Expert Systems With Applications 150 (2020) 113265

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Opinion mining and emotion recognition applied to learning environments

María Lucía Barrón Estrada[a], Ramón Zatarain Cabada[a,*], Raúl Oramas Bustillos[b], Mario Graff[c]

[a] Tecnológico Nacional de México, Instituto Tecnológico de Culiacán, Juan de Dios Bátiz 310 Pte. Col. Guadalupe, Culiacán, Sinaloa CP 80220, México
[b] Universidad Autónoma de Occidente, Blvd. Lola Beltrán y Blvd. Rolando Arjona Amabilis s/n, col. 4 de marzo, Culiacán, Sinaloa CP 80020, México
[c] INFOTEC Aguascalientes Circuito Tecnopolo Sur No 112, Col. Fracc. Tecnopolo Pocitos, Aguascalientes, Aguascalientes C.P. 20313, México

ARTICLE INFO

Article history:
Received 2 September 2019
Revised 17 January 2020
Accepted 30 January 2020
Available online 1 February 2020

Keywords:
Opinion mining
Sentiment analysis
Deep learning
Evolutionary algorithms
Machine learning
Intelligent learning environments

ABSTRACT

This paper presents a comparison among several sentiment analysis classifiers using three different techniques – machine learning, deep learning, and an evolutionary approach called EvoMSA – for the classification of educational opinions in an Intelligent Learning Environment called ILE-Java. To make this comparison, we develop two corpora of expressions into the programming languages domain, which reflect the emotional state of students regarding teachers, exams, homework, and academic projects, among others. A corpus called sentiTEXT has polarity (positive and negative) labels, while a corpus called eduSERE has positive and negative learning-centered emotions (engaged, excited, bored, and frustrated) labels. From the experiments carried out with the three techniques, we conclude that the evolutionary algorithm (EvoMSA) generated the best results with an accuracy of 93% for the corpus sentiTEXT, and 84% for the corpus eduSERE.

© 2020 Elsevier Ltd. All rights reserved.

1. Introduction

The World Wide Web and its applications in social networks, discussion forums, blogs, and others allow users to generate huge amounts of information expressing their points of view and emotions about different products, brands, social events, and more. The opinions that users express in those applications are able to have a great influence on readers, suppliers of products and services, online educational platforms, and so on.

In the field of education, particularly during the learning process in Intelligent Learning Environments (ILE), it is important to consider students' opinions (feedback) and affective states in order to modify and improve learning contents. This feedback can help teachers understand student behavior, and can be useful in refining the educational content of the ILE (Rowe A.D, 2017; Rowe A.D & Fitness, 2018).

Automatic processing of student opinions is a complicated task due to the nature of the language used by young people. A simple example is the need to understand the terms and abbreviations that they use to communicate with each other on the Web. To deal with the task of automatic classification of opinions, different techniques of opinion mining or sentiment analysis can be used (Zhang L.& Liu, 2016), which include other areas such as natural language processing (NLP), machine learning (ML) in combination with neural networks of deep learning as well as evolutionary algorithms, among others (Tul et al., 2017).

Traditionally, experts are used to implement opinion mining using machine learning techniques, obtaining acceptable results, but with the popularity of neural networks with deep learning, experts are moving to use this approach to get better results. However, this technique faces a serious problem that affects its efficiency: the optimization or tuning of many hyperparameters. This process is commonly carried out through trial and error or based on previous and similar works. Due to the dimensions of the hyperparameters (from a hundred to a thousand), when using this technique, it is difficult to reach a quasi-optimal topology that ensures the improvement in the accuracy of recognition. However, this problem has been approached by using a genetic algorithm (Zatarain Cabada, Rodriguez Rangel, Barron Estrada, & Cardenas Lopez, 2019). In this approach the algorithm defines and iterates a group of individuals named a population to obtain the best model for a problem. An individual is a specific representation of a solution (a chromosome) represented by some genes that define the CNN topology.

In this research work, we describe the process of creating a corpus for the educational domain in the field of programming

[*] Corresponding author.
E-mail addresses: lbarron@itculiacan.edu.mx (M.L. Barrón Estrada), rzatarain@itculiacan.edu.mx (R. Zatarain Cabada), raul.oramas@itculiacan.edu.mx (R. Oramas Bustillos), mario.graff@infotec.mx (M. Graff).

https://doi.org/10.1016/j.eswa.2020.113265
0957-4174/© 2020 Elsevier Ltd. All rights reserved.

languages. We also present a comparative study for different models of Machine Learning, Deep Learning, and one Evolutionary Algorithm, in addition to showing an application of the results of this study. The main novelty of this work is the comparative study between different classifiers including one using an evolutionary method, in an educational context through two new datasets of educational opinions labeled with learning-centered emotions. To the best of our knowledge, there is no research work in the mining of opinions labeled with learning-centered emotions. In addition, one important contribution is the integration of the best classification model into an intelligent learning environment.

We organized the rest of the paper as follows: Section 2 discusses the background and related work. Section 3 describes our contribution in depth. Section 4 describes and discusses all experimental results. The model integration to a learning environment is presented in Section 5. Finally, Section 6 presents discussions and conclusions.

2. Related work

There are several AI techniques that tackle the problem of classifying opinions. Traditional approaches such as machine learning or neural network can solve this classification problem as well as deep learning combined with evolutionary algorithms to optimize hyper-parameters.

In this section, we present important work related to the three topics that influenced our research.

2.1. Deep learning frameworks or platforms

Neural Networks of Deep Learning is a research area of fast growth due to the availability and ease of use of many open source frameworks, and to the trends in the low price of hardware resources. Moreover, different studies (Tul et al., 2017; Zhang, Wang, & Liu, 2018) show that it is possible to get the same or better results using Deep Learning than those implementing traditional techniques such as Machine Learning. However, it is important to bear in mind that traditional techniques are also improving and evolving and have obtained competitive results.

Frameworks provide prefabricated solutions for most common problems in Deep Learning. Table 1 shows information of several frameworks that are available so far and do not require proprietary licensing. Table 1 contains in each column: the name of the framework and its year of release, the programming language used for the implementation, the platform or operating system in which it is executed, and the available programming interfaces.

Some frameworks like DL4J, CNTK, Caffe, TensorFlow, and BigDL are adequate for industry due to their speed, scale and stability. Other frameworks like PyTorch, MXNet, TensorFlow, and CNTK are indicated from research organizations due to their flexibility and easy debuggability. For DL beginners we recommend starting with Keras because it is easy to configure, compile and train DL models. Frameworks like Caffe, MXNet, PyTorch, CNTK are better for computer vision. Finally, every framework tries to use the advantages offered by available hardware, and other frameworks focus on other aspects such as portability and ease of use. Therefore, each framework has advantages and disadvantages in terms of quality attributes such as portability, configurability, ease of use, APIs available in several programming languages, etc.

2.2. Opinion mining with deep learning

The implementation of different models of Neural Networks of Deep Learning, such as the Convolutional Neural Networks (CNN), Recursive Neural Networks (RNN), Long Short-Term Memory Networks (LSTM), among others, efficiently carry out the task of opinion mining. CNNs were initially applied to artificial vision problems (Sermanet, Chintala, & LeCun, 2012) but they have also proven to be effective in Opinion Mining tasks. For example, in Zhang, Zhao, and LeCun (2015), the authors describe how they performed an empirical exploration using convolutional networks (ConvNets) for text classification. They also made comparisons with traditional models such as bag-of-word, n-grams and their variations of TFIDF, and deep learning models such as word-based ConvNets and RNN, concluding that there is not a single machine-learning model that can work for all kinds of datasets.

