An Automatic Text Comprehension Classifier Based on Mental Models and Latent Semantic Features

Felipe Bravo-Marquez
University of Chile
República 701
Santiago, Chile
fbravo@dcc.uchile.cl

Gaston L'Huillier
University of Chile
República 701
Santiago, Chile
glhuilli@dii.uchile.cl

Patricio Moya
University of Chile
República 701
Santiago, Chile
pmoyam@gmail.com

Sebastián A. Ríos
University of Chile
República 701
Santiago, Chile
srios@dii.uchile.cl

Juan D. Velásquez
University of Chile
República 701
Santiago, Chile
jvelasqu@dii.uchile.cl

ABSTRACT

Reading comprehension is one of the main concerns for educational institutions, as it forges the students' ability to comprehend and learn accurately a given information source (e.g. textbooks, articles, papers, etc.). However, there are few approaches that integrates digital sources of educational information with automated systems to detect whether an individual has comprehended a given reading task. This work main contribution is a text comprehension classification methodology for the detection of reading comprehension failures in educational institutions. The proposed approach relates situational model theories and latent semantic analysis from fields of psycholinguistics and natural language processing respectively. A numerical characterization of students' documents using structural information, such as the usage of text connectors, and latent semantic features are used as input for traditional classification algorithms. Therefore, an automated classifier is built to determine whether a given student could or not comprehend the information in the given stimulus documents. For the evaluation of the proposed methodology, using a set of stimulus documents, a set of questions must be answered by an experimental group of students. We have performed experiments using first year students from Engineering and Linguistics undergraduate schools at the University of Chile with promising results.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data Mining*; J.5.5 [**Computer Applications**]: Arts and Humanities—*Linguistics*

General Terms

Text Mining, Text Comprehension Evaluation

Keywords

Text Comprehension, Situational Models, Latent Semantic Analysis, Classification

1. INTRODUCTION

Considering that reading comprehension, learning, listening skills, among other cognitive properties, are strongly related [15], evaluating the reading comprehension of students has become an important task in educational establishments. Moreover, students with learning failures can be identified and actions to support them can be developed accordingly. This task can be improved by using advanced analytical tools on digital documents delivered by students.

Nowadays, a common comprehension measuring instrument consists in giving students a reading based assignment, using a set of sources (papers, books, articles, etc.) about a specific common topic (*stimulus documents*). Then, students need to answer some questions oriented to integrate information from these sources. Afterwards, assignments are reviewed and scored by experts (e.g. teachers) according to the level of comprehension and integration detected in the responses. Nevertheless, if previous evaluation is needed to be implemented in a large scale number of students, the amount of resources needed could be significant. An automatic measuring instrument could help substantially to evaluate a large number of students.

The main contribution of this work is an automatic text comprehension detection system, inspired in the situational model theory [11] and latent semantic analysis (LSA) [7]. Our proposal is to compute structural and linguistic features from students' documents, such as latent semantic features obtained after considering both students' and stimulus documents. Then, using machine learning algorithms, a classification hypothesis based on extracted features is proposed. The effectiveness of the proposed classifier is based on a linguistic evaluation instrument, which considers a set of both

.

stimulus source documents and questions which integrates all source documents' information.

This paper is organized as follows. First, Section 2 presents related work in text comprehension and automated text analysis techniques. In Section 3, the proposed methodology for document characterization and evaluation is introduced. Then, in Section 4, the experimental setup is detailed and their results discussed. Finally, in Section 5, main conclusions and future work are presented.

2. RELATED WORK

Whenever someone reads a document, he or she develops a mental representation of what is been read. According to Kintsch [11], this representation is developed following the objectives of each reader with respect to the source document. In this sense, someone who understands a text produces a *situational model*.

DEFINITION 1. ***Situational Model***
A representation created from their context and experiences stored in long-term memory which leads to a overall understanding and learning of an given document [11].

This aspect of understanding and learning is very important for educational institutions, since its goal is often to lead to learning by using textbooks and reading assignments.

