that integrates the layers constituting SLM architecture.

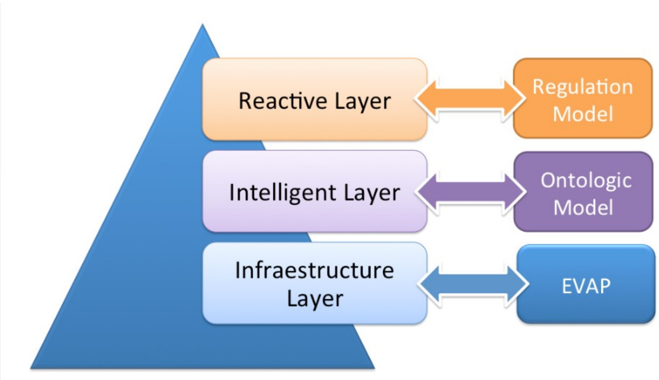


Fig. 2. Meta-model of the strategic learning

Each layer is formed of a particular model, which together can offer students an environment to achieve a strategic learning.

Reactive layer. The reactive layer within the SLM, contains the regulation model - which involves the excellence monitoring cycle. The aim of the reactive layer is to meet the requirements of the pedagogical principle of teaching action research, focused on providing better learning opportunities. This sets the interactions that occur in different contexts considering the reactive actions of the actors involved.

Intelligent layer. Intelligent layer of SLM is formed from an ontological model (ontologies set) , making up the domain of knowledge (general course, multimedia educational resources for self-study , multivariate learning activities) , and the student learning profile from the NLP theory and VARK of Fleming Neil and Mills Collen [19] and the neuroscience total brain theory of Ned Herrmman [20] .

Infrastructure layer. The infrastructure layer is responsible for enabling interface between the intelligent and the reactive layer. It aims to provide interfaces that allow users to connect to the system from any device (computer, laptop, tablet, ipad, cell phone, etc.), while automating the actions of the previous layers and provides ubiquity.

There are a considerable number of models and techniques within the SLM layers. However, for purposes of this article, only expose those built in the reactive layer and a brief definition, design and implementation of some ontologies of intelligent layer of meta-model proposed.

5.2 Regulation Model

The reactive layer within the SLM contains the regulation model. The regulation model that we propose incorporates some elements Monereo model presented in Section 2 and shown in Fig 1, which integrates the actors, contexts and their interaction. However, the Monereo model does not contemplate facilitators. Therefore, in the regulation model proposed, we consider the interaction with two facilitators who favor accompaniment of students and the motivation, which encourages strategic learning. The objective of the regulation model within the reactive layer is error detection (learning opportunities) and maintain the harmony of the educational environment through motivation, support, and self-knowledge of the student.

The interaction between contexts is the foundation of strategic learning as it integrates self-cognition, self-regulation, motivation, and cooperative learning. This is integrated into the excellence monitoring cycle that determines the interaction mechanisms between actors in the Architecture of Strategic Learning Meta-Model (SLM).

Regulation model components, constitute the elements of excellence monitoring cycle (as shown in Figure 3), consists of four actors: teachers, learners, facilitator A and facilitator B. Three of which are evaluators, and be evaluator is to know, understand, accept students into their own differences and their own learning strategies. The teacher is responsible for program evaluation activities, verify the results and re-plan the educational activity. The facilitator A identifies recurring errors of students to provide feedback to the teacher and retake the points that were not clear. The facilitator B is responsible for verifying harmony in the collaborative environment through the support and monitoring of the students progress. The answer of the learner is the starting point to identify strategies to improve learning. The interaction between all involved in the regulation model can learn while teaching and teach while you learn it, encouraging cooperation by sharing knowledge.