In Kim (2014), classification and sentiment analysis at sentence level is carried out using a CNN with a convolution layer and pre-trained word vectors, on 100 billion words from Google News. This model, despite being simple, achieved excellent results for different corpora. For example, in Stanford Sentiment Treebank (SST-2), the experiments with CNN-Multichannel model reached an 88.1% accuracy. The vectors of pre-trained words were first proposed in Mikolov et al. (2014) and Bengio, Courville, and Vincent (2013) and together with the CNN, they work as extractors of universal features that encode semantic features of words, that can be used for several classification tasks. This research comes with the conclusion that an unsupervised pre-training of word vectors is an important component in deep learning for NLP. Also, in Kalchbrenner, Grefenstette, and Blunsom (2014) they used a Dynamic Convolutional Neural Network (DCNN) for semantic modeling of sentences for classification. In this work, authors mention that the modeling tasks involve NLP and Sentiment Analysis, among other things.

They extract the features of the words based on n-grams to capture the semantic relationships of the text. This DCNN model uses a bag of n-gram words to capture the semantic relationships between the words. The network achieves a high performance on question and sentiment classification without requiring external features as provided by parsers or other resources.

In dos Santos and Gatti (2014) the Convolutional Neural Network (CharSCNN) extracts the relevant features of simple words and sentences such as Twitter messages, using pre-trained word vectors. The authors tested their model with two datasets from different domains: the Stanford Sentiment Treebank (SSTb), which contains sentences from movie reviews, and the Stanford Twitter Sentiment corpus (STS), which contains Twitter messages. For the SSTb corpus, the sentiment prediction of a single sentence in a binary classification (positive/negative) reached an accuracy of 85.7%, and a sentiment prediction accuracy of 86.4% for the STS corpus.

It is important to mention that theoretically, CNN and DCNN techniques could process a whole sentence of arbitrary length by encoding the context cyclically, but the length of reachable context is often limited when using stochastic gradient descent. Besides that, CNN and DCNN architectures are not powerful enough to deal with complex sentiment expressions. Also, fixed input limits the network's ability of learning task-specific representations and simple additive combination of hidden activations; and input activations have difficulty capturing more complex linguistic phenomena. That's why researchers are also looking for another approach by mean of simulating the interactions of words during the compositional process, using models such as LSTM recurrent neural networks for sentiment classification.

In Wang, Liu, Sun, Wang, and Wang (2015) the authors used LSTM to predict the sentiment of phrases on Twitter. In their experiments, they used the STS corpus to compare different Machine Learning models such as support-vector machines (SVM), Bernoulli Naives Bayes, and RNN with different configurations. They obtained an accuracy of 87.2% with better results than the other models. It is important to note that the research in this work focuses on sentiment classification on Twitter by means of simulating the interactions of words during the compositional process. When Comparing

Table 1
Frameworks available for deep learning.

	Programing Language								Platform								Programing interfaces																
Framework Year	Scala	C++	C	Java	Python	Small C++ core library	CUDA	Lua	Apache Spark	Linux	MAC OS	Windows	iOS	Android	AWS*	Java Script**	Scala	Python	Python (keras)	MATLAB	C++	Java	Clojure	Kotlin	R	C#	Go	Perl	Java Script	C	Lua	Lua (JIT)	Julia
Apache MXNet 2015						X				X	X	X	X	X	X	X	X	X		X	X	X			X		X	X	X				X
Deeplearning4j 2004		X		X						X	X	X		X			X		X			X	X	X									
Microsoft Cognitive Toolkit 2016		X								X	X	X							X		X	X				X							
Torch 2000			X					X		X	X	X	X	X																X	X	X	
Caffe 2013		X								X	X	X					X		X	X													
BigDL 2016	X								X								X	X															
DyNET 2016		X								X	X	X						X			X												
Keras 2015					X					X	X	X						X							X								
PyTorch 2016			X		X		X			X	X	X						X															
TensorFlow 2015		X			X		X			X	X	X						X															
PaddlePaddle 2018		X			X					X	X	X						X															
Chainer 2015					X					X	X	X						X															
Theano 2010		X							Cross-platform											X													
Dlib 2002		X							Cross-platform													X											

*Amazon Web Services (AWS).
**The framework allows execution using JavaScript.

LSTM with different machine models that rely on lexicons and extracted entities, the researchers discover that the ability of LSTM getting access to longer-distance context is not the determinant of improvement, but the capabilities of learning appropriate representations of sentiment information through compositional and sequential manner. This is what make it able to cover the diversification of Twitter sentiment expressions.

In Wang, Yu, Lai, and Zhang (2016) the authors proposed a dimensional sentiment analysis to recognize continuous numerical values in multiple dimensions such as the valence-arousal (VA) space, instead of using a categorical approach with discrete classes (e.g. negative-positive) or multiples categories such as Ekman's six basic emotions (anger, happiness, fear, sadness, disgust, surprise), with the aim of providing a more fine-gained sentiment analysis in text. They used a CNN with another LSTM network consisting of two parts: a regional CNN and a LSTM, to predict the ratings of texts with a valence scale. The regional CNN, unlike a conventional CNN that considers a complete text as input, uses individual sentences as regions so that it can be weighted according to the valence scale. Afterwards, all regions using LSTM for prediction with the valence scale, sequentially integrate regional information. Their experiments show good results against some methods for VA prediction based on lexicons, regression, and those based on an Artificial Neural Network (ANN).

In Qian, Huang, Lei, and Zhu (2016), authors address that there is a variety of neural networks models that depend on expensive phrase-level annotation, most of which has a remarkably degraded performance when trained with only sentence-level annotation or with not fully employing linguistic resources. They propose linguistically regularized LSTMs for sentiment classification with annotations at the sentence level that employ regularizers for modeling the linguistic role of sentiment lexicons, negation words, and intensity words. Results showed 82.1% accuracy for Movie Review (MR) datasets and they proved that their models were able to address the linguistic role of sentiment, negation, and intensity words. Also, in the work of Nguyen and Nguyen (2017), the authors focused on the challenges of Tweet-level sentiment classification in Twitter social networking, exploiting syntax, semantic, sentiment, and context in Tweets. They used a lexical-based approach to apply semantic rules and then used a DeepCNN with pre-trained vectors at the character level to capture morphological information of each word to determine how those words are formed and their relation to each other. They then used a Bi-LSTM bidirectional network that produces the feature representation of the whole sentence from pre-trained vectors at the word level.

In Poria, Cambria, Hazarika, and Vij (2016) the authors used a pre-trained CNN that is capable of extracting sentiment, emotion, and personality features from sarcasm detection. The detection of sarcasm is a key task for many NLP tasks. In Opinion Mining tasks, sarcasm can change the polarity of a “seemingly positive” sentence and, therefore, adversely affect the performance of polarity detection. To date, most approaches to the detection of sarcasm have treated the task primarily as a problem of categorization of the text. Sarcasm, however, can be expressed in a very subtle way and requires a deeper understanding of natural language.

2.3. Text classification based on evolutionary algorithms

Evolutionary Algorithms (EAs), and Genetic Programming (GP), have been used on different Natural Language Processing tasks. Perhaps one of the first appearances of GP is in the task of grammatical induction (Smith, Smith, & Witten, 1995), and parsing (Spector, Langdon, & O'Reilly, 1999). Moving closer to the aim of this study, Escalante et al. (2015) proposed the use of GP to optimize the weights of the bag-of-words model in a text classifier scenario. The idea was to evolve a function that combines traditional term weighting schemes such as Boolean, Term-Frequency, among others using arithmetic functions as well as transcendental function. EAs have been used in text classification problems. In Diab and El Hindi (2017) the authors optimized the parameters of a Naive Bayes classifier using Genetic Algorithms (GA), Differential Evolution, and Simulated Annealing, enhancing significantly the Naïve Bayes classification rate, with a clear limitation of longer training times.