DEFINITION 2. ***Learn from text***
To construct a situational model *from a source document that will be remembered and used effectively when required in later events [13].*

However, most of learning tasks in educational institutions involve students in reading two or more texts. Given this, it is considered that the reader must be able to represent and integrate many *situational models* adequately, task which is not fulfilled by most students [17].

To determine if *situational models* derived after reading multiple source documents are appropriate, researchers have been using manual analysis of documents built by test subjects (e.g. essays composed by sets of students using multiple sources). Nevertheless, this process is expensive in time and human resources, with difficulties to extend the analysis for large documents. Recently, researchers have been able to improve their methodologies using computer science approaches [8, 11].

As described in [11], LSA has been used as a technique to describe the acquisition and usage of knowledge. Furthermore, in order to make this analysis when information across texts is combined with previous knowledge, LSA has been proposed as captures the integration of information, representing concepts in a semantic space, in which vector similarity between concepts represents a characterization of semantic relatedness [8].

Several studies have been concerned in determining the *situational models* developed by students when reading multiples sources. A first approach, developed by [8], uses LSA on documents created on history courses. Results indicates that LSA captures a deep association structure between concepts, similar to the reader's situational model of texts. A second approach [1], attempt to simulate the process by which humans comprehend texts. In this case, results indicate that LSA model can be used to provide a good semantic representation of a predicate-argument expression.

The basic LSA representation does not make any distinction between the order of words (e.g. $w_i w_j$ and $w_j w_i$, for given two words w_i and w_j). When considering the semantic representation of documents this could be considered. Also, Kintsch in [12] suggested a complex representation of a document by a network composed of the predicate, the argument, and a fixed number of neighbor terms of the predicate. Finally, Rus et al. [19], proposed several methods to automatically detect students' situational models using an e-learning tool, which evaluates reading comprehension using source documents about the same topics.

3. PROPOSED METHODOLOGY

In this section, the proposed methodology and all of its components are introduced. First, the basic notation and all techniques for document analysis and characterization are discussed. Then, all a short introduction to machine learning classification algorithms and their main characteristics are presented. Finally, the overall methodology for *situational model* representation is detailed.

3.1 Basic Notation and Document Analysis

Let us introduce some concepts. In the following, let $\mathcal{V}$ a vector of words that defines the vocabulary to be used. We will refer to a word w, as a basic unit of discrete data, indexed by $\{1, \ldots, |\mathcal{V}|\}$. A document is a sequence of S words defined by $\mathbf{w} = (w^1, \ldots, w^S)$, where w^s represents the s^{th} word in the document. Finally, a corpus is defined by a collection of $\mathcal{D}$ documents denoted by $\mathcal{C} = (\mathbf{w}_1, \ldots, \mathbf{w}_{|\mathcal{D}|})$.

A vectorial representation of the documents corpus is given by $\texttt{TF-IDF} = (m_{ij}), i \in \{1, \ldots, |\mathcal{V}|\}$ and $j \in \{1, \ldots, |\mathcal{D}|\}$, where m_{ij} is the weight associated to whether a given word is more important than another one in a document. The m_{ij} weights considered in this research is defined as the *tf-idf* term [20] (*term frequency times inverse document frequency*), defined by

$$m_{ij} = \frac{n_{ij}}{\sum_{k=1}^{|\mathcal{V}|} n_{kj}} \times log\left(\frac{|\mathcal{C}|}{n_i}\right) \tag{1}$$

where n_{ij} is the frequency of the i^{th} word in the j^{th} document and n_i is the number of documents containing word i. The *tf-idf* term is a weighted representation of the importance of a given word in a document that belongs to a collection of documents. The *term frequency* (TF) indicates the weight of each word in a document, while the *inverse document frequency* (IDF) states whether the word is frequent or uncommon in the document, setting a lower or higher weight respectively.

