Informatics in Education, 2012, Vol. 11, No. 2, 169–190

Detection and Evaluation of Cheating on College Exams using Supervised Classification

Elmano Ramalho CAVALCANTI[1], Carlos Eduardo PIRES[1],
Elmano Pontes CAVALCANTI[2], Vládia Freire PIRES[3]

[1]*Federal University of Campina Grande, Computing and Systems Department*
Campina Grande, PB, Brazil
[2]*Federal University of Campina Grande, Business Management Department*
Campina Grande, PB, Brazil
[3]*Motiva School*
Campina Grande, PB, Brazil
e-mail: elmano@copin.ufcg.edu.br, cesp@dsc.ufcg.edu.br, elmanopc@gmail.com,
vladiafreire@hotmail.com

Received: December 2011

Abstract. Text mining has been used for various purposes, such as document classification and extraction of domain-specific information from text. In this paper we present a study in which text mining methodology and algorithms were properly employed for academic dishonesty (cheating) detection and evaluation on open-ended college exams, based on document classification techniques. Firstly, we propose two classification models for cheating detection by using a decision tree supervised algorithm. Then, both classifiers are compared against the result produced by a domain expert. The results point out that one of the classifiers achieved an excellent quality in detecting and evaluating cheating in exams, making possible its use in real school and college environments.

Keywords: architectures for educational technology system, evaluation methodologies, improving classroom teaching, pedagogical issues

1. Introduction

In a world where most of the corporate data is available in textual format, text mining has emerged as a powerful tool to support knowledge management. Considered as a branch of data mining, the purpose of text mining is to find patterns, tendencies and regularities in documents written in natural language (Feldman and Sanger, 2007). Examples of text mining applications include: extraction of domain-specific information from text, email filtering, search engines, and document categorization (Berry, 2004).

Although data and text mining applications are commonly employed for industrial and commercial purposes, they can also be used for educational aims. Most related work are focused on e-learning environments (Romero *et al.*, 2008; Delavari *et al.*, 2008; Lin *et al.*, 2009) and/or plagiarism detection (Adeva *et al.*, 2006; Sorokina *et al.*, 2006; Butakov and Scherbinin, 2008). However, this work addresses another practical application of

text mining in the education domain: detection and evaluation of academic dishonesty (cheating) on written scholar exams.

According to researches from Brazilian's public universities and schools, in an academic environment the occurrence of cheating is extremely common (da Silva *et al.*, 2006; Silva *et al.*, 2009). This practice represents an old problem without a concrete solution (Rangel, 2001). There is not a precise definition of cheating, but it is supposed that the practice occurs every time two or more exams have a certain degree of similarity with respect to their answers. The cheating dimension is variable. It can be part of a question, the whole question, some questions, or the whole exam. In addition, cheating can be harsh (i.e., copy-paste), or subtle (i.e., a partial copy-paste).

The practice of cheating is present all over the world, in all segments of education, from grade school to graduate school (Davis *et al.*, 2009; Guthrie, 2009). Efforts have been done to find ways to prevent students from cheating (Guthrie, 2009; Broeckelman-Post, 2008) or even to predict when a student will probably cheat (Passow *et al.*, 2006; Kremmer *et al.*, 2007).

Besides prevention and prediction techniques, it is also possible to use computer programs to detect cheating on exams. In this sense, most of the papers propose statistical techniques to detect cheating on multiple choice tests or exams (McManus *et al.*, 2005; Sotaridona *et al.*, 2006; van der Ark *et al.*, 2008; DiSario *et al.*, 2009). On the other side, in this paper we show how text mining algorithms can be used together as a promising technique not only to detect but also to evaluate cheating on open-ended exams. To the best of our knowledge this is the first work that shows how to use the text mining technology in order to develop a solution that detects and evaluates cheating on scholar exams.

The rest of this paper is organized as follows. The related work concerning plagiarism and cheating detection is discussed in Section 2. A background of text mining concepts is presented in Section 3. In Section 4, we describe a case study performed at a Federal University in Brazil, where a supervised classification algorithm was employed to create inference models capable to detect the presence and level of cheating in a real set of scholar exams. Section 5 presents the evaluation of the models, comparing them against a model produced by a human specialist. Section 6 offers an analysis of the results. Finally, our conclusions and suggestions for further work are presented in Section 7.

2. Related Work

A problem that is pedagogically similar to cheating on scholar exams is plagiarizing academic work. Plagiarism is an act of fraud that involves both stealing someone else's work and lying about it afterward. Plagiarism usually occurs in academia where documents are typically essays or reports. However, plagiarism is also widely present in scientific papers, art designs, and program source code.

The widespread use of computers and the advent of the Internet have made it easier to plagiarize others' work. Students are less likely to commit plagiarism if they know that their work will be checked by a plagiarism detection application. Plagiarism detection is the process of locating instances of plagiarism within a work or document. Our related work emphasizes industry and academic solutions for plagiarism detection.

Table 1
Some commercial and free plagiarism tools

Name	License	Description
Ephorus[1]	Proprietary	Web-based application used to prevent and detect plagiarism in scholar work. The user can upload documents to be checked for similarities against Internet sources and other student papers uploaded by instructors. As a result, the application returns a report containing the similarities between the submitted document and the sources found.
Plagium[2]	Proprietary	Web-based application that checks whether the content of a website or research paper has been copied and used elsewhere. It works similar to a search engine. However, differently from Google or Yahoo that often imposes a limit of 10–12 keywords per search, the application accepts much larger blocks of text for searching online. Plagium breaks up the input text into smaller "snippets". These snippets are matched against Web content, with the matches scored to determine what documents match the input text.
Sherlock[3]	Free	It uses digital signatures to find similar pieces of text. Sherlock works on text files such as essays, computer source code files, and other assignments in digital form. The program output offers the percentage of similarity between each pair of documents in the set of documents provided as input.
Urkund[4]	Free	It checks a document against three central sources: the Internet, published materials and materials previously submitted by students, e.g., memos, case studies, and degree work (theses/dissertations). The system highlights the parts of a document that disclose similarities with the three sources. A percentage indication for each hit in the document is offered as output. It is then up to a tutor to decide whether this should be regarded as a piece of plagiarism.

[1]`http://www.ephorus.pt`, [2]`http://www.plagium.com`,
[3]White and Joy (2004), [4]`http://www.urkund.com`.

2.1. *Tools*

There are several commercial and free online applications for text-plagiarism detection. A short description of some of them is presented in Table 1. Most of them use a web-based architecture, checking if a certain document is similar to others available online. However, this reality diverges from the task of detecting cheat on scholar exams since that plagiarism in this case occurs locally, i.e., at a physical location.

2.2. *Academic Papers*

In the literature, many researches deal with the plagiarism problem. (Lukashenko *et al.*, 2007) present a survey of methods and applications to detect plagiarism. More recent articles (Barron-Cedeno and Rosso, 2009; Butakov and Scherbinin, 2008) propose new techniques to deal with the plagiarism problem.