Specifically, EAs have been used on sentiment analysis (SA). In Carvalho, Prado, and Plastino (2014) the authors used GA to find a subset of paradigm words that improve the classification of either positive or negative polarity in a sentence, with a clear domain classification bias, where movie reviews achieved less accuracy than banks and automobile reviews. Continuing with the use of GA, Keshavarz and Abadeh (2017) posed the SA problem as an optimization one where the goal was to find the optimum sentiment lexicon, obtaining higher accuracy due to the better understanding of specific language and culture of twitter users. GP has been used as a replacement of a linear SVM in Graff et al. (2017). This replacement is challenging for GP in that the texts are represented using a bag-of-word model, producing a high-dimensional vector space, proposing a novel GP system called Root Genetic Programming.

The use of ensemble to create a sentiment analysis classifier has been studied in the EAs community. Winkler et al. (2015) created an ensemble of different machine learning algorithms (where GP is one of them) to identify the sentiment of sentences. Their main contribution was to create binary and multi-class models to represent the relations between words on sentences in an Amazon corpus written in German. On the other hand, EAs have been used to combine the output of different classifiers. López, Valdivia, Martínez-Cámara, Luzón, and Herrera (2019) proposed an ensemble composed of off-the-shelf sentiment methods to produce competitive classifiers. The authors tested the classifiers on a set of 13 benchmark problems where one of the main limitations was the ability of generalization in the machine learning algorithms and lexical coverage of linguistic resources. Onan, Korukoğlu, and Bulut (2016) used a multi-objective differential evolution to generate a voting scheme composed of traditional classifiers like Naive Bayes, Logistic Regression, and SVM, among others. The objectives being optimized were precision and recall. Besides the use of EAs, GP has been used as a stacking mechanism where the idea is to combine the output of different text classifiers and text representations to produce a competitive text classifier.

2.4. Opinion mining in intelligent tutoring systems

An application of opinion mining in learning environments is the automatic evaluation of the students' opinions about the material provided by the learning system, as well as the usability of the educational tool. The aim of using opinion mining is to obtain information that allows teachers to understand the needs of students in order to determine at a given moment whether it is necessary to incorporate changes in material, exercises, tests, teaching strategies, or any other aspect during the learning process. Next, we present a brief description of the work related to this field.

In Kechaou, Ammar, and Alimi (2011) they apply a method to extract opinions from e-learning blogs and use a model based on SVM and Hidden Markov Models (HMM) to obtain knowledge about the opinions of users and build an evaluation with respect to it. The main contribution of this work is to blend sentiment classification with an e-learning platform.

In El-Halees (2011) the author use a model to extract knowledge of opinions to improve the evaluation of a course using the opinions of students published in Internet forums, discussion groups, and blogs. They classify students' opinions as positive or negative using Self-Learning methods. Then, he/she extracts opinions, such as the teacher, exams and resources, from the content generated by the user for a specific course and grouped these features for each course. The most important contribution of this research work is a novel way of improving course quality.

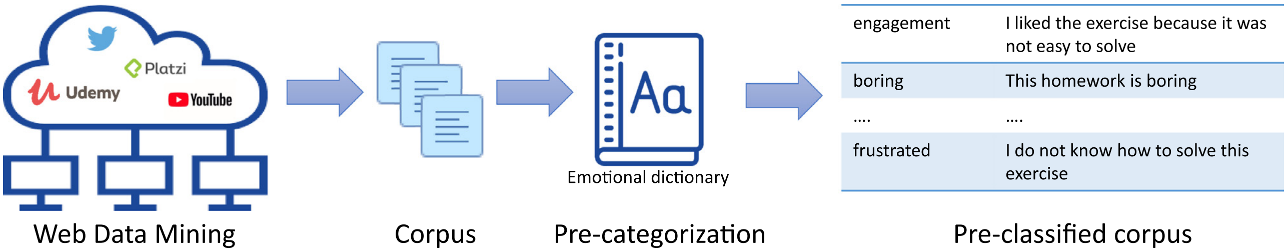
In Altrabsheh, Gaber, and Cocea (2013) the authors present a study where they analyzed the feedback of the students who took a course via Twitter messages, and using Naïve Bayes algorithms and SVM to identify the students' positive or negative sentiment. The aim was to collect the comments and analyze these data to help improve the teaching process.

In Ortigosa, Martín, and Carro (2014) the authors present an application called SentBuk that runs on Facebook and extracts information about user sentiment automatically and non-intrusively. It classifies students' opinions as positive, negative, or neutral to detect significant emotional changes. SentBuk system records the information provided by the students, which enriches the system, so that it proposes an adapted task to the student, based on his/her emotional state. These emotional states are useful feedback for the teacher. The classification method implemented in SentBuk follows a hybrid approach: it combines lexical techniques with machine learning. The authors mention some limitations that make sentiment analysis harder: the bounds of the domain are not defined, and the messages in Facebook are mostly informal, requiring the use of additional tools as language detectors or spelling checkers.

In Wen, Yang, and Rose (2014) the authors analyze the opinions of the students written in discussion forums of Online Massive Courses (MOOC) in order to monitor those opinions using simple techniques of opinion mining and lexicon dictionaries. In this study, they report a correlation between the feeling index (measured according to the daily publications in the forum) and the number of students who abandon the course every day.

In Altrabsheh, Cocea, and Fallahkhair (2014) the student's emotions are used to address problems such as confusion and boredom affecting student participation. They used a Naïve Bayes, Complement Naive Bayes (CNB), Maximum Entropy, and Support Vector Machines algorithms to classify emotions.

In (Chaplot, Rhim, & Kim, 2015) they present an algorithm based on a Neural Network to predict the student's dropout in MOOCs using sentiment analysis and show the importance of students' affect in this task.



engagement	I liked the exercise because it was not easy to solve
boring	This homework is boring
….	….
frustrated	I do not know how to solve this exercise

Fig. 1. Corpus construction.

In Kumar and Jain (2015) the authors collect students' opinions in the form of running text and then perform sentiment analysis to identify important aspects along with guidance using supervised and semi-supervised learning techniques.

In the work of Dhanalakshmi, Bino, and Saravanan (2016) they find the polarity of the students' comments based on predefined features of teaching and learning using opinion mining with supervised automatic learning algorithms such as SVM, Naïve Bayes, nearest neighbors (KNN), and neural networks. The study involves the application of a combination of machine learning and NLP techniques in student feedback data. They compared the results of different algorithms to find the best performance with respect to several evaluation criteria.

In Rani & Kumar (2017) the authors applied NLP and machine learning techniques to student comments to help university administrators and teachers address areas with problems in teaching and learning. The system analyzes students' comments on course surveys and online sources to identify the sentiment polarity, the emotion expression, and student satisfaction.

In conclusion, there are different approaches that exploit syntax, semantic, sentiment, and context for modeling and classification of opinion sentences, trying to cover the diversity of sentiment expressions by creating models that rely on lexicons and extracted entities, or by employing regularizers for modeling the linguistic role of sentiment lexicons, negation words, and intensity words. A novelty approach is the use of Evolutionary Algorithms on a variety of text classification problems and sentiment analysis applications; these works are focused on creating binary and multiclass models to represent the relation between words in sentences, searching for ways to find the optimum sentiment lexicon and to improve the accuracy of language and culture understanding in text. The blending of opinion mining and sentiment analysis in Intelligent Tutoring Systems and similar applications allows for knowledge to be obtained about the opinion of users and their emotions. Most e-learning platforms use model-based machine learning strategies, like SVM and Hidden Markov Models (HMM), but there is also the use of deep learning methods (RNN, LSTM, etc.) that give the same or better results than using traditional Machine Learning. Platforms or frameworks like TensorFlow, Caffe, and MXNet offer different advantages and disadvantages in terms of quality attributes that make them more adequate for industry or research applications. These frameworks have also created the pathways for new evolutionary optimization algorithms for these deep learning algorithms in both training and data preparation, which has also created many new ways to approach the NLP problem.