The common similarity measure used in information retrieval is the cosine of the angle between vectors presented in Equation 2.

$$\cos(\angle(\mathbf{w}_i, \mathbf{w}_j)) = \frac{\sum_{k=1}^{|\mathcal{V}|} m_{ki} \cdot m_{kj}}{\sqrt{\sum_{k=1}^{|\mathcal{V}|}(m_{ki})^2}\sqrt{\sum_{k=1}^{|\mathcal{V}|}(m_{kj})^2}} \tag{2}$$

Furthermore, considering that an n-gram is a sequence of n contiguous words in a document, $cos_n : \mathbb{R}^{|\mathbf{w}_1|} \times \mathbb{R}^{|\mathbf{w}_2|} \rightarrow \mathbb{R}$ is defined extending definition for Equation 2. This is achieved by computing the distance with respect to a vectorial representation of input documents $\mathbf{w}_1$ and $\mathbf{w}_2$ n-grams. This representation could be, for example, defined as Equation 1, where the weights could be computed using n-grams instead of words.

3.1.1 Latent Semantic Analysis

Using the `TF-IDF` matrix [20], its singular value decomposition (SVD) reduces the dimensions of the term by document space. SVD considers a new representation of the feature space, where the underlying semantic relationship between terms and documents is revealed. Let matrix M be an $|\mathcal{V}| \times |\mathcal{M}|$ `TF-IDF` representation of documents and r the low rank of matrix M, an appropriate number for the dimensionality reduction and term projection [7].

Given, $U_r = (u_1, \cdots, u_r)$ an $|\mathcal{V}| \times r$ matrix, the singular values matrix $\mathcal{D}_r = diag(d_1, \cdots, d_r)$, where $\{d_i\}_{i=1}^{r}$ represents the roots of the eigenvalues of MM^{T} or $M^{\mathrm{T}}M$, and $V_r = (v_1, \cdots, v_r)$ an $|\mathcal{M}| \times r$ matrix, then the SVD decomposition of M is represented by,

$$M = U_r \cdot \mathcal{D}_r \cdot V_r^{\mathrm{T}} \tag{3}$$

As described by [16], SVD preserves the relative distances in the vector space model matrix (e.g. `TF-IDF`), while projecting it into a semantic space model, which has a lower dimensionality. Similar to what principal component analysis (PCA) achieves by projecting features into its principal components, allowing to keep the minimum information needed to an appropriate representation of the dataset.

LSA consider co-occurrence of terms in different documents, where semantic similarities between terms and documents can be computed. In [5], for plagiarism detection, the matrix decomposition is applied over a n-gram-document matrix.

3.1.2 Latent Semantic Features

Latent semantic features are attributes extracted from assignments written by students, which aim to model semantic similarities between assignments and stimulus documents together with the level of information integration achieved.

In order to extract the latent semantic features, the corpus must be processed removing stopwords and stemming words to their root. Then, several n-grams-documents matrices are created using different values of n, where a 1-gram matrix has terms rows, the 2-grams matrix has pair of contiguous words rows, and so on. Moreover, each cell in the matrix has as value the number of occurrences of the n-gram in the document. Furthermore, as proposed in [5], for each n-gram where $n > 2$, the terms within it are sorted alphabetically in order to increase frequency of n-grams in different documents, combining similar concepts in one single dimension. In order to speed the singular value decomposition of matrices, n-grams with frequency less to 2 in the corpus were not considered as dimensions.

Once we have the n-gram-Document matrices M_n, for each of them, a SVD is applied and low rank approximations matrices are created. Obtaining therefore, a set of low dimensional document matrices V_n. Then, as columns from V are concept space representations of each document within the corpus, for each V_i matrix the cosine similarity (Equation 2) from the vector space model between students documents columns and stimulus columns are computed.