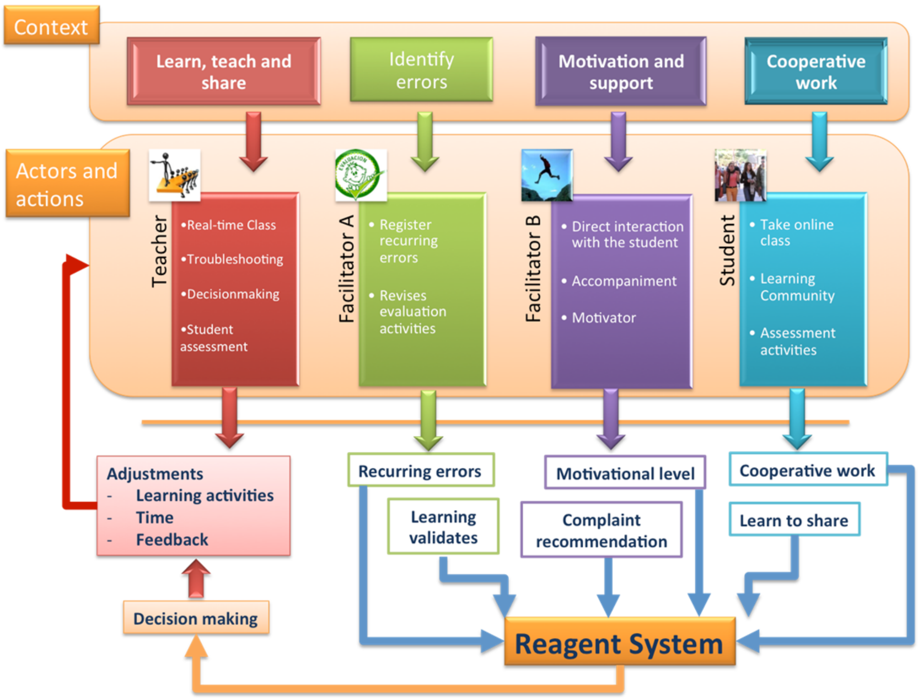


Fig. 3. Regulation model

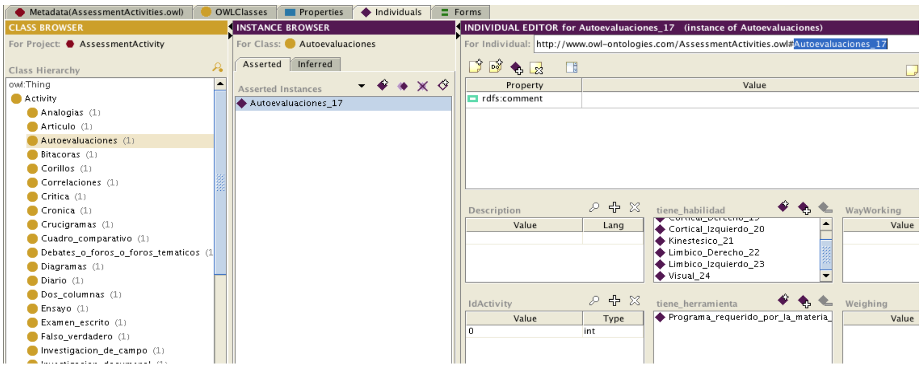


Fig. 4. Learning activities classified by the learner profile

5.3 Ontological Model

The ontological model is the intelligence of the system, through inference rules determines the customization of multimedia educational resources for self-study and learning activities according to the profile of each learner. The ontological model is composed of five ontologies, the top three are the profiles, courses and activities ontologies. The profiles ontology incorporates cognitive theories that will determine the learner's learning profile. The courses ontology, consider the instructional plan, multimedia educational resources for self-study and cognitive skills that students must develop. Finally, the activities learning ontology allows activities customization in accordance with learner's learning profile.

We show in Figure 4 the learning activities classified by the learner profile. Profiles, activities and courses ontologies were designed and implemented in Protègè, [21], as shown in Figure 5.