On the other hand, given the success of word embedding and pre-trained deep learning architectures, it is common practice to tackle NLP tasks using as many resources as possible. However, there are problems where the use of different resources (e.g., word embedding, pre-trained models, lexicons, among others) does not make a significant impact on the performance of the final model. The datasets created in this contribution belong to these problems where the use of additional resource does not have a considerable impact on the performance.

3. Methodology

Our methodology follows the traditional scheme used to develop machine learning models, including in the end the integration of the best model into an application. The main steps within this methodology are: creation of the corpus to train machine learning models, construction of classification models using different learning strategies, training and testing with the corpus of the different classification models, selection of the best model, and finally, integration of the model into an application.

We start by describing the creation process of two corpus: SentiTEXT and eduSERE. The first one has labels that denote a positive or negative polarity, and the second one has labels that represent learning-centered emotions such as engaged, excited, bored, and frustrated.

3.1. Creation of the corpus

For the construction of the sentiTEXT and eduSERE corpus, we first established that students' expressions should reflect positive or negative opinions, as well as learning-centered emotions (engaged, excited, bored, or frustrated). Therefore, these expressions should reflect opinions in an educational context (particularly in the domain of programming languages) with opinions towards teachers, students, learning objects, academic projects, moods, among others. Fig. 1 illustrates the process of Corpus construction. Each step is described in the next subsections where we explained how we collected the students' expressions.

3.2. Data mining

In a first approach, we used the Web Scrapping technique to obtain users' comments from educational platforms Udemy, Platzi, and YouTube through its HTML code. We also used the API of Twitter to get learning related content from that social network. The mining of Twitter was completed based on keywords such as *teacher, exam, task, laboratory practice, failing, programming, algorithm*, among others. All these comments expressed by users constituted the datasets SentiTEXT and eduSERE.

Next, we developed a Web application named Educational Resource Evaluation System (SERE) (Barrón-Estrada, Zataraín-Cabada, Bustillos & Ramírez-Ávila, 2017) to capture the opinions written by students enrolled in several courses related to programming such as introduction to programming, object-oriented programming, data structures, etc., in the semester January-July 2017 at Instituto Tecnológico de Culiacán (TecNM).

In this process of collecting phrases, it was necessary to eliminate repeated phrases and to discard those phrases with words not related to the educational context of Mexico or those that did not express any polarity or educational emotion. We obtained 46,322 sentences. Table 2 shows an extract of some of the phrases that we discarded.

The phrases in lines 1 and 2 are identical except for their URL and do not express any emotional content. The sentence in row

Table 2
Example of discarded phrases.

Num.	Phrase
1	#BuenosDias (Good Morning) ... https://t.co/uKub53lEOV
2	#BUENOSDIAS (GOOD MORNING) ... https://t.co/wFnv4bfZ1q
3	ACTUALIZACIÓN CENTRAL DE APUNTES agregamos: (Central update of notes we add:)
4	Pibes dejan un video en un telo (Children let a video in a “telo”)
5	Java es un lenguaje de programación (Java is a programming language)

Table 3
Representative examples of dictionary SentiDict.

Word	Emotion	Polarity
Aprender (Learn)	Engaged	Positive
Genial (Cool)	Engaged	Positive
Entretenido (Amused)	Excited	Positive
Eufórico (Euphoric)	Excited	Positive
Olvidar (Forget)	Boring	Negative
Cansado (Tired)	Boring	Negative
Complicado (Complex)	Frustrated	Negative
Inseguro (Unsure)	Frustrated	Negative

3 does not express any emotion, while the sentence in row 4 contains two common words in the Spanish spoken in Argentina (pibes and telo). Although the phrase of row 5 fits into the domain of programming language vocabulary, it expresses a fact but not an emotion.

3.2.1. Building an emotional dictionary

Once we selected different phrases to build both corpora, the next step was categorizing each sentence with labels *positive* and *negative* for the corpus sentiTEXT and labels *engagement, excitement, boring*, and *frustration* in the case of the corpus eduSERE.

To establish which word expresses a polarity or a learning-centered emotion, we built an emotional dictionary called SentiDict. We developed SentiDict using three dictionaries. First we used MADS dictionary (Hinojosa, Pozo, & Montoro, 2016) which measures two emotional dimensions (valence and arousal) and five discrete emotional categories (happiness, anger, sadness, fear, and disgust). Second we consulted a dictionary which is marked with emotions and weighted for Spanish (Rangel, Guerra, & Sidorov, 2014), and contains words grouped into the basic emotional categories *joy, anger, fear, sadness, surprise* and *repulsion*, and finally we also looked up a dictionary of emotions, activities, and behaviors (Pereira & Valcárcel, 2016).

We adapted the words from dictionary SentiDict to the Mexican context. We labeled the emotional categories *joy, surprise*, and *happiness* as positive and *anger, fear, sadness*, and *repulsion* as negative. For the categorization of learning-centered emotions (engagement, excited, boring, and frustration) we used the definitions of D’Mello (Baker, D’Mello, Rodrigo, & Graesser, 2010) as a guideline. This process produced a dictionary with 2200 words adapted to the Mexican context.

The dictionary has 1310 words with label (polarity) *positive* and 890 with label *negative*; 95 have label (emotion) *engaged*, 57 with label *excited*, 53 with label *bored*, and 95 with label *frustrated*. Table 3 presents some words of dictionary SentiDict and their respective labels for emotion and polarity.

From the previous table, the word *Learn*, represents an affective state (emotion) of engagement, and it is also a positive word. Another example is the word *Tired*, representing an affective state of boredom and a negative polarity.

3.2.2. Pre-classification of the corpus

We implemented a simple classifier based on the words of dictionary SentiDict in order to pre-classify the phrases collected previously. We applied an algorithm based on a word count to determine the polarity and the learning-centered emotion of a sentence.

Given an opinion, for example, “it is easy to learn”, the algorithm obtains the individual components called tokens, and calculates the occurrence of each word in SentiDict. For example, if the algorithm finds more words labeled with emotion *engaged*, then it classifies the sentiment of the phrase as positive and engaged. With this algorithm, we pre-labeled the collected opinions building the sentiTEXT and eduSERE corpus. The idea of this simple classifier was to establish a method to pre-label the opinions collected previously. Subsequently, a computer science expert reviewed and eliminated those opinions that were difficult to classify. As a result, we had a balanced corpus for polarity classification (sentiTEXT) with 24556 opinions, where 12278 are positive opinions and 12278 are negative opinions. In addition, we had an unbalanced corpus for emotion classification (eduSERE) with 12084 opinions, with 5599 positive opinions that were labeled as *engaged* (including the emotion *excited*), and 6483 negative opinions which 3238 has label *bored,* and 3245 were labeled as *frustrated*.

3.3. Construction of classification models

In this section we will describe how we carry out a preprocessing of the corpus for cleaning it. In addition, we will present the details of the models selected to implement the classification of sentiments and emotions.

3.3.1. Preprocessing the Corpus

One of the most important factors in opinion mining is to have a non-noisy corpus. A noisy corpus refers to text with punctuation marks, numeric values, links, URLs, etc. The elimination of these elements in the text would increase the reliability of the different text classifiers. However, it is essential to keep in mind that elements, such as emoticons (e.g. “:)”), are important for the sentiment analysis, since they add an emotional charge in the expression or phrase.