Concerning the practice of cheating, studies point out that this is a habit present all over the world, in all segments of education, from elementary school to graduation (Davis

et al., 2009; Guthrie, 2009). Many efforts have been done to find ways of avoiding students from cheating (Guthrie, 2009; Broeckelman-Post, 2008) or even of preventing a student from cheating (Passow *et al.*, 2006; Kremmer *et al.*, 2007).

Besides the techniques applied to prevent cheating, it is also possible to use computer programs to detect cheating on scholar works and exams. In this sense, most of the articles propose statistical techniques to detect cheating on multiple choice scholar exams (McManus *et al.*, 2005; Sotaridona *et al.*, 2006; van der Ark *et al.*, 2008; DiSario *et al.*, 2009). Instead, in this paper we show how text mining algorithms can be employed to detect and evaluate cheating in exams based on open-ended questions.

3. Background

Data mining usually deals with structured data, i.e., data stored in a well-defined format such as worksheets and databases (Tan *et al.*, 2005). Text mining is considered a type of data mining that deals with non-structured data (Feldman and Sanger, 2007). Information Retrieval as well as supervised and non-supervised classification of documents are some of the research areas in which text mining is applied.

Classification techniques can be defined as the task of assigning objects to one of several predefined categories (also known as class labels; Tan *et al.*, 2005). The classification is said to be supervised when we already have the information of the classes. On the other hand, the non-supervised classification is used when this information is missing.

A representation of the classification task is shown in Fig. 1, where the input x is the set of attributes of an object and the output y is the class label that informs the class of that object. A classification model has an hybrid usage, either as a descriptive model or a predictive model. The former can serve as an explanatory tool to distinguish between objects of different classes. The latter can be used to predict the class label of unknown data. Examples of classification models include: decision tree classifiers, k-nearest neighbors, neural networks, support vector machines, rule-based classifiers, and naive Bayes classifiers (Witten and Frank, 2005).

3.1. *Document Representation*

Due to the non-structured aspect of text documents, an essential task executed at the preprocessing step of the text mining process is to assign some structure to the content stored in the documents (Feldman and Sanger, 2007). This task ensures that documents can be

Fig. 1. Classification as the task of mapping an input x into its class label y (Tan *et al.*, 2005).

better handled by knowledge extraction algorithms. Although some of these algorithms require sophisticated information, such as the ones based on linguistic knowledge, most of the pattern extraction algorithms only require the documents to be represented in a spreadsheet format. In such format, denoted as bag of words, lines correspond to documents and columns represent the terms contained in the document collection. Terms are independent and form an unordered set in which the order of occurrence is not taken into consideration. One possibility to represent a bag of words is using attribute-value tables (Berry, 2004).

An example of such representation is illustrated in Table 2, where d_i corresponds to the ith document, t_j represents the jth attribute (term), a_{ij} is the measure that relates d_i and t_j. y_i represents the class (or label) in which the document is classified.

According to Table 2, each document can be represented as a vector $d_i = (a_i, y_i)$, where $a_i = (a_{i1}, a_{i2}, \ldots, a_{iM})$ and y_i represents the class of the document. Several measures have been proposed to compute the values of a_{ij}. These measures are classified into two types: binary and frequency-based. Binary measures indicate the occurrence (or not) of a term in a certain document. They can be used to extract information about the similarity of documents considering the number of terms in common.

Frequency-based measures aim at counting the occurrences of a certain term in a given document. They can be used for instance to extract statistical measures in the extraction of patterns. Among the frequency-based measures, it is possible to distinguish two other groups: supervised measures, which depend on the availability of data with a well-known class value (last column of Table 2), measuring the importance of a certain attribute to determine the class value; and non-supervised measures which are applicable to non-labelled data.

ConfWeight (Soucy and Mineau, 2005) and Mutual Information (Berry, 2004) are examples of supervised measures. As examples of non-supervised measures we have TF (term frequency), which considers the absolute frequency of terms in documents (Rijsbergen, 1979), IDF (inverse document frequency) (Salton *et al.*, 1975), which computes the inverse frequency of a term, favoring those terms that appear in few documents of the collection; and TF-IDF (Salton and Buckley, 1988), consisting in a combination of the two previous measures (TF and IDF).

Table 2

Documents as a vector representation

	t_1	t_2	$\cdots$	t_j	$\cdots$	t_M	class
d_1	a_{11}	a_{12}	$\cdots$	a_{1j}	$\cdots$	a_{1M}	y_1
d_2	a_{21}	a_{22}	$\cdots$	a_{2j}	$\cdots$	a_{2M}	y_1
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	y_3
d_i	a_{i1}	a_{i2}	$\cdots$	a_{ij}	$\cdots$	a_{iM}	y_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	y_2
d_N	a_{N1}	a_{N2}	$\cdots$	a_{Nj}	$\cdots$	a_{NM}	y_3

3.2. *Similarity Between Documents*

A common way to check whether two documents are similar is to verify the terms (words) contained in both documents. Additionally, it is necessary to verify the frequency of each term in the document. Such method is called term frequency (TF). However, due to the high occurrence of some kinds of terms (e.g., articles or prepositions), the inverse frequency factor of a document (IDF) is used to ponder the frequency of terms. As a result, frequent terms have a lower weight than unusual terms. This method is denoted as TF-IDF (term frequency – inverse document frequency). It was proposed by (Salton and Buckley, 1988) and is commonly used in Information Retrieval (Soucy and Mineau, 2005).

Formally, the frequency of a term i that appear in a document d_j is:

$$\mathrm{TF}_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}, \tag{1}$$

where $n_{i,j}$ is the occurrence of the term i in document d_j and the denominator is the sum of occurrences of all terms in d_j. Given that N is the total of documents, the formula that computes the inverse frequency of a document (IDF) is:

$$\mathrm{IDF}_i = \log \frac{N}{|d\colon t_i \in d|}, \tag{2}$$

where $|d\colon t_i \in d|$ represents the number of documents in which the term t_i appears. In this sense, the value of TF-IDF for a term i in a document j is:

$$\mathrm{TFIDF}_{i,j} = \mathrm{TF}_{i,j} \times \mathrm{IDF}_i. \tag{3}$$

The computational cost of the method TF-IDF is O(NM), where N is the number of documents and M is the number of terms (see Table 2).

As discussed in Section 3.1, a document is represented as a vector $d_i = (a_{i1}, a_{i2}, \ldots, a_{iM})$, where each term a_{ij} is calculated according to the TF-IDF method. The similarity between two documents D_1 and D_2 is determined by the cosine between the two vectors (4).

$$\mathrm{Cosine}(\vec{D}_1, \vec{D}_2) = \frac{\vec{D}_1 \bullet \vec{D}_2}{|\vec{D}_1||\vec{D}_2|}, \tag{4}$$

where $\vec{D}_1 \bullet \vec{D}_2$ represents the scalar product of the vectors whilst $|\vec{D}_1|$ and $|\vec{D}_2|$ represent the module of the vectors.