In this work, we assume that comprehension level in students is strongly related with the semantic similarity between their responses and what they have read together with the ability to integrate concepts from multiple sources. Considering that for each student document $\mathbf{w}$ a set of m stimulus documents $\mathcal{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_m\}$ has been read, we propose the following LSA feature given a n-gram representation n:

$$LSAF_n(\mathbf{w}, \mathcal{S}) = \frac{1}{m} \cdot \sum_{\mathbf{s}_i \in \mathcal{S}} cos_n(\mathbf{w}, \mathbf{s}_i) \times \phi_n(\mathbf{w}, \mathcal{S}) \tag{4}$$

$$\phi_n(\mathbf{w}, \mathcal{S}) = \sum_{\mathbf{s}_i \in \mathcal{S}} \sum_{\mathbf{s}_j \in \mathcal{S}, j=i+1} \frac{\min\left\{ \frac{cos_n(\mathbf{w},\mathbf{s}_i)}{cos_n(\mathbf{w},\mathbf{s}_j)}, \frac{cos(\mathbf{w},\mathbf{s}j)}{cos(\mathbf{w},\mathbf{s}_i)} \right\}}{\frac{m(m-1)}{2}} \tag{5}$$

The left part of the product represents the average cosine similarity with the stimulus documents, and the term $\phi : \mathbb{R}^{|V|} \times \mathbb{R}^m \rightarrow \mathbb{R}$, inspired in the similarity measure proposed in [22], represents a balance of similarities, where the level of integration achieved by the student is captured.

It is possible to compute several $LSAF_n$ values for a document $\mathbf{w}$ using different n-gram representations. Also, it is important to consider, that as smaller is the value of n, conceptual similarities between documents are captured. Likewise, as higher is the value, syntactic information is captured [19], and issues like plagiarism or quoting can be detected [5].

3.1.3 Structural Features: Sentence Connectors

The second type of features consists in counting the occurrence of sentence connectors in delivered documents, since these components characterizes logical-semantic relations, which may be related with the level of comprehension achieved [25]. In the following, three categories of connectors are described [4]:

1. **Temporal connectors**: Introducing temporal relationships between sentences. Narrative texts (i.e. stories, novels) have a lot of temporal connectors, since they refer to events that happen in a literary space.
2. **Causal and consecutive connectors**: Introducing the cause or consequence between textual segments. Expository texts (i.e. descriptions, reports) used a large number of such connector to explain the theme or topic of each document.
3. **Contra-argumentative connectors**: Introduces a shift in the opposite direction to the segment immediately preceding, in whole or in part. Appropriate argumentative texts (i.e. essays) employ a lot of these connectors to indicate the different positions presented.

For each document, these features, described and listed in [9], were extracted considering their number of occurrences. Then, each value is calculated as the min max normalized frequency of connectors from a particular type in the document.

3.2 Document Classification

Once the document collection is processed, LSA and connector features, together with a target binary label assigned by experts, are used to build a training dataset from documents delivered by students. Afterwards, several machine learning algorithms like SVMs [21], artificial neural networks [23], among others classification techniques, can be used to train a classifier. Finally, the best classifier obtained using proposed features, can be used to classify large collections of documents without requiring further human intervention.

In the following, let consider a dataset $(\mathcal{X}, \mathcal{Y})$, where objects $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ are characterized by a feature set $\mathcal{A}$, for which $\mathbf{x}_i = \{x_{1,1}, \ldots, x_{1,|\mathcal{A}|}\}$, and target label $\mathcal{Y} = \{y_1, \ldots, y_N\}$, are determined by binary values $y_i \in \{+1, -1\}$, $\forall i \in \{1, \ldots, N\}$.

3.2.1 Support Vector Machines Classifier

The main idea of SVMs [3, 21] is to find the optimal hyperplane that separates objects belonging to two classes ($y_i \in \{+1, -1\}$) in a feature space $\mathcal{X}$, maximizing the margin between these classes. This feature space is considered to be a Hilbert space defined by a dot product, known as the kernel function, $K(\mathbf{x}, \mathbf{x}') = (\phi(\mathbf{x}) \cdot \phi(\mathbf{x}^{\mathrm{T}}))$, where $\phi : \mathcal{A} \rightarrow \mathcal{X}$, is the mapping defined to translate an input vector into the feature space. The objective of the SVM algorithm is to find the optimal hyperplane $\omega^{\mathrm{T}} \cdot \mathbf{x} + b$, defined by the following optimization problem,

$$\begin{aligned} \min_{\omega,\xi,b} \quad & \frac{1}{2}\sum_{j=1}^{|\mathcal{A}|}\omega_j^2 + C\sum_{i=1}^{N}\xi_i \\ \text{subject to} \quad & y_i\left(\omega^T\mathbf{x}_i + b\right) \geq 1 - \xi_i \ \forall i \in \{1, \ldots, N\} \\ & \xi_i \geq 0 \ \forall i \in \{1, \ldots, N\} \end{aligned} \tag{6}$$