To maintain an adequate balance, the corpus must go through a preprocessing stage, which represents a cleaning process and corpus preparation for later use. The preprocessing allows the normalization of the corpus and helps improve the performance of the text classifiers. For basic tasks of preprocessing, we considered the following:

a) Punctuation marks. Usually users who write an opinion omit or abuse some punctuation marks such as period, comma, question mark, or exclamation mark. To normalize these sentences, we used regular expressions deleting all extra punctuation marks.
b) Common abbreviations (slang terms). It is common for users to include abbreviations of their opinions. Given a dictionary of slang terms, some terms are translated, but not all to preserve the grammatical content of an opinion. For example, we replace the term “KO” with the sentence in Spanish “Estoy muerto de cansancio” ("I am dead" in English) and term “a2” with “adios” ("goodbye" in English).
c) Interjection normalization. we also use regular expressions to normalize elements such as ‘hahaha’, ‘jajjjaja’, ‘jijiji’, as they cause dispersion problems.

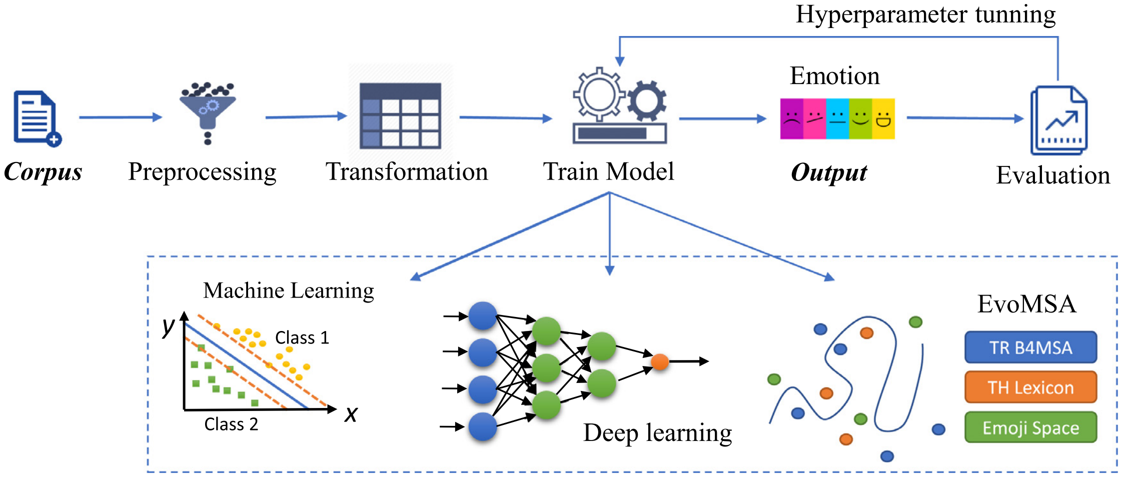


Fig. 2. Basic steps for configuring, training, and evaluating the models for Opinion Mining.

d) Emoticon replacement. As in the case of interjections, users often express their moods using emoticons with different connotations. We use a special emoticon dictionary to identify and replace the most frequent emoticons. For example, the text “happy” replaces the emoticon “:)”.
e) Link simplification. Often, an opinion includes a link to external resources, such as images or addresses to other web pages. We use regular expressions to detect those links, since they do not represent relevant information.
f) Stop words. We consider articles, pronouns, prepositions, among others, as *Stop words* because they do not provide information nor have an associated emotional charge to the classification process.
g) Stemming. It consists of taking each word to its root to group words with a similar meaning. For example, in this process we consider the words “ran” and “running” as equals because their roots (run) are the same.

3.3.2. Learning models

We use a four-step process to implement the learning models. Fig. 2 shows the complete process, representing the steps: preprocessing, transforming, training with one of the methods or models (Automatic Learning, Deep Learning, and the Evolutionary algorithm EvoMSA), and evaluating the models.

3.3.2.1. Machine learning model. The first model uses Machine Learning, which is one of the most popular techniques for opinion mining. In our research work, we applied several classification algorithms like Bernoulli Naïve Bayes, Support Vector Machine (SVC), Linear SCV, and Random Forest, from the Python library *scikit-learn*.

First, the corpus goes through a pre-processing step where we apply some basic NLP tasks like eliminating punctuation marks, stop words, and reducing the words to their root (stemming). Second, for each phrase, we obtain the corpus vocabulary (non-repeated words) and for each individual word or token, we transform it into a numerical representation, creating a TFIDF matrix. Third, we configure the parameters (size of the vocabulary, percentage of phrases to be used for training, etc.) of the classifying algorithm, to check if they are considered unigram or bigrams. Fourth, we proceed to train the classification algorithm and test the results based on an evaluation metric. If we doní obtain satisfactory results, the parameters are changed, and the classifying algorithm is re-trained until we obtain the desired results.

3.3.2.2. Deep learning model. For Deep Learning, we used the Keras framework available for Python 3. The step for preprocessing is like the previous section; the difference with it is that DL models use the corpus with the representation through a bag of words (Bag of Words, BoW for short). Bag of Words is a way of extracting characteristics from the text of the corpus (like a TFIDF matrix) that describes the occurrence of words within a text document.

For configuring the DL models, we consider the number of layers, the number of input and output neurons per layer, the layer type (Dense, CNN, LSTM), the activation function for each layer (e.g. relu), the loss function (e.g. binary cross entropy) and the optimization methods that are associated with the learning rate.

Another important aspect that we had to consider for configuring the DL model was the learning rate. There are different optimization methods such as Momentum, Adagrad, Adam, or RMSProp. For this work, we decided to use the Adam method, since it saved us the manual choice of the initial learning rate. Fig. 3 shows the configuration of a DL model that uses a CNN neural network and another that uses an LSTM neural network.

The neural network receives as input a text in the form of a bag of words, which the convolutional layer process. This layer returns a one-dimensional vector, which another layer of type LSTM process. Its output is processed by a fully-connected neural network which returns as a result the emotion *positive* or *negative*, if it is a model that classifies polarities, and *engaged*, *bored*, or *frustrated* if it is a model that classifies learning-centered emotions.

3.3.2.3. The evolutionary model. EvoMSA (Graff, Miranda-Jiménez, Tellez & Moctezuma, 2018) is a multilingual sentiment classifier based on Genetic Programming (Poli, Langdon, McPhee, & Koza, 2008). It works by combining the output of different text classifiers to perform the final prediction. EvoMSA can be seen as stack generalization (Wolpert, 1992) where the base systems are text classifiers and the final prediction is performed by a GP classifier specifically EvoDAG (Graff, Tellez, Miranda-Jiménez & Escalante, 2016). Furthermore, the text classifiers used are pairs of text model and classifier. In general, a text model is a function that transforms a text into a vector space. Consequently, given a set of pairs, text and label, it is straightforward to train a classifier when firstly all the texts are represented in a vector space, and, then, these pairs are used to train a classifier such as neural networks, logistic regression, and SVM, among others.

EvoMSA uses a linear SVM classifier and implements different text models. The basic one is B4MSA (Tellez et al., 2017;

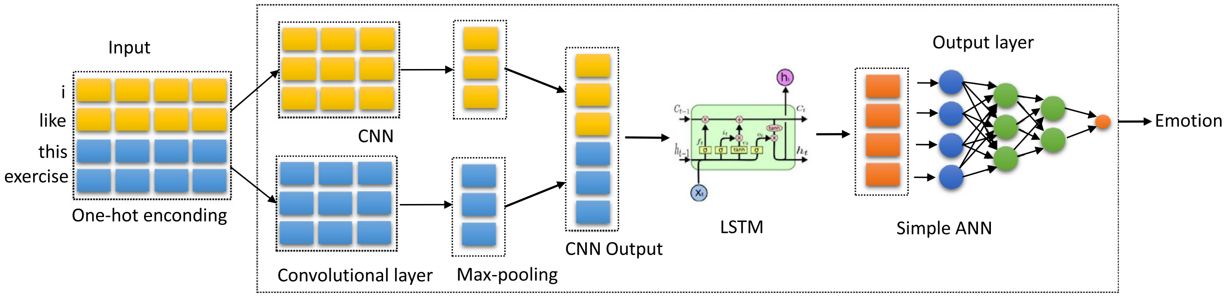


Fig. 3. A model of DL with CNN and LSTM layers.