The cosine similarity value is a positive number which varies between 0 (minimum) and 1 (maximum). The first value implies that the two documents are totally different, and the second that they are completely similar. The cosine similarity method is considered a standard measure in text mining researches (Berry, 2004; Weiss *et al.*, 2005).

Another text similarity metric is the overlap coefficient, derived from the Jaccard coefficient (Berry, 2004). To compute this metric, instead of using the document as a vector

we make use of the document itself, which can be viewed as a set of words. The overlap between two documents/sets D_1 and D_2 is equal to the intersection between the two sets divided by the size of the smaller one (5). As the cosine similarity, the value ranges from 0 (minimum) to 1 (maximum). Similarly, 0 indicates no document similarity, and 1 maximum similarity. Examples of open source libraries containing text similarity metrics include SimMetrics (Chapman, 2004) and SecondString (Cohen *et al.*, 2003).

$$\text{Overlap}(D_1, D_2) = \frac{|D_1 \cap D_2|}{\min(|D_1|, |D_2|)}. \tag{5}$$

3.3. *Quality Metrics*

In the text mining literature, there are several metrics that quantify and qualify the predictive models (e.g., supervised classification and regression). Table 3 presents the number of correct classifications in contrast with predicted classifications for the classes '+' e '−' of a binary model. This table, denoted confusion matrix, enables the computation of the following metrics: accuracy, precision, and recall.

The recall of a class is defined as the ratio between the number of correctly classified documents and all documents belonging to the class. Precision is the ratio between the number of correctly classified documents and all documents considered by the model as belonging to the class (Feldman and Sanger, 2007).

While the previous metrics are calculated for each class of the model, Accuracy is a global metric. It reflects the hit ratio, i.e., the proportion between the correctly inferred classifications and the total of inferred classifications. Considering the example of Table 3, we have that:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}}. \tag{6}$$

Besides the previous metrics, there is a statistical coefficient denoted Kappa index or K Statistic (Cohen, 1960), which is a measure of agreement in nominal scales, largely used in Medicine, although there is also an occurrence of using this metric on detecting answer copying in exams (Sotaridona *et al.*, 2006). When applied to the context of the

Table 3

An example of a confusion matrix for a problem involving two classes

Prediction	True		Precision
	+	−	
+	TP[a]	FP[c]	$\text{TP}/(\text{TP}+\text{FP})$
−	FN[b]	TN[d]	$\text{TN}/(\text{TN}+\text{FN})$
Recall	$\text{TP}/(\text{TP}+\text{FN})$	$\text{FP}/(\text{FP}+\text{TN})$	

[a] True positive. [b] False negative. [c] False positive. [d] True negative.

text mining task classification, the Kappa index indicates the level of agreement between the model classification and a reference classification. In other words, it determines how much the two models agree with respect to the classification.

Considering the confusion matrix described in Table 3, the Kappa index is calculated according to (refeq:k). As the cosine similarity metric, $\hat{k}$ also varies between 0 (minimum agreement) and 1 (maximum agreement).

$$\hat{k} = \frac{n^2 \cdot \text{accuracy} - [X + Y]}{n^2 - [X + Y]}, \tag{7}$$

where $X = (\text{TP} + \text{FN}) \cdot (\text{TP} + \text{FP})$, $Y = (\text{FN} + \text{TN}) \cdot (\text{FN} + \text{FP})$ and accuracy is given by (6). The use of the previous quality metrics enables the adequate evaluation of the cheating classification models presented in the following section.

4. Methodology

To check how text mining and supervised classification techniques can be applied together in the detection of cheating on scholar exams, we developed a case study. It was performed at the Federal University of Campina Grande – Brazil, in a project involving the Business Management and Computer Science departments. Considering that text mining is a sub-area of data mining, the steps followed in the case study are based on the data mining methodology proposed by Tan *et al.* (2005). The steps include: data selection, preprocessing, data transformation, data mining, and analysis.

4.1. *Data Selection and Preprocessing*

A set of thirty scholar exams written in the Brazilian Portuguese language were selected to compose the case study. Each exam contained four open-ended questions in the area of administration and sub-area of marketing. The exams were answered by the students and stored in electronic format as plain text (e.g., text files). There was no need for sampling operation, since all the exams were used in the data mining process.

In a real life situation, a teacher detects cheating when comparing the answer of some question answered by a student A against the answer of the same question provided by a student B. In this sense, we divided each exam into four distinct parts. Each part corresponds to a different question. The answer (text) of each question was considered as the target for the text mining process. For each question, we defined a controlled dictionary containing a set of words that could be used by students to answer the question. In this light, when two answers of the same question contained a high number of identical words, it was considered a strong evidence of cheating.

To enable the correct application of the data mining algorithms, punctuation and accentuation were removed from each document. In general, this task is needed to minimize the size of document vectors (Table 2) as well as to avoid the need to distinguish words that in fact are lexically the same (e.g., 'elétrico' vs. 'eletrico'[1]). Although such opera-

[1] In English, electric.

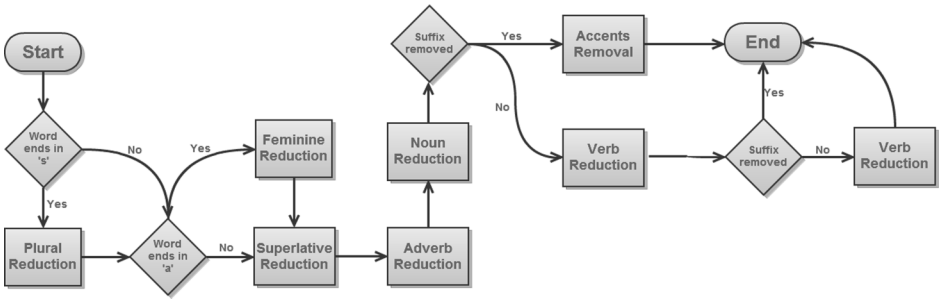


Fig. 2. Portuguese stemming. Source: adapted from Morais (2007).

tion can bring benefits in most of the cases, we are aware that it can treat two lexically different words as the same. For instance, words that are different due to the use of accentuation (e.g., pelo/pélo/pêlo[2])[3]. However, since these cases are unusual, we believe that the operation can bring more gains than losses. This task was implemented using the Java API 1.5 and the Eclipse IDE tool (Eclipse, 2010).

4.2. *Data Transformation*

After removing punctuation and accentuation, we started a tokenization process to transform each document into a set of words or tokens. Tokens with less than three characters were not considered. This enabled the removal of common grammatical elements, e.g., prepositions, articles, and conjunctions. It also helped to minimize the size of document vectors and optimize the data mining algorithm.

The next step consisted in removing irrelevant words (denoted as stopwords). To this end, we used an adaptation of the stopword dictionary from the Snowball project (Porter and Boulton, 2002), which is written in the Brazilian Portuguese language.