The objective function minimizes errors $\sum_i^N \xi_i$, while obtaining the maximum margin hyperplane, adjusted by regularization parameter C. Its dual formulation is defined by the following expression,

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j \cdot K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{subject to} \quad & \alpha_i \geq 0, \forall i \in \{1, \ldots, N\} \\ & \sum_{i=1}^{N}\alpha_i y_i = 0 \end{aligned} \tag{7}$$

Finally, after determining the optimal dual parameters α, for a given document $\mathbf{x}_j$, its classification is determined by,

$$\mathcal{C}(\mathbf{x}_j) = sign\left(g(\mathbf{x}_j)\right), \text{ where } g\left(\mathbf{x}_j\right) = \sum_{i=1}^{N}\alpha_i y_i \cdot K(\mathbf{x}_i, \mathbf{x}_j) + b$$

3.2.2 Logistic Regression

The idea of this classification algorithm is to determine for a given object $\mathbf{x}_i$, the posterior probabilities of label y_i. For this, logistic regression estimates parameters $\beta_{\mathcal{L}}$ and $\beta_{\mathcal{T}}$ for a linear regression over the set of features, which is mapped by approximating a function into the interval $[0, 1]$. In general, the model has the form,

$$\mathcal{C}(\mathbf{x}_i) := P(y_i|\mathbf{x}_i) = \frac{exp(\beta_{\mathcal{T}} + \beta_{\mathcal{L}}^T \cdot \mathbf{x}_j)}{1 + exp(\beta_{\mathcal{T}} + \beta_{\mathcal{L}}^T \cdot \mathbf{x}_i)} \tag{8}$$

Parameters $\beta_{\mathcal{L}}$ are determined either by maximizing the conditional likelihood on the training set or by minimizing the class-loss over the training set [10].

3.2.3 Decision Trees

A Decision Tree is a discriminant classifier represented by a tree data structure. Each node from the tree corresponds to a feature, branches are conditions on the father node, and leafs nodes are assigned to label values.

Trees are constructed by repeated splits of subsets of data based on the selection of features. For several algorithms used to generate decision trees from data (e.g. ID3 [18]), the common criterion used to select the attribute for splitting the data at each node is the information gain criteria. This criteria is based on the concept of entropy that comes from information theory. For further information on these classification algorithms, please refer to [2].

3.2.4 Evaluation Criteria

As presented in Table 1, the resulting confusion matrix of the classification task can be described using four possible outcomes: Correctly classified documents which achieved the situational model or True Positives (TP), correctly classified documents without the situational model or True Negative (TN), wrongly classified documents without the situational model as documents which achieved the situational model or False Positive (FP), and wrongly classified documents which achieved the situational model as documents without it, or False Negative (FN).

Table 1: Confusion Matrix for binary classification problems.

	$y = +1$	$y = -1$
$\mathcal{C}(\mathbf{x}) = +1$	TP	FP
$\mathcal{C}(\mathbf{x}) = -1$	FN	TN

The following evaluation criteria are common machine learning measures, which are constructed using the before mentioned classification outcomes.