Table 4
Experiments made with fourteen different classifiers.

Number	Type	Model	Accuracy	
			SentiTEXT	EduSERE
1	Evolutionary Algorithm	EvoMSA	93%	84%
2	Machine Learning	Multinomial NB	87%	79%
3	Machine Learning	KNN	79%	68%
4	Machine Learning	Decision Tree	85%	72%
5	Machine Learning	B4MSA	92%	83%
6	Machine Learning	Bernoulli NB	87%	76%
7	Machine Learning	SVC	90%	79%
8	Machine Learning	Linear SVC	90%	79%
9	Machine Learning	Random Forest	89%	77%
10	Deep Learning	LSTM_3b	89%	79%
11	Deep Learning	CNN_5a	91%	68%
12	Deep Learning	CNN_10a	90%	80%
13	Deep Learning	CNN_LSTM_7a	88%	74%
14	Deep Learning	BERT	93%	83%

Tellez, Moctezuma, Miranda-Jiménez & Graff, 2018) which can be considered as a baseline to create sentiment analysis classifiers. B4MSA uses a set of simple techniques to transform a text into a vector space; it starts by first applying text transformations such as replacing numbers, users, and URL by words *_num, _usr*, and *_url*, respectively. Then it transforms the resulting text by removing diacritics and duplicated letters. After these transformations are performed the text is split into tokens and words; the tokens were skip-grams of two words with a skip of one word, e.g., in the sentence “an excellent lecture on java” the first skip-gram is “an lecture”, the second is “excellent on”, the third is “lecture java”, and so on. The rest of the tokens are q-grams of characters with size 2, 3, 4, 5, and 6. Finally, the tokens are the coordinates in a vector space and the weights are computed using the term frequency–inverse document frequency.

The second text model is EvoMSA’s implementation of DeepMoji (Felbo, Mislove, Søgaard, Rahwan & Lehmann, 2017). The idea is to represent each text in a vector space of the 64 most common emoticon or emoji, that is, given a text then predicts which would be the most probable emoji of that text. In order to create this space, 3.2 million of tweets were selected from a collection of approximately 3.7×10^9 Spanish tweets. The selected tweets correspond to 50,000 tweets per emoji and were selected using some simple rules; all tweets have only one type of emoji. The tweets were selected uniformly from the collection trying to avoid seasonal effects, and all re-tweets were removed.

The last text model used is a Lexicon-based model were the lexicons are a set of sentimental words. These sentimental words are positive and negative words, and the idea is, in a given text, to count the number of positive and negative words. Consequently, the resulting output is a vector in two dimensions: one corresponding to the positive words and the other to the negative words.

4. Training and testing the different classification models

We trained and evaluated fourteen models to obtain the best model for the detection of emotions in the student opinions. These models are the evolutionary algorithm EvoMSA, eight Machine Learning classifiers, and five Deep Learning classifiers. We describe in the next section the results produced by each classifier.

4.1. Experiment results

A metric is a function used to evaluate the performance of a model. The common evaluation metrics in the field of machine and deep learning are accuracy, logarithmic loss, area under the ROC curve, confusion matrix, among others. For this research work, we decided to use accuracy because it is one of the most used metrics. The accuracy is the ratio between the number of correct predictions and the total number of input samples. For example, if the sample number of the test is 1000 and the model classifies 952 of them correctly, then the model’s accuracy reported is 95.2%.

For machine learning classification we decided to use the Bernoulli and Multinomial Naïve Bayes methods, that are based on the Bayes theorem, k-nearest neighbors (KNN), Support Vector Machine and Linear Vector Machine that represent data as vectors in a hyperplane, Decision Trees and Random Forest that perform classification based on tree-like model of decisions, and B4MSA. We also evaluated five different deep learning architectures: two CNN models with different layers, one LSTM model, one hybrid between a CNN and a LSTM model, and model BERT (Bidirectional Encoder Representations from Transformers).

To test the models, we created a procedure to select the opinions of the corpus in a random way, and thus have the same number of opinions with positive and negative polarity for corpus sentiTEXT, and opinions with engaged, bored, and frustrated emotion

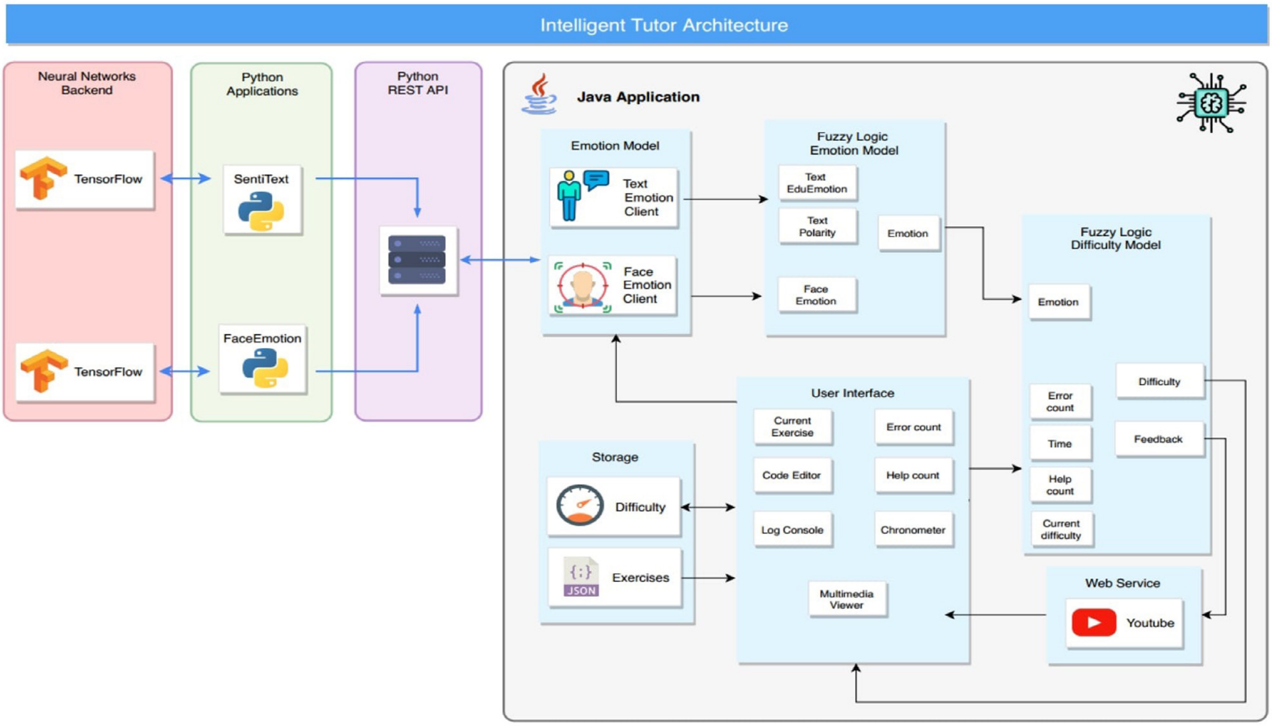
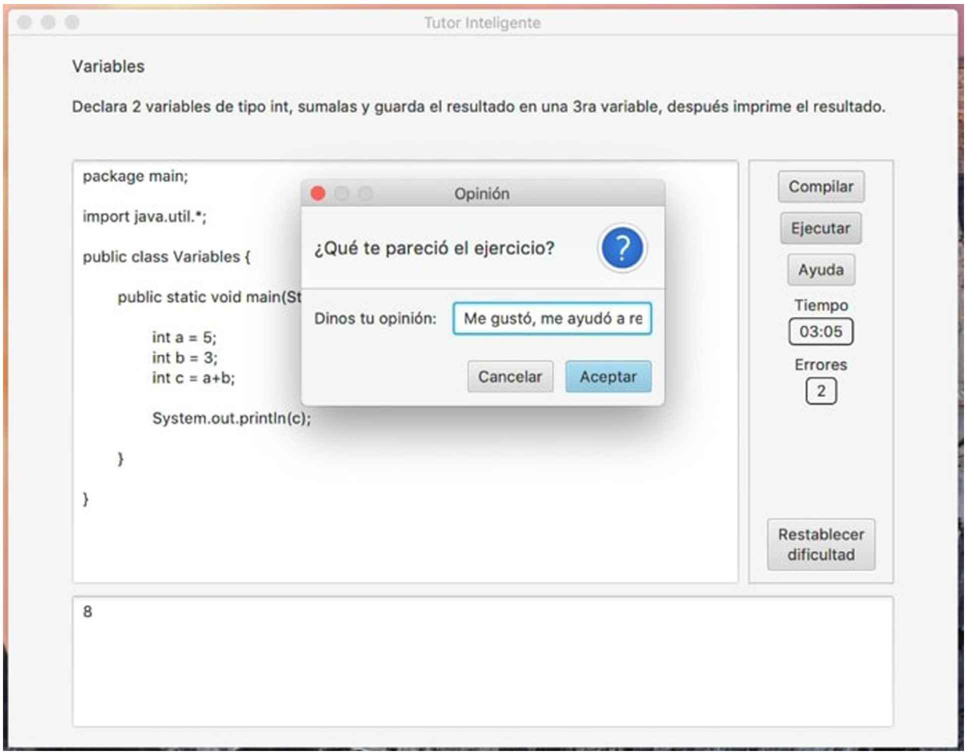