Afterwards, a morphological normalization process (denoted as stemming) was performed. Such process consists in transforming words into primitive terms (see Fig. 2). For instance, consider that in the answer of the question X a student A uses the phrase '... processes the product to ...', whilst student B takes a look at the exam of student A and writes '... the product is processed to ...'. We can notice that the words 'processes' and 'processed' have the same radical 'process'. The morphological normalization enables the removal of characters referring to plural, feminine gender, augmentative, diminutive, etc., keeping only the radical of the words. This step was implemented using the stemming algorithm of the Snowball project (Porter and Boulton, 2002).

Documents also need to be semantically normalized. This task consists in mapping all the synonyms of a word into a single base term. To this end, a lexical base written in the same language of the documents can be used. Examples of lexical bases for the English

[2] In English, for/to strip/pelage.

[3] With the new Portuguese language spelling international agreement, some words that were differentiated by accentuation are now written in the same way. As an example, the words pelo/pélo/pêlo will be written without accentuation (Cunha, 2009).

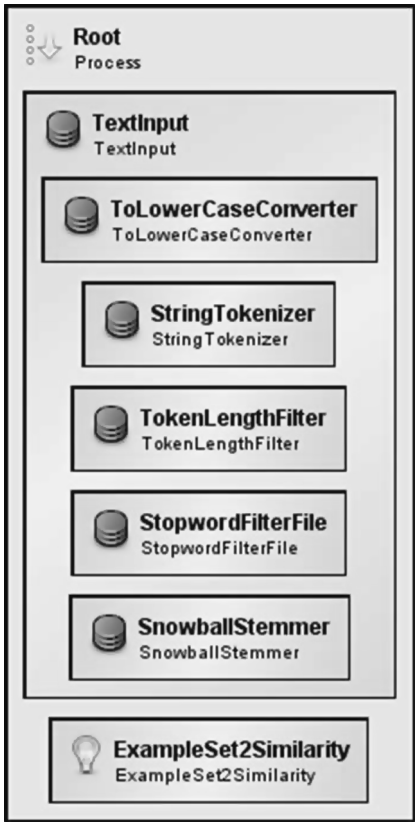


Fig. 3. Text mining process flow in the RapidMiner tool.

and Portuguese languages are WordNet (Miller, 1995) and WordNet.PT (Palmira *et al.*, 2010), respectively. After all the previous tasks, each question of each student (document) was transformed into a vector of words (as detailed in Section 3.1), according to the TF-IDF method (equation (3) in Section 3.2).

Since the size of the answers was small (i.e., one or two paragraphs), the use of vector compressing techniques was not required. The average size of the vectors was nearly 450 columns. In addition, pruning techniques were not considered, since its use led to worst results during the similarity computation between documents.

All tasks related to data transformation were performed using the RapidMiner tool (Mierswa *et al.*, 2006; Rapid-I, 2010), an open source software for knowledge discovery, machine learning, and data mining. Fig. 3 illustrates the tasks of the data transformation step, detached as a rectangle.

4.3. *Text Mining*

The data mining process involved two tasks. First, we computed the cosine similarity and the overlap coefficient for each pair of documents (operator *ExampleSet2Similarity*

Table 4
Portion of the data obtained after text mining and used for the supervised classification models

ID	Cosine Similarity	Overlap Coefficient	Cheating Level[a]
$\langle 1, 1, 25\rangle$	0.817540505	0.803810564	High
$\langle 1, 1, 30\rangle$	0.771951927	0.780808721	High
$\langle 1, 2, 29\rangle$	0.71391944	0.68778898	High
$\langle 1, 25, 30\rangle$	0.69305863	0.690211565	High
$\langle 1, 13, 24\rangle$	0.501569582	0.457819896	High
$\langle 1, 13, 23\rangle$	0.475967316	0.395380058	High
$\langle 1, 23, 24\rangle$	0.384379516	0.384832118	High
$\langle 1, 12, 22\rangle$	0.348145528	0.358832118	Intermediary
$\langle 1, 1, 27\rangle$	0.262484497	0.244609219	Low
$\langle 1, 25, 27\rangle$	0.25834858	0.244129913	Low
$\langle 1, 13, 8\rangle$	0.240827327	0.151298724	None
$\langle 1, 27, 30\rangle$	0.239963138	0.245380561	Low
$\langle 1, 1, 26\rangle$	0.21861438	0.301880606	Intermediary
$\langle 1, 26, 30\rangle$	0.19805587	0.29802218	Low
$\langle 1, 11, 17\rangle$	0.187388827	0.198169741	None
$\langle 1, 11, 21\rangle$	0.181043601	0.231319903	Low
$\langle 1, 24, 8\rangle$	0.166791662	0.118873948	None
$\langle 1, 20, 24\rangle$	0.160367724	0.182427147	None

[a] According to the human specialist.

of Fig. 3). Then, we created a new data sheet containing the values of these two metrics. Table 4 shows an excerpt from this data sheet. The first column, ID, specifies the question identifier Q and the students code, X and Y. The second and third columns contain respectively the values of the cosine similarity (4) and the overlap coefficient (5) between the answers provided by students X and Y for question Q. The last column was filled with the cheating level identified after a traditional exam evaluation done by the course lecturer, denoted here as the *specialist*. Each pair of exams contains different levels of cheating: none, low, intermediary, and high. The cheating mapping between all students is detailed in Table 5. Since our sample has thirty exams, and each one contains four questions, both the cosine similarity and the overlap coefficient were executed $4 \cdot \binom{30}{2} = 4 \cdot 435 = 1740$ times.

5. Results

To simplify the visualization of cheating on exams, consider a graph $G = \langle V, A\rangle$, where V is the set of exams (identified by the student code) and A is the set of edges that link questions whose similarity is higher than a threshold γ. For each value of γ there is a unique similarity graph. With a correct adjust of γ, it is possible to obtain similarity graphs for each level of cheating.

Table 5
Cheating on exams done by students according to the human specialist

	Question 1			Question 2			Question 3			Question 4		
ID	L[a]	I[b]	H[c]	L[a]	I[b]	H[c]	L[a]	I[b]	H[c]	L[a]	I[b]	H[c]
1	27	26	25, 30			30			30			30
2			29		25	29			29		27	29
3						27						
5						26	27			28		
7									25, 26			26
11	21				21		21			21		
12		22				22, 24	22				22	
13			23, 24			23		23				23, 24
21	11				11		11			11		
22		12				12, 24	12				12	
23			13, 24			13		13			24	13, 24
24			13, 23			12, 22					23	13, 23
25	27		1, 30		29	2	5		7, 26			
26		1				5			7, 25			7
27	1, 25, 30					3					2, 29	
28										5		
29			2		25	2			2		27	2
30	27		1, 25			1			1			1

[a] Low, [b] Intermediary, and [c] High cheating.