- Precision, that states the degree in which documents identified as positive indeed achieved a situational model.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{9}$$

- Recall, that states the percentage of delivered documents that the classifier manages to classify correctly. Can be interpreted as the classifier's effectiveness.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{10}$$

- F-measure, the harmonic mean between the precision and recall

$$\text{F-measure} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (11)$$

- Accuracy, the overall percentage of correctly classified documents.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (12)$$

3.3 Building a Text Comprehension Classifier

The proposed methodology is described as follows:

1. Students are asked to answer a set of questions using *stimulus documents* $\mathcal{S}$ as sources (e.g. papers, articles, textbooks, etc.).
2. Using both $\mathcal{S}$ and documents delivered by students $\mathcal{D}$, a document corpus is built and processed.
3. The extraction of latent semantic features, described in Section 3.1.2, and structural features, described in Section 3.1.3 is performed. This is executed in order to extract information about the latent semantic similarities between students documents $\mathcal{D}$ and stimulus documents $\mathcal{S}$, and the usage of sentence connectors.
4. Afterwards, students' documents are labeled by experts according to a binary representation of the comprehension level achieved. Together with all extracted features are used to build a training/testing dataset.
5. Then, different machine learning classification techniques are trained using the training dataset, and evaluated over the test dataset.
6. The classification hypothesis to determine whether a student achieves a good situational model, is determined by using the best algorithm in previous step.

4. EXPERIMENTAL SETUP AND RESULTS

A real world implementation of the proposed methodology was applied to first year students from the careers of engineering and linguistics at the University of Chile. The aim of this experiment is to explore the feasibility, utility, and potential of the proposed methodology in educational establishments.

In the following, the experiment instrument design, the evaluation context, and experimental results are presented. In terms of experimental results, both evaluation criteria for classification algorithms, and an exploratory analysis for closeness of documents are depicted in order to analyze the potential of the proposal.

4.1 Experiment Instrument

To evaluate the proposal, it is necessary to ensure at least two input stimulus documents for the development of the task. These stimulus documents must fulfill the following set of attributes:

- Refer to the same topic.
- Have the presence of contradictory or complementary information.
- Present a similar extension.
- Must be, necessarily, expository or argumentative.

The writing task, derived from the reading of all stimulus documents, should be able to integrate the information from sources. Also, it must be well suited for learning, considering specially the age and educational environment where test students are involved. Thereby, the set of questions to be answered must be created considering argumentative and global aspects, since promote an deep understanding and learning [24].

Finally, all answers and students' writings are validated by external experts, whose label is $y = +1$ is the document achieved the situational model, or $y = -1$ otherwise.

To achieve the situation model, extending Definition 1 in Section 2, students must integrate information from different sources appropriately concerning the task requirements.

4.2 Building and Processing the Dataset

Assignments for engineering and linguistics students were developed considering properties described in Section 4.1. Each assignment consisted using two stimulus documents on topics related to their undergraduate courses, and then answering 3 information integration questions, which needs the usage of both stimulus documents. Afterwards, assignments were answered and delivered by students in a digital format. A total of 204 delivered documents were reviewed by two experts, where comprehension achievement level was evaluated and classified as defined in Section 4.1.

In order to process the situational model corpus, a document processing tool was developed using the Java programming language (`JDK 1.6.0`). Given that all documents where in Spanish, a Spanish list of stopwords was elaborated and the Snowball[1] stemmer was used in the text processing procedure. Furthermore, a n-gram-document matrix builder was developed, and using the SVD decomposition module provided by the Java Matrix Package[2], the proposed LSA features were computed.

All documents were processed using the software described above, where for each delivered document in the corpus, connectors features and LSA (uni-grams and bi-grams) features were extracted.

4.3 Classification Results

Using previously stated dataset, a 10×10 cross-validation evaluation procedure was used in order to train and evaluate the classification performance of different classification algorithms. SVMs where evaluated using `LIBSVM` [6], using a grid-search procedure, an radial basis function (RBF) Kernel was evaluated using $\gamma = 0.9$ and the regularization parameter was considered as $C = 1000$. Logistic regression, as well as decision trees, where developed using `RapidMiner v.5.0` software [14], where the implementation for logistic regression is non-parametric, and decision trees were evaluated

[1] `http://snowball.tartarus.org/` [online: accessed 05-05-2011]

[2] `http://math.nist.gov/javanumerics/jama/` [online: accessed 05-05-2011]

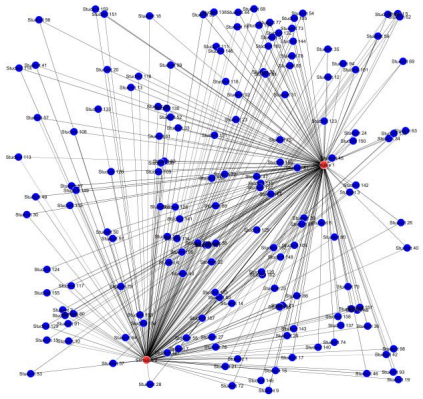

Figure 1: Engineering students' documents and their closeness with source documents.