Fig. 4. Architecture of the ILE-Java



Variables

Declara 2 variables de tipo int, sumalas y guarda el resultado en una 3ra variable, después imprime el resultado.

```
package main;

import java.util.*;

public class Variables {

        public static void main(St

                int a = 5;
                int b = 3;
                int c = a+b;

                System.out.println(c);

        }

}
```

¿Qué te pareció el ejercicio?

Dinos tu opinión: Me gustó, me ayudó a re

```
8
```

Fig. 5. Example of the ILE interface

in the case of corpus eduSERE. With 70% of the opinions we created a set to train the models, and with 30%, a set to test them. As a result, the corpus sentiTEXT was left with a training set of 17,187 opinions and a testing set of 7365 opinions. In the case of eduSERE, given that the corpus is unbalanced, we restructured the three categories, leaving each of them with 3238 opinions. We used the same percentage of opinions to train and test as in sentiTEXT. The training set came with 6795 opinions and the testing set with 2917 opinions. Table 4 shows the results of the experiments performed with two corpus and fourteen classifiers.

In the tests carried out with corpus sentiTEXT, the best result was obtained (93%) with EvoMSA and BERT. However, the machine learning model B4MSA obtained a good and close 92%. It is interesting to note that three Machine Learning models

(B4MSA, linear SVC and SVC) obtained satisfactory results (90% and 92%).

The results obtained with corpus eduSERE are not as good as with sentiTEXT (one reason is the greater number of labels), and EvoMSA is the one that obtained the best results with 84%, followed by B4MSA and BERT with 83%. One reason for these somewhat low results is that each category of the corpus does not have enough phrases and that does not allow establishing differences between categories.

5. Model integration in an intelligent learning environment (ILE)

ILE-Java is an Intelligent Learning Environment for learning the Java programming language, which analyzes the cognitive and affective states of the student, to personalize their learning rate. The system consists of modules to control the user interface, to store the student's progress, to detect emotions based on text and faces, and to calculate the difficulty level of the exercises the student must solve. Fig. 4 shows the architecture of the system.

The user interface module is responsible for displaying the student's Java programming exercises in a code editor. The interface also supervises the time spend by the student to solve the exercise and the number of compilation and execution errors produced by the created program. The ILE stores this information in the storage module. To recognize student emotions in the text dialogues and on the face expression, the user interface uses an external module (an API REST). The fuzzy model will receive as input variables the recognized emotions together with the programming time, the errors, and the aids requested to the system. The fuzzy logic difficulty model is a module used to determine the difficulty (basic, intermediate, advanced) of the following exercise. Fig. 5 shows an example of the ILE interface. In this example, the main window shows on the top the problem description. Below the problem statement, there is a window where the student types the Java code. On the right side, we can observe several buttons to compile or execute the code, to ask for help, and some important elements like the time taken in coding, and the number of errors produced by the code. In the center of the window, we also show a dialogue box asking the student to write an opinion of the exercise. The text that the student types allows the system to recognize the polarity (positive / negative) of the opinion and the affective state of the student. Based on all the values observed by the program code (time, errors, petition for helps, opinion, and affective state), the ILE decides the level of difficulty (basic, intermediate, or advanced) of the next exercise. The ILE has 50 different exercises.

6. Discussions and conclusions

In this research work we sought to investigate the effectiveness (accuracy) of different methods to classify polarities and learning-centered emotions. To this end, we created one corpus for opinions labeled with polarities (sentiTEXT) and one corpus for opinions labeled with learning-centered emotions (eduSERE). The best model was EvoMSA, a multilingual sentiment classifier based on Genetic Programming with 93% for polarity recognizing and 84% for emotion recognizing. The advantage in performance of EvoMSA is that it incorporates extra knowledge like a human-annotated dataset, an own DeepEmoji implementation, and a lexicon model; these elements are combined with the dataset. This information is not used by any of the other machine learning techniques. In addition, EvoMSA uses a macro-F1 optimization that greatly improves its performance when working with datasets that have classes that are not balanced.

The results of this work surpass outcomes of other research works in which sentiment analysis or opinion mining in educational contexts was measured (Altrabsheh, Gaber, & Cocea, 2013; Ortigosa, Martín & Carro, 2014; Chaplot, Rhim, & Kim, 2015) and other related works in deep learning (Kim, 2014; Santos & Gatti, 2014; Wang, Liu, Sun, Wang, & Wang, 2015; Qian, Huang, Lei & Zhu, 2016). Further investigations comparing other classifications methods and other datasets are necessary to identify different points of improvements. On the other hand, we also need to increase the number of expressions or opinions of each of the two datasets to obtain better results both in training and in tests. Additionally, we need to conduct tests with students using the learning environment (ILE-Java) to evaluate the effectiveness of the use of polarity and emotion recognition in that ILE and perform tests (polarity and learning-centered emotions) of the classifiers with new datasets, of student opinion mining, about the performance of teachers in their different courses. We will carry out this experiment with groups of students who use the ILE without recognition of emotions and groups that use the ILE with recognition of emotions.

The results of this study must be appraised considering some limitations. First, it involved comparisons between only 14 different classification algorithms. We need to perform tests with more algorithms based on deep learning using different configurations of hyper-parameters and layers, based on evolutionary algorithms, and other different techniques. Second, there has been no evidence of the recognition of polarities and emotions other than in the laboratory. We need to conduct tests with students using new learning environments, which is mentioned before as future work.

Based on the results presented in this study, it can be concluded that the Evolutionary algorithm EvoMSA and Deep Learning (model BERT) were more effective than the traditional machine learning methods for classifying opinions in learning contexts. The results obtained by the classifiers depend very much on the quality of the expression datasets. Therefore, we can also conclude that the method to create such datasets was also important and valuable.