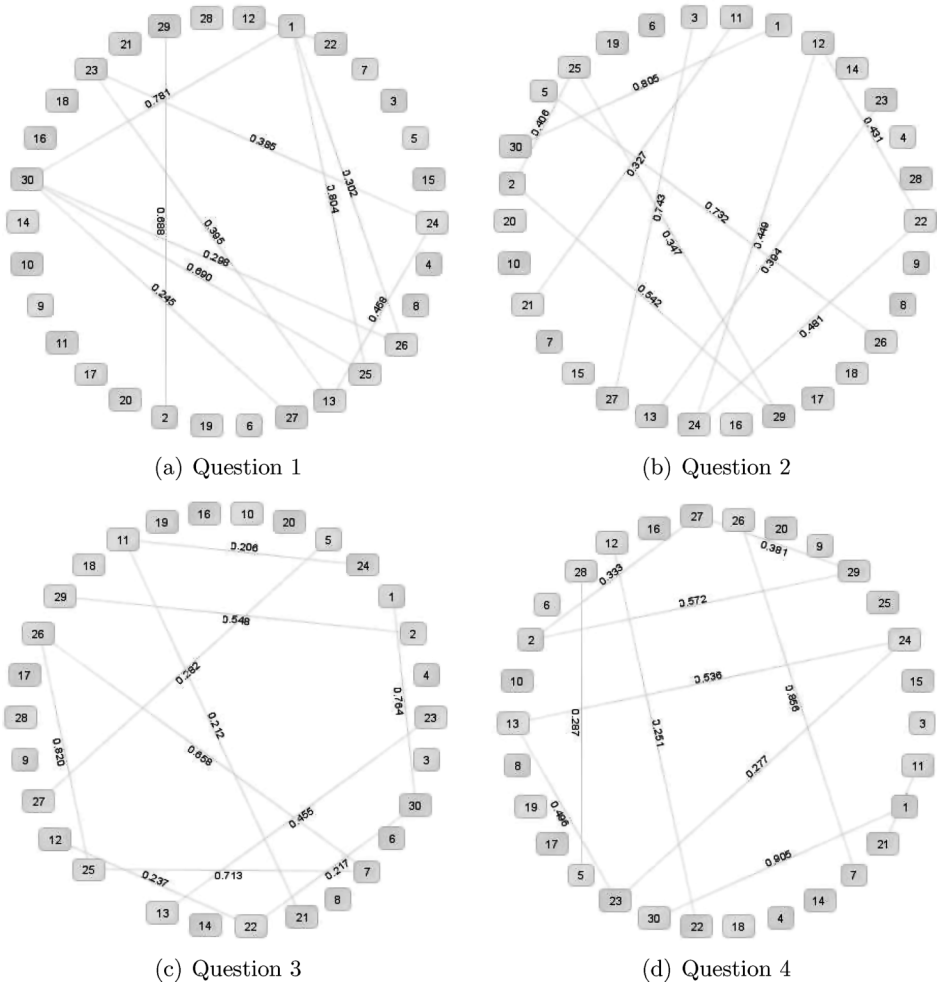
Figure 4(a) is a circular graph illustrating the pairs for the first question that had higher values regarding the overlap coefficient, meaning the students that probably cheat on this question. The circular graphs for the remaining questions are presented in Figs. 4(b), 4(c) and 4(d). Another form of visualization is shown in Fig. 5 where the most similar questions are placed near each other[4]. This form helps a teacher to quickly discover the students that answered the question in a similar manner.

5.1. *Supervised Classification Models*

A Decision Tree (DT) algorithm was employed in order to build models able to detect and evaluate cheating on scholar exams. DT is considered one of the most widespread and consolidated supervised classification algorithms (Larose, 2004).

We use a DT algorithm similar to the C4.5 algorithm (Quinlan, 1993). The maximum tree depth was set to 4, which corresponds to the number of classes (high, intermediary, low and none), and the confidence level for pessimistic pruning was set to 0.25. Both the cosine similarity and the overlap coefficient between all pairs of questions were used as input data (i.e., attribute), resulting in two classification models of cheating.

[4]The graph was drawn according to Peter Eades' method for drawing undirected graph (Eades *et al.*, 2010).



(a) Question 1 (b) Question 2

(c) Question 3 (d) Question 4

Fig. 4. Similarity graphs for all exam questions.

The validation of the DT models was done through the stratified ten-fold cross-validation approach, which is the standard statistical technique for validating a learning algorithm (Larose, 2004). In this technique, the data is divided randomly and uniformly into 10 parts (stratified sampling). Each part is used as a holdout set and the other nine parts are used to train the model, totalizing ten combinations for testing. For each one, the error rate is calculated on the holdout set, and thus the learning procedure is executed 10 times using different training sets. Finally, the 10 error estimations are averaged to yield an overall error estimate.

The cheating percentages defined by the specialist and the two DT models are presented in Fig. 6. The DT cosine based-similarity hit the real intermediary cheating percentage (i.e., 19%), but the low and high cheating percentages were far from the specialist's model (17%/64% and 26%/55%). On the other hand, the DT overlap-based cheating

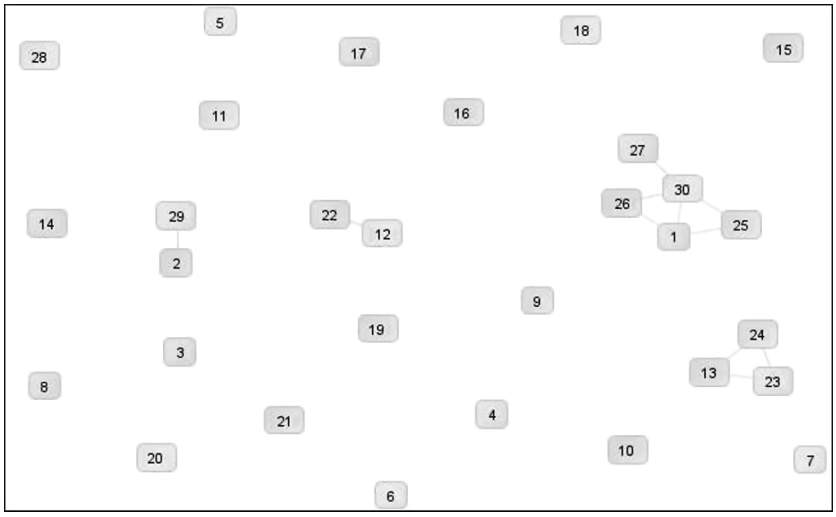


Fig. 5. Groups of similar answers for the first question of the exam.

percentages were closer to the specialist's model. Thus, for this first comparison, the DT overlap-based showed results closer to the specialist's conclusion.

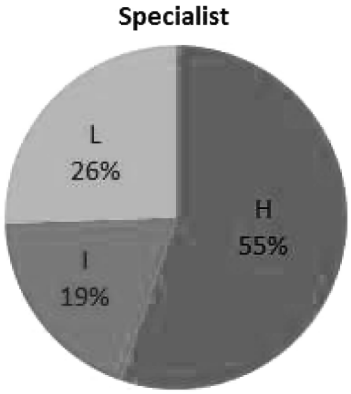
The decision tree model based on the cosine similarity is shown in Fig. 7. Let $\mathrm{Cosine}(Q, X, Y)$ mean the cosine similarity between the answers to the question Q provided by students X and Y. According to this decision tree, if $\mathrm{Cosine}(Q, X, Y) > 0.358$, then the model classifies the cheating as *high*. If $0.288 < \mathrm{Cosine}(Q, X, Y) \leqslant 0.358$ than it is an *intermediary* cheating. Obviously, this model can produce wrong levels of cheating. These errors are reported in the confusion matrix (Table 6). The precision for detecting *high* cheating was 92.59% but only 37.50% for detecting *intermediary* cheating. In addition, the model presented low recall values for *low* and *intermediary* cheating. In short, this model was good on detecting cheating, but reasonable for evaluating cheating dimension.