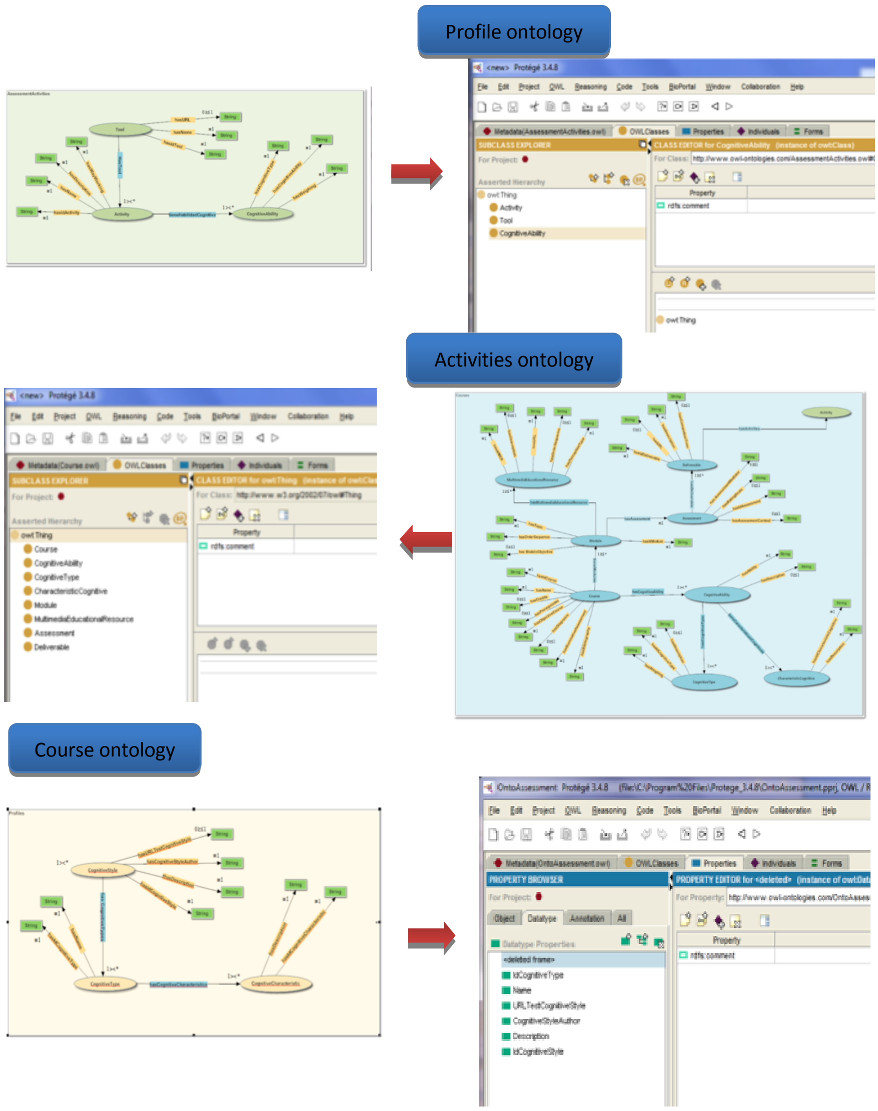


Fig. 5. Design and implementation of profiles, activities and courses ontologies

In Figure 6 we show the cognitive skills ontology considered in evaluation activities.

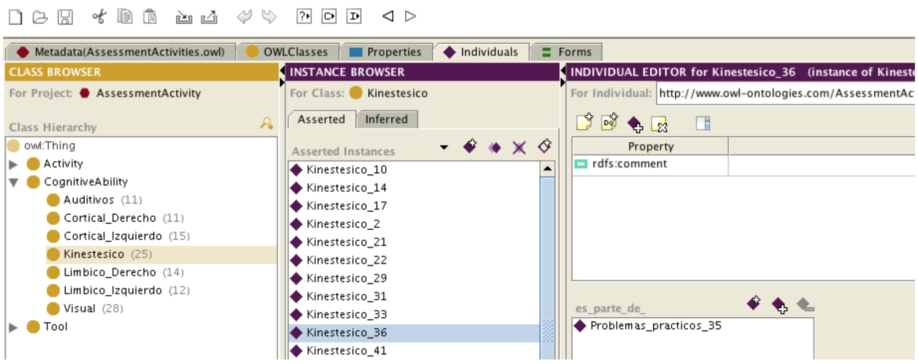


Fig. 6. Cognitive skills ontology considered in evaluation activities

6 Application Case

The first part of the proposed architecture (SLM), in particular the regulation model of the reactive layer, its performance was verified through an application case carried out during the trimesters 12P to 13I in groups with modality: Course No Presential (CNP), of Structured programming course (required course for engineering students) in the Autonomous Metropolitan University – Azcapotzalco (UAM-A).