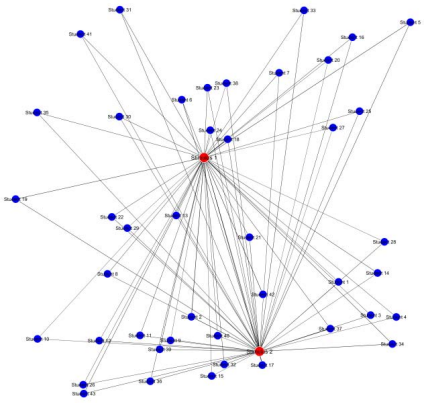

Figure 2: Linguistic students' documents and their closeness with source documents.

using information gain as partition criteria, with 4 as a minimum number of objects in the split, 4 as the minimum size of nodes, 0.01 as minimum information gain criteria to stop branching, and a confidence for the hypothesis testing in evaluating splits of 0.02.

Table 2 shows results obtained from a 10×10 cross-validation evaluation. Results shows that SVMs and decision trees outperforms logistic regression classification algorithm. Also, despite SVMs presents a highest accuracy, decision trees achieves better F-measure results, which indicates that the classifier is more stable, and achieves better results on both recall and precision.
An interesting insight revealed from trained decision trees is that for all cases where $LSAF_1$ value was smaller than 0.26 the student did not achieve the situational model. Moreover, by analyzing well classified class 0 documents, we realize that these documents achieve the lowest level of comprehension within the collection according to experts who reviewed all documents.

4.4 Exploratory Analysis for Closeness of Documents

Using cosine similarities from the uni-gram LSA space in the corpus, a network which represents the closeness between all documents and their stimulus was built for engineering students (Figure 1), and linguistics students (Figure 2). Stimulus and delivered documents were defined as red and blue vertexes respectively, and cosine similarities between documents were used to weight their respective edges.

From networks presented, it's easy to see how stimulus documents are centrally located and the interaction between all documents is uniformly distributed over the canvas.

5. CONCLUSIONS AND FUTURE WORK

Results obtained allow us to validate that LSA is an adequate technique for analyzing the degree of text comprehension by from students, by evaluating how they build situational models. Furthermore, we can conclude that our methodology was able to classify, with satisfactory results, when a student has not achieve the situation model. Hence, weak students can be identified automatically and the level of several educational establishments could be evaluated and compared.

We can also conclude, that LSA features extracted from documents are able to represent the situation model in a proper manner. This is a novel quantitative measure which combines comprehension and the integration of multiple stimulus documents.

As future work, the model could be extended into a large scale processing environment where teachers can submit students assignments and obtain comprehension measures automatically. Nevertheless, in order to improve the accuracy of the model, a largest training dataset is required. In this case, supervised properties could be determined by using *Amazon Mechanical Turk*[3], or similar large scale crowd intelligence mechanisms.

6. ACKNOWLEDGMENT

Authors would like to thank continuous support of "Instituto Sistemas Complejos de Ingeniería" (ICM: P-05-004- F, CONICYT: FBO16; `www.isci.cl`); and FONDEF project (DO8I-1015) entitled, DOCODE: Document Copy Detection (`www.docode.cl`).

7. REFERENCES

[1] Benoît Lemaire, Guy Denhière, Cédrick Bellissens, and Sandra Jhean. A Computational Model for Simulating Text Comprehension. *Behavior Research Methods*, 38(4):628–637, 2006.