The other DT model (Fig. 8), based on the overlap coefficient, had better results for all quality metrics (Table 7). There was only one occurrence of false positive, when the model detected a false *low* cheating. The major improvements against the cosine model occurred in the prediction of intermediary and low cheating (66.67% and 69.23% versus 37.50% and 42.86%), and the recall of low cheating (75.00% versus 25.00%).

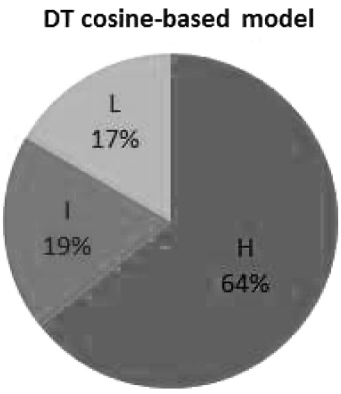
A comparison between the two decision tree models is given in Table 8. The DT overlap model achieved better results for both Accuracy and Kappa index[5] as well as a lower standard deviation for these quality metrics. The table also shows a 99% confidence interval for the accuracy and 95% for the Kappa index.

We defined a hypothesis test in order to check if the classifier models based on the overlap metric had a high agreement with the reference model (i.e., specialist). To this end, we consider (9) as the null hypothesis, and (9) as our research hypothesis. We con-

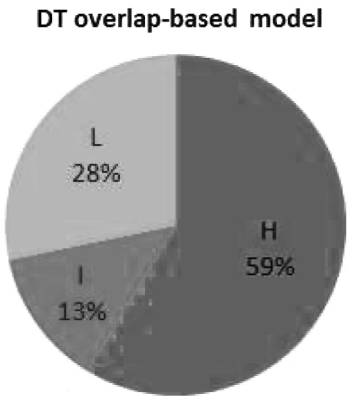
[5]The Kappa index was calculated according to Fleiss *et al.* (1969, 2003).



(a) Specialist



(b) Decision tree based on the cosine similarity



(c) Decision tree based on the overlap coefficient

Fig. 6. Percentages of cheating related to the classification models and the specialist.

Table 6

Decision tree confusion matrix when using cosine similarity as unique attribute

Prediction	True				
	High	Intermediary	Low	None	Precision
High	25	2	0	0	92.59%
Intermediary	0	3	3	2	37.50%
Low	1	2	3	1	42.86%
None	0	2	6	1690	99.53%
Recall	96.15%	33.33%	25.00%	99.82%	

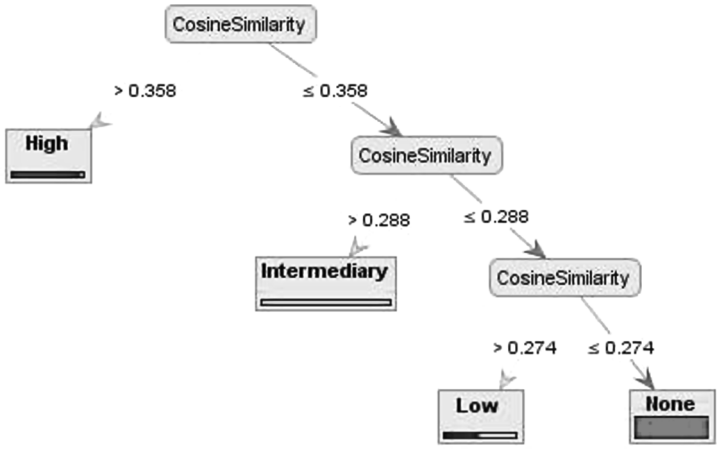


Fig. 7. Decision tree classification model based on the cosine similarity value between two exam answers.

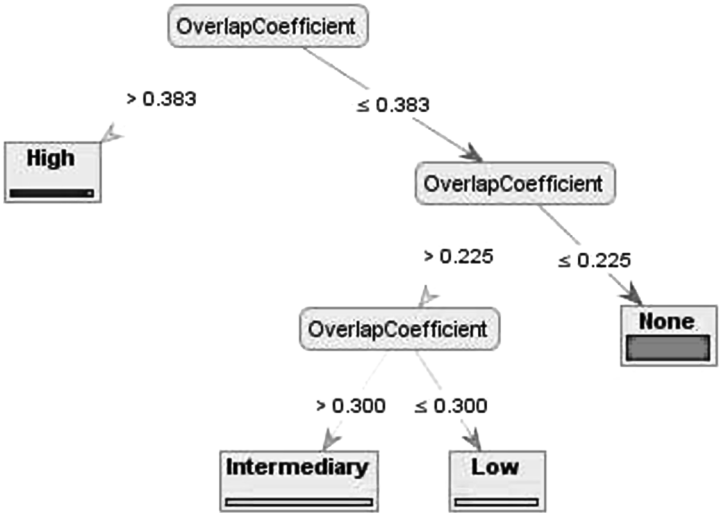


Fig. 8. Decision tree classification model based on the overlap coefficient value between two exam answers.

Table 7

Decision tree confusion matrix when using overlap coefficient as unique attribute

Prediction	True				
	High	Intermediary	Low	None	Precision
High	25	2	0	0	92.59%
Intermediary	1	4	1	0	66.67%
Low	0	3	9	1	69.23%
None	0	0	2	1692	99.88%
Recall	96.15%	44.44%	75.00%	99.94%	

Table 8
Comparison between the decision tree classification models

DT model	Accuracy				Kappa			
	mean	std.	.99 Conf.	Int.	mean	std. error	.95 Conf.	Int.
Cosine	98.91%	0.60%	98.55%	98.90%	0.785	0.0443	0.6957	0.8693
Overlap	99.43 %	0.36 %	99.05 %	99.43 %	0.8904	0.0318	0.828	0.9528

sidered $\hat{k}_1$ as the Kappa index for the model based on the cosine similarity and $\hat{k}_2$ for the model using the overlap coefficient. The hypothesis test is stated at the 95% confidence level.

$$H_0\colon \hat{k}_1 - \hat{k}_2 = 0, \tag{8}$$

$$H_1\colon \hat{k}_1 - \hat{k}_2 < 0. \tag{9}$$

Therefore, we solve the (10) to find the p-value associated to the hypothesis tests:

$$z = \frac{\hat{k}_1 - \hat{k}_2}{\sqrt{\mathrm{Var}(\hat{k}_1) - \mathrm{Var}(\hat{k}_2)}} = \frac{0.785 - 0.8904}{\sqrt{0.00196 - 0.00101}} = -1.9328$$
$$(p = 0.027). \tag{10}$$

We rejected the null hypothesis with 5% of significance level, meaning that the agreement level between the DT overlap-based and the reference models is higher than the DT cosine-based and the reference models.