6.1 Methodology

To address the problems described in Section 1 about failure and terminal efficiency, particularly in the UAM-A, were carried out several experiments. The design of experiments considered a set of variables that were modified to obtain good results in learning, reduced desertion and increased approval. Each experiment was performed in a different trimester, it was adjusting the variables involved in order to optimize the resources needed to address groups in CNP modality, and improve student learning.

6.2 Experiments

Table 2 shows the experiments and modifications to the variables involved. The sample size is approximately 100 students who enroll in part-time course contemplating a total of 500 engineering students who enroll in structured programming course per trimester. To calculate the sample size we use the following formula:

$$n = \frac{N\sigma^2 Z^2}{(N-1)e^2 + \sigma^2 Z^2}$$

For a population N = 500, 95% confidence Z = 1.96, and since there is no other values, then σ = 0.5 and e = 0.05. For N = 120, there is a minimum sample size n = 92.

The variables involved in the experiments are detailed in the Table 1.

Table 1. Variables for test cases

Variable	Values	Indicators
Assessment Activities	Unique for all Various Activities according to learning profile	% Approval
Creation of learning communities	Random Free choice of student according to their learning profile	% Desertion Integration
Monitoring the teaching – learning process	Teacher Hierarchical Excellence monitoring cycle	% Desertion Motivation

The experiments started from the trimester 12-P and conclude in 13-I, which allows 3 repetitions with different values in the variables involved, as shown in Table 2.

Table 2. Changing variables in the test cases of the trimester 12-P to the 13-I

	Cases		
Variable	**12-P**	**12-O**	**13-I**
Creation of learning communities	Free	Random	Combining thinking styles (diversity)
Assessment Activities	Mind Maps. Programs, Exams	Mind Maps, Self-Assessment , Exams, Programs	Self-Assessment, Exams, Programs
Monitoring	Excellence monitoring cycle	Excellence monitoring cycle	Excellence monitoring cycle
Cognitive tools	Virtual classroom	Collaboratory	Collaboratory

The first variable shown in Table 2 is the formation of learning communities, considering the free conformation to student's decision, random and according to their learning profile (maintaining diversity). The second variable is the assessment activities which have passed from homogeneous activities as mind maps, programs and exams to activities as self-assessment, programs and exams. The third variable is the monitoring and it has been maintained in this excellence monitoring cycle. Finally, the last variable is cognitive tools, which have passed from virtual classroom to collaboratories.

6.3 Analysis of Results

Structured programming course is part of the basic trunk so all engineering students must attend it. In this course the student is confronted with a new way of driving the teaching-learning process, understand the use of the platform, learn the subject discipline of course, develop critical thinking and abstract. This causes that students acquired new cognitive skills during the trimester.

Table 3 shows the results in each of the trimesters regarding the students' performance. Where attended students is equal to the number of students enrolled less truancy.

Table 3. Results in the test cases of the trimester 12-P to the 13-I

Trimester	Enrolled	Truancy	Attended	Approved	Performance
12-P	250	57	193	60	0.310880829
12-O	212	149	63	32	0.507936508
13-I	150	56	94	53	0.563829787

Figure 7 shows the results of average performance obtained in each experiment. The performance is calculated as the ratio of students approved between students attended. Clearly, the performance starts at 0.31 for the first experiment until the 0.56 in the third experiment, which shows a tendency to improve learning, applying the proposed model in Section 5.