[2] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[3] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *COLT '92: Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, New York, NY, USA, 1992. ACM Press.

[4] H. Calsamiglia and A. Tusón. *Las cosas del decir. Manual de Análisis del Discurso*. Ariel, 2007.

[3] `https://www.mturk.com` [online: accessed 05-05-2011]

Table 2: Accuracy, Recall, Precision, and F-Measure for all classification algorithms.

Model	Accuracy	Recall	Precision	F-Measure
SVM	0,7712 ± 0,13	0,7186 ± 0,14	0,7443 ± 0,13	0,7255 ± 0,12
Logistic Regression	0,5919 ± 0,85	0,6045 ± 0,13	0,6899 ± 0,78	0,6045 ± 0,13
Decision Tree	0,6129 ± 0,32	0,9288 ± 0,64	0,6200 ± 0,01	0,7455 ± 0,24

[5] Z. Ceska. Plagiarism detection based on singular value decomposition. In *Proceedings of the 6th international conference on Advances in Natural Language Processing*, GoTAL '08, pages 108–119, Berlin, Heidelberg, 2008. Springer-Verlag.

[6] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[7] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[8] P. W. Foltz, M. A. Britt, and C. A. Perfetti. Reasoning from multiple texts: An automatic analysis of readers' situation models. In *Proceedings of the 18th Annual Cognitive Science Conference*, pages 110–115, NJ, USA, 1996. Lawrence Erlbaum Associates.

[9] M. N. D. García. *Conectores discursivos en textos argumentativos breves*. Arco Libros, Madrid, España, 2007.

[10] J. Hilbe. *Logistic Regression Models*. Chapman & Hall/CRC Press, 2009.

[11] W. Kintsch. *Comprehension*. Cambridge University Press, 1998.

[12] W. Kintsch. Metaphor comprehension: a computational theory. *Psychon. Bull. Rev.*, 7(2):257–266, June 2000.

[13] W. Kintsch and E. Kintsch. *Children's Reading Comprehension and Assessment*, chapter Comprehension, pages 71–92. Lawrence Erlbaum Associates, 2005.

[14] I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz, and T. Euler. Yale: Rapid prototyping for complex data mining tasks. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 935–940, New York, NY, USA, 2006. ACM.

[15] R. Moreno and R. E. Mayer. Verbal redundancy in multimedia learning: When reading helps listening. *Journal of Educational Psychology*, 94(1):156 – 163, 2002.

[16] C. H. Papadimitriou, H. Tamaki, P. Raghavan, and S. Vempala. Latent semantic indexing: a probabilistic analysis. In *PODS '98: Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 159–168, New York, NY, USA, 1998. ACM.

[17] C. A. Perfetti, J. Rouet, and M. A. Britt. *The Construction of Mental Representations During Reading*, chapter Toward a Theory of Documents Representation, pages 99–122. Lawrence Erlbaum Associates, 1999.

[18] J. R. Quinlan. Induction of decision trees. *Mach. Learn.*, 1:81–106, March 1986.

[19] V. Rus, M. C. Lintean, and R. Azevedo. Automatic detection of student mental models during prior knowledge activation in metatutor. In *EDM '09: Proceedings of the 2nd International conference on Educational Data Mining*, pages 161–170, 2009.

[20] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18:613–620, 1975.

[21] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1999.

[22] J. D. Velásquez, H. Yasuda, T. Aoki, and R. Weber. Using the kdd process to support web site reconfigurations. In *WI' 2003: Proceedings of the IEEE/WIC International Conference on Web Intelligence*, pages 511–515, 2003.

[23] P. J. Werbos. *The roots of backpropagation: from ordered derivatives to neural networks and political forecasting*. Wiley-Interscience, New York, NY, USA, 1994.

[24] J. Wiley and J. Voss. Contructing arguments from multiple sources: tasks that promote understanding and not just memory for text. *Journal of Educational Psychology*, 91:301–311, 1999.

[25] R. A. Zwaan and G. A. Radvansky. Situation models in language comprehension and memory. *Psychological Bulletin*, 123:162 – 185, 1998.