6. Discussion

Besides the aforementioned points, we proposed and compared two possible classification models for cheating detection using the decision tree supervised algorithm: one based on the cosine similarity, and the other based on the overlap coefficient. The latter presented better results, achieving an accuracy of 99.43% $+-$ 0.36%, and an agreement level (Kappa index) of 0.89 $+-$ 0.032 in comparison with the specialist's result. This suggests an excellent inference quality in the detection and evaluation of cheating (Landis and Koch, 1977).

The decision tree depicted in Fig. 8 can be used as a kind of oracle for cheating detection without the need for the teacher to manually detect the cheating. After the preprocessing and transformations steps (Sections 4.1 and 4.2), the only necessary task is to compute the overlap coefficient for all pairs of exams' answers. All these steps can be done automatically using, for example, the RapidMiner tool. After that, one can directly

use the decision rules provided by the decision tree (Fig. 8). Considering that A and B are the answers for the same question provided by two different students, then we have that:[6]

(i.) $\text{overlap}(A, B) \leqslant 0.22$: no cheating,
(ii.) $0.22 < \text{overlap}(A, B) \leqslant 0.30$: low cheating,
(iii.) $0.30 < \text{overlap}(A, B) \leqslant 0.38$: intermediary cheating,
(iv.) $\text{overlap}(A, B) > 0.38$: high cheating.

However, it also important to mention that we cannot affirm that the cheating detection model can be used for any kind of exam (e.g., a mix between close-ended and open-ended questions), as well as for any kind of course (e.g., mathematics or physics). The results presented in this paper are valid and indicated to be used only in similar exam's conditions (i.e., only open-ended questions).

7. Conclusions

The first point to mention is that a successful case study was employed on the utilization of data mining's methodology and algorithms for helping teachers to deal with an old educational problem: academic dishonesty (cheating) on exams. Besides that, it is noteworthy that only open source (i.e., free) programs were used for all data mining tasks. Thus, any person can take advantage of this work in order to repeat the methodology for his/her own purpose and without any additional financial charge.

In this paper we have detailed a potential application that employs text mining in education domain. Precisely, it is shown that text mining can be satisfactorily used to develop a mechanism for detection and evaluation of cheating on exams based on open-ended questions. The solution presented in this paper also fits the need for cheating detection on other written-based methods for student assessment (e.g., homeworks).

The solution presented in this paper can assist a teacher in the difficult and labor-intensive task of detecting and evaluating cheating on exams. As further work we intend to execute more experiments with scholar exams on other research areas. In addition, we intend to consider the physical distribution of students in the classroom as an input to the cheating classification model.

References

Adeva, J., Carroll, N., Calvo, R. (2006). Applying plagiarism detection to engineering education. In: *Proceedings of the 7th International Conference on Information Technology Based Higher Education and Training*, IEEE, 722–731.

Barron-Cedeno, A., Rosso, P. (2009). On automatic plagiarism detection based on n-grams comparison. In: *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval* (ECIR'09), Springer-Verlag, Berlin, Heidelberg, 696–700.

Berry, M. (2004). *Survey of Text Mining I. Clustering, Classification, and Retrieval*. Springer.

[6]For the sake of simplicity, one could consider the values 0.2, 0.3 and 0.4 as the cutoff values for the cheating size.

Broeckelman-Post, M.A. (2008). Faculty and student classroom influences on academic dishonesty. *Informatics in Education*, 51(2), 206–211.

Butakov, S., Scherbinin, V. (2008). The toolbox for local and global plagiarism detection. *Computers & Education*, 52(4), 781–788.

Chapman, S. (2004). Simmetrics: a java & c# .net library of similarity metrics. `http://sourceforge.net/projects/simmetrics/`.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1), 37–46.

Broeckelman-Post, M.A. (2008). Facultu and student classroom influences on academic dishonesty. *Informatics in Education*, 51(2), 206-211.

Cohen, W.W., Ravikumar, P., Fienberg, S. (2003). Second string: Open source java-based package of approximate string matching. `http://secondstring.sourceforge.net/`.

Cunha, A.G. (2009). *Vocabulário Ortográfico da Língua Portuguesa*, 2nd Edition. Lexikon, São Paulo.

da Silva, G.A., da Rocha, M.M.O.E., Pereira, Y.L., Bussab, V.S.R. (2006). Um estudo sobre a prática da cola entre universitários. *Psicol. Refl., Crít.*, 19(1), 18–24.

Davis, S.F., Drinan, P.F., Gallant, T.B. (2009). *Cheating in School: What We Know and What We Can Do*. Wiley-Blackwell.

Delavari, N., Phon-Amnuaisk, S., Beikzadeh, M.R. (2008). Data mining application in higher learning institutions. *Informatics in Education*, 7(1), 31–54.

DiSario, R., Olinsky, A., Quinn, J., Schumacher, P. (2009). Applying Monte Carlo simulation to determine the likelihood of cheating on a multiplechoice professional exam. *CS-BIGS*, 3(1), 30–36.

Eades, P., Gutwenger, C., Hong, S.-H., Mutzel, P. (2010). Graph drawing algorithms. In: *Algorithms and Theory of Computation Handbook: Special Topics and Techniques*, 2nd edn. CRC Press, 6–6.

Eclipse (2010). Eclipse ide. `http://www.eclipse.org/`.

Feldman, R., Sanger, J. (2007). *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press.

Fleiss, J., Cohen, J., Everitt, B. (1969). Large sample standard errors of kappa and weighted kappa. *Psychological Bulletin*, 72(5), 323–327.

Fleiss, J.L., Levin, B., Paik, M.C., Fleiss, J. (2003). *Statistical Methods for Rates and Proportions*, 3rd edn., Wiley-Interscience, New York.

Guthrie, C. (2009). Plagiarism and cheating: A mixed methods study of student academic dishonesty. PhD thesis, University of Waikato.

Kremmer, M.L., Brimble, M., Stevenson-Clarke, P. (2007). Investigating the probability of student cheating: the relevance of student characteristics, assessment items, perceptions of prevalence and history of engagement. *International Journal for Educational Integrity*, 3(2), 3–17.

Landis, J.R., Koch, G.G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 33(1), 159–174.

Larose, D.T. (2004). *Discovering Knowledge in Data – An Introduction to Data Mining*, Wiley.

Lin, F.-R., Hsieh, L.-S., Chuang, F.-T. (2009). Discovering genres of online discussion threads via text mining. *Computers & Education*, 52(2), 481–495.

Lukashenko, R., Graudina, V., Grundspenkis, J. (2007). Computer-based plagiarism detection methods and tools: an overview. In: *Proceedings of the 2007 International Conference on Computer Systems and Technologies* (CompSysTech'07), ACM, New York, USA, 40, 1–6. `http://doi.acm.org/10.1145/1330598.1330642`.