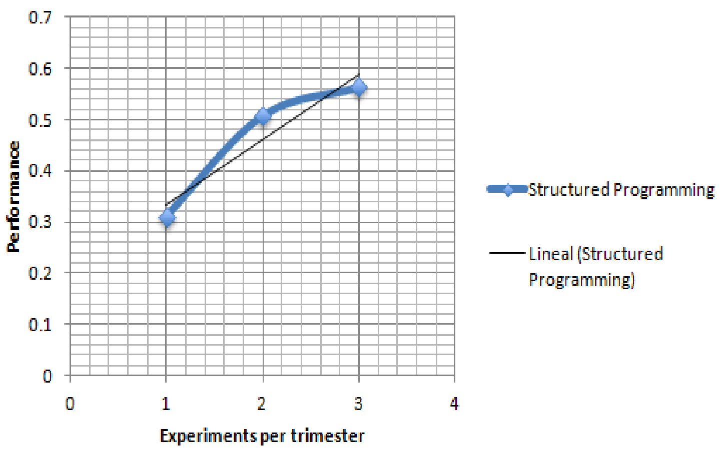


Fig. 7. Performance obtained in experiments

7 Conclusions

Based on the preliminary results of performance obtained shows a positive increase in the average performance of students who participated in the experiments, which covers part of the objective proposed in this research. Therefore, the Strategic Learning Meta-Model proposed in this paper is an alternative solution for problems such as: school failure, desertion, coverage and attention of students in massive groups because SLM optimizes both human and physical resources.

The backbone of the SLM is the regulation model of the reactive layer that gives life excellence monitoring cycle in which recurring errors are detected, maintaining a harmonious work environment and again provides personalized attention which in turn encourages learning communities work encouraging key values like a sharing knowledge, helping others to achieve success in community. The intelligent part of the SLM is the ontological model whose function is to recommend learning activities and tools that support the development of some cognitive abilities. This work is done manually on the platform from the provided recommendations. However the complete ontology is in construction and is part of future work to try automating the process through ontological model built.

The SLM potency the evaluation as a mechanism not only for change, but also for strategic learning. The evaluation, by customizing learning activities and self-regulation is an excellent learning strategy for both students and teachers. Evaluation becomes the main engine of a new culture of learning, enabling them to continue learning throughout life.

References

1. Jiménez M.G., Vega L., Bernal J.: La reprobación escolar como objeto de estudio de la investigación educativa. En: Primer Congreso Latinoamericano de Ciencias de la Educación (2010)
2. Baena, Paz G.: Trayectoria escolar en la licenciatura de ciencias políticas (4ta. Ed.). Estudios políticos. México. No. 7, Época, pp. 117−138 (1995)
3. Pozo J., & Monereo, C.: El apendizaje estratégico (1° edición). Aula XXI / Santillana. Madrid España (1999)
4. Weinstein, C., Powdril, L., Husman, J., Roska, L., Dierking, D.: Aprendizaje estratégico: un modelo conceptual, instruccional y de evaluación. In: S. Castañeda (coord.), *Evaluación y fomento del desarrollo intelectual en la enseñanza de ciencias, artes y técnicas.* México: Miguel Ángel Porrúa, pp. 197−228 (1998)
5. García, T. & Pintrich, P.R.: Regulating motivation and cognition in the classroom: the role of self-schemas and self-regulatory strategies. In: D.H. Schunk and B.J. Zimmerman (Eds.) Self-Regulation on Learning and Performance: Issues and Applications, NJ, Hillsdale, Lawrence Erlbaum Associates, pp. 132−157 (1994)
6. Castañeda, S. & López, M.: *La psicología cognitiva del aprendizaje, aprendiendo a aprender* [antología]. México: Facultad de Psicología, UNAM (1989)
7. Monereo, C.: De los procedimientos a las estrategias: implicaciones para el Proyecto Curricular Investigación y Renovación Escolar (IRES). Investigación en la escuela. 27, 21−38 (1995)

8. Monereo, C., Pozo, J.I., y Castelló, M.: La enseñanza de estrategias de aprendizaje en el contexto escolar. En: Coll, C.; Palacios, J; Marchesi, A. (Comps.). Desarrollo Psicológico y educación II (Edición revisada). Madrid: Alianza (2001)
9. Monereo, C.: Hacia un nuevo paradigma del aprendizaje estratégico: el papel de la mediación social, del self y de las emociones. Revista de investigación educativa, 5(3) 239−265 (2007)
10. Angulo, L.: Proyecto: educación en línea. Revista Electrónica Educare, vol. XIII, pp. 123−133 (2009)
11. Hernández, G.: Miradas constructivistas en psicología de la educación. México: Paidós Educador (2006)
12. Rocha, E.: Educación @ Distancia. Retos y Tendencias. Universidad Autónoma de Nuevo León. México: Arbor (2007)
13. Anijovich R., Cappelletti G., Hoffmann J. et al.: La evaluación significativa (1ª ed.). Buenos Aires: Paidós SAICF (2010)
14. UNESCO. Hacia las Sociedades del Conocimiento (2005)
15. De la Fuente, J. & Justicia, F.: El modelo DIDEPRO de regulación de la enseñanza y del aprendizaje: avances recientes. Revista electrónica de investigación psicoeducativa, 13, vol 5 (3) pp. 535−564 (2007)
16. Lee, S., Banker, T., & Kumar, V.: Learning preferences and self-regulation - Design of a learner-directed e-learning model. Communications in Computer and Information Science 257 CCIS, pp. 579−589 (2011)
17. Joo, K. & Park, N.: Design and application of the u-SM teaching and learning model for an efficient ubiquitous learning. In: 2013 International Conference on Computing, Management and Telecommunications, ComManTel 2013; Ho Chi Minh City; Viet Nam; 21- 24 January 2013, Article number 6482402, pp. 264−268 (2013)
18. Barbosa, D., Barbosa, J., Bassani, P., Rosa,J., Martinis, M.: Content management in a ubiquitous learning environment. International Journal of Computer Applications in Technology, 46(1), pp. 24−35 (2013)
19. Fleming N. & Mills C.: Not Another Inventory, Rather a Catalyst for Reflection. To improve the Academy, Vol. 11, p. 137 (1992)
20. Herrmann, N.: The creative brain. Búfalo: Brain books (1989)
21. Protégé 3.0. http://protege.stanford.edu/

Reviewing Committee

(Comité de revisión del volumen)

Fernando Aldana
Sivaji Bandyopadhyay
María Lucía Barrón Estrada
Bert Bredeweg
Hiram Calvo
Alejandro Canales Cruz
Niladri Chatterjee
Erik Cuevas
Iria da Cunha
Asif Ekbal
Andrea Formisano
Claude Frasson
Sofia N. Galicia-Haro
Alexander Gelbukh
Pilar Gomez-Gil
Monique Grandbastien
Christian Grimme
Yasunari Harada
Donato Hernandez Fusilier
María Yasmín Hernández Pérez
Dieter Hutter
Laetitia Jourdan
Pinar Karagoz
Bruno Lara
Mikhail Mikhailov
Raul Monroy
Rafael Morales Gamboa
Jaime Muñoz Arteaga
Masaki Murata
Juan Carlos Nieves
Juan Arturo Nolazco Flores
Felipe Orihuela-Espina
Héctor Pérez-Urbina
David Pinto
Carlos Adolfo Piña-García
Natalia Ponomareva
Marco Antonio Ramos Corchado
Carlos A. Reyes García
Horacio Rodriguez
Guillermo Rodríguez Ortiz
Víctor Germán Sánchez Arias
Abraham Sánchez López
Jose Santos
Manuel Vilares Ferro
Ramón Zatarain Cabada

Additional reviewers

Roberto Alonso
Miguel Ballesteros
Santiago E. Conant-Pablos
Victor Darriba
Esteban Guerrero
Alev Mutlu
Hugo Terashima
Santanu Pal
Soujanya Poria
Daisuke Yokomori

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
noviembre de 2013
Printing 500 / Edición 500 ejemplares