McManus, I. C., Lissauer, T., Williams, S. E. (2005). Detecting cheating in written medical examinations by statistical analysis of similarity of answers: pilot study. *British Medical Journal*, 330(7499), 1064–1066.

Mierswa, I., Lemmen, F., Wurst, M., Klinkenberg, R., Scholz, M., Euler, T. (2006). Yale: Rapid prototyping for complex data mining tasks. In: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, New York, USA, 935–940.

Miller, G.A. (1995). Wordnet: A lexical database for English. *Communications of the ACM*, 38(11), 39–41.

Morais, E.A.M. (2007). Contextualização de documentos em domínios representados por ontologias utilizando mineração de textos. Master's thesis, Institute of Informatics of the Federal University of Goiais, Brazil.

Palmira, M., Amaro, R., Chaves, R. P., Lourosa, S., Martins, C., Mendes, S. (2010). Rede léxico-conceptual do português. `http://www.clul.ul.pt/clg/wordnetpt`.

Passow, H.J., Mayhew, M.J., Finelli, C.J., Harding, T.S., Carpenter, D.D. (2006). Factors influencing engineering students' decisions to cheat by type of assessment. *Research in Higher Education*, 47(6), 643–684.

Porter, M.F., Boulton, R. (2002). *Snowball: A Language for Stemming Algorithms*. `http://snowball.tartarus.org/`.

Quinlan, J.R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

Rangel, M. (2001). O problema da cola sob a ótica das representações. *Revista Brasileira de Estudos Pedagógicos*, 82, 78–88.

Rapid-I (2010). Rapidminer – open-source data mining with the java software rapidminer. `http://rapid-i.com/`.

Rijsbergen, C.J.V. (1979). *Information Retrieval*. Butterworths.

Romero, C., Ventura, S., García, E. (2008). Data mining in course management systems: Moodle case study and tutorial. *Computers & Education*, 51(1), 368–384.

Salton, G., Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5), 513–523.

Salton, G., Wong, A., Yang, A.C.S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18, 229–237.

Silva, J.C.X., Leal, C.E., Lanes, S.M., Barbosa, L.F., Santos, L.F., Corrêea, M.B., Pessanha, P.R., de Azeredo, S.R., Fejolo, T., Silva, W.J., Alves, A., Jan (2009). O uso da cola como fator que prejudica a relação ensino-aprendizagem. In: *XVIII Simpósio Nacional de Ensino de Física*, Vitória, Espírito Santo.

Sorokina, D., Gehrke, J., Warner, S., Ginsparg, P. (2006). Plagiarism detection in arxiv. In: *Proceedings of the IEEE International Conference on Data Mining*, IEEE Computer Society, Los Alamitos, CA, USA, 1070–1075.

Sotaridona, L.S., van der Linden, W.J., Meijer, R.R. (2006). Detecting answer copying using the kappa statistic. *Applied Psychological Measurement*, 30(5), 412–431.

Soucy, P., Mineau, G. (2005). Beyond TFIDF weighting for text categorization in the vector space model. In: *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, Edinburgh, Scotand, 1130–1135.

Tan, P.-N., Steinbach, M., Kumar, V. (2005). *Introduction to Data Mining*. Addison-Wesley.

van der Ark, L.A., Emons, W.H.M., Sijtsma, K. (2008). Detecting answer copying using alternate test forms and seat locations in small-scale examinations. *Journal of Educational Measurement*, 45(2), 99–117.

Weiss, S.M., Indurkhya, N., Zhang, T., Damerau, F.J. (2005). *Text Mining: Predictive Methods for Analyzing Unstructured Information*. Springer.

White, D.R., Joy, M.S. (2004). Sentence-based natural language plagiarism detection. *J. Educ. Resour. Comput.*, 4. `http://doi.acm.org/10.1145/1086339.1086341`.

Witten, I.H., Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edn., Morgan Kaufmann.

E.R. Cavalcanti is PhD candidate in computer science at Federal University of Campina Grande, Brazil. He holds a MSc in computer science from the same institution. His main research topics and areas of interest include mobile ad hoc network, network simulation, data mining, and informatics in education. He is one of the recipients of the IARIA Ubicomm 2010 Best Paper Award and was indicated for the best paper award in the ACM International Symposium on Mobility Management and Wireless Access (MobiWac 2011). Currently he is a professor of information systems at the Center for Higher Education and Development (CESED) at Campina Grande, Brazil. He's a Brazilian Computer Society and ACM member.

C.E. Pires holds a PhD in computer science from the Universidade Federal de Pernambuco, Brazil. Since November 2009, he is an associate professor at the Computing and Systems Department of the Universidade Federal de Campina Grande, Brazil, where he is currently a member of the Information Systems and Databases Laboratory. He is an Oracle Certified Professional (OCP) and has worked as a database consultant for several years. He has experience in computer science, with emphasis on databases, acting on the following topics: decision support systems, knowledge discovery, database tuning, and information integration.

E.P. Cavalcanti is an associate professor of business administration at Federal University of Campina Grande since the summer of 2007. He began teaching at the Paraba State University in early of 1989. At the end of 1996 he began to teach at the Federal University of Paraba. He worked in a large food processing industry for several years. During this period, he created a software company with three partners. Currently, he teaches and runs a project of the Ministry of Education of the Brazilian government called PET. Professor Cavalcanti's research focuses on competitive intelligence and marketing. Professor Cavalcanti holds a PhD and MSc in business administration.

V.F. Pires graduated from Universidade Federal da Paraiba, Brazil, in 2006, with a degree in pedagogy. Her areas of interest involve early child development and literacy. She obtained her specialist degree in institutional and clinical psychopedagogy from IBPEX/Facinter. Pires worked as a preschool teacher for almost six years. Currently, she works at Colégio Motiva supervising the preschool education program, while continuing to expand her knowledge about many other cultural teaching.

Sukčiavimo per egzaminus nustatymas ir įvertinimas naudojant prižiūrėjimo klasifikaciją

Elmano Ramalho CAVALCANTI, Carlos Eduardo PIRES,
Elmano Pontes CAVALCANTI, Vládia Freire PIRES

Teksto gavyba buvo naudojama įvairiems tikslams, pavyzdžiui, dokumentų klasifikavimui ir specifinės srities informacijos ištraukimui iš teksto. Šiame straipsnyje autoriai pateikia tyrimą, kuriame teksto gavybos metodika ir algoritmai buvo išsamiai naudojami nustatyti ir įvertinti akademinį nesąžiningumą (apgaulę) per neterminuotus kolegijos egzaminus, remiantis dokumentų klasifikavimu. Visų pirma, siūlomi du klasifikavimo modeliai sukčiavimui nustatyti, naudojant sprendimų medžio prižiūrėjimo algoritmą. Abiejų klasifikatorių rezultatai palyginti su išvadomis, pateiktomis tos srities eksperto. Pasirodė, kad vienu iš klasifikatorių puikiai nustatomas ir įvertinamas sukčiavimas per egzaminus, todėl juo galima naudotis realiame mokyklos ir kolegijos darbe.