Finally, it is important to mention that EvoMSA has been tested on different application domains. For example, it has been tested in sentiment analysis competitions, obtaining competitive results on a variety of tasks and languages. It has been tested in the identification of positive or negative emotions that arose in the news (Martınez-Camara et al., 2018), the classification of aggressive texts (Álvarez-Carmona et al., 2018), and humor analysis (Castro, Chiruzzo, & Rosá, 2018). All these tasks were conducted in the Spanish language. In addition, EvoMSA has been applied to polarity detection in English and Arabic languages, (Rosenthal, Farra, & Nakov, 2019), and on different emotion detection tasks (Mohammad, Bravo-Marquez, Salameh & Kiritchenko, 2018) in English, Arabic, and Spanish languages.

Declaration of Competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Credit authorship contribution statement

María Lucía Barrón Estrada: Conceptualization, Supervision, Writing - original draft, Writing - review & editing. **Ramón Zatarain Cabada:** Methodology, Resources, Writing - original draft, Writing - review & editing. **Raúl Oramas Bustillos:** Software, Visualization, Investigation, Data curation, Formal analysis. **Mario Graff:** Software, Validation, Data curation, Resources, Formal analysis.

Acknowledgment

The work described in this paper is fully supported by a scholarship from CONACYT (Consejo Nacional de Ciencia y Tecnología) in Mexico and a grant from TECNM (Tecnológico Nacional de México) under the program Support for Scientific and Technological Research in the Educational Programs of the Federal Technological Institutes.

References

Altrabsheh, N., Cocea, M., & Fallahkhair, S. (2014). Learning sentiment from student's feedback for real-time interventions in classrooms. In *In adaptive and intelligent systems* (pp. 40–49). Springer.

Altrabsheh, N., Gaber, M., & Cocea, M. (2013). SA-E: sentiment analysis for education. *International Conference on Intelligent Decision Technologies, 255*, 353–362.

Baker, R. S. J. d., D'Mello, S. K., Rodrigo, M. M. T., & Graesser, A. C. (2010). Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive–affective states during interactions with three different computer-based learning environments. *International Journal of Human-Computer Studies, 68*(4), 223–241. doi:10.1016/J.IJHCS.2009.12.003.

Barrón-Estrada, M. L., Zataraín-Cabada, R., Bustillos, R. O., & Ramírez-Ávila, S. L. (2017). Building a corpus of phrases related to learning for sentiment analysis. *Research in Computing Science, 146*, 17–26.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*(8), 1798–1828.

Carvalho, J., Prado, A., & Plastino, A. (2014). A statistical and evolutionary approach to sentiment analysis. In *2014 IEEE/WIC/ACM international joint conferences on web intelligence (WI) and intelligent agent technologies (IAT)* (pp. 110–117). doi:10.1109/WI-IAT.2014.87.

Chaplot, D.S., Rhim, E., & Kim, J. (2015). Predicting Student Attrition in MOOCs using Sentiment Analysis and Neural Networks. AIED Workshops.

Dhanalakshmi, V., Bino, D., & Saravanan, A. M. (2016). Opinion mining from student feedback data using supervised learning algorithms. In *Big data and smart city (ICBDSC), 2016 3rd MEC international conference on* (pp. 1–5).

Diab, D. M., & El Hindi, K. M. (2017). Using differential evolution for fine tuning naïve Bayesian classifiers and its application for text classification. *Applied Soft Computing, 54*(C), 183–199. doi:10.1016/j.asoc.2016.12.043.

dos Santos, C., & Gatti, M. (2014). Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of {COLING} 2014, the 25th international conference on computational linguistics: technical papers* (pp. 69–78). Dublin: Ireland: Dublin City University and Association for Computational Linguistics.

El-Halees, A. (2011). Mining opinions in user-generated contents to improve course evaluation. In *International conference on software engineering and computer systems* (pp. 107–115).

Escalante, H. J., García-Limón, M. A., Morales-Reyes, A., Graff, M., Montes-y-Gómez, M., Morales, E. F., & Martínez-Carranza, J. (2015). Term-weighting learning via genetic programming for text classification. *Knowledge-Based Systems, 83*, 176–189. doi:10.1016/J.KNOSYS.2015.03.025.

Felbo, B., Mislove, A., Søgaard, A., Rahwan, I., & Lehmann, S. (2017). Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. 1615–1625. 10.18653/v1/D17-1169.

Graff, M., Tellez, E. S., Miranda-Jiménez, S., & Escalante, H. J. (2016). EvoDAG: A semantic Genetic Programming Python library. In *2016 IEEE international autumn meeting on power, electronics and computing (ROPEC)* (pp. 1–6). doi:10.1109/ROPEC.2016.7830633.

Graff, M., Miranda-Jiménez, S., Tellez, E.S., & Moctezuma, D. (2018). EvoMSA: {A} multilingual evolutionary approach for sentiment analysis. CoRR, abs/1812.0.

Graff, M., Tellez, E.S., Jair Escalante, H., & Miranda-Jiménez, S. (2017). Semantic genetic programming for sentiment analysis. 10.1007/978-3-319-44003-3_2

Hinojosa, J.A., Pozo, M.A., & Montoro, P.R. (2016). Affective norms of 875 Spanish words for five discrete emotional categories and two emotional dimensions. 272–284. 10.3758/s13428-015-0572-5

Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A convolutional neural network for modelling sentences. In *Proceedings of the 52nd annual meeting of the association for computational linguistics: 1* (pp. 655–665). Long Papers. doi:10.3115/v1/P14-1062.

Kechaou, Z., Ammar, M. B, & Alimi, A. M. (2011). Improving e-learning with sentiment analysis of user's opinions. In *Global engineering education conference (EDUCON), 2011 IEEE* (pp. 1032–1038).

Keshavarz, H., & Abadeh, M. S. (2017). ALGA: Adaptive lexicon learning using genetic algorithm for sentiment analysis of microblogs. *Knowledge-Based Systems, 122*, 1–16. doi:10.1016/J.KNOSYS.2017.01.028.

Kim, Y. (2014). Convolutional neural networks for sentence classification. 10.3115/v1/D14-1181.

Kumar, A., & Jain, R. (2015). Sentiment analysis and feedback evaluation. In *MOOCs, innovation and technology in education (MITE), 2015 IEEE 3rd international conference on* (pp. 433–436).

López, M., Valdivia, A., Martínez-Cámara, E., Luzón, M. V., & Herrera, F. (2019). E2SAM: Evolutionary ensemble of sentiment analysis methods for domain adaptation. *Information Sciences, 480*, 273–286. doi:10.1016/J.INS.2018.12.038.

Álvarez-Carmona, M. Á., Guzmán-Falcón, E., Montes-y-Gómez, M., Escalante, H. J., Villasenor-Pineda, L., Reyes-Meza, V., & Rico-Sulayes, A. (2018). Overview of MEX-A3T at IberEval 2018: Authorship and aggressiveness analysis in Mexican Spanish tweets. *Notebook Papers of 3rd SEPLN Workshop on Evaluation of Human Language Technologies for Iberian Languages (IBEREVAL)*: 6. Spain: Seville.

Castro, S., Chiruzzo, L., & Rosá, A. (2018). Overview of the HAHA task: Humor analysis based on human annotation at IberEval 2018. In *In IberEval@ SEPLN* (pp. 187–194).

Martínez Cámara, E., Almeida Cruz, Y., Díaz Galiano, M.C., Estévez-Velarde, S., García Cumbreras, M.Á., García Vega, M., et. al. (2018). Overview of TASS 2018: Opinions, health and emotions.

Mohammad, S., Bravo-Marquez, F., Salameh, M., & Kiritchenko, S. (2018). Semeval-2018 task 1: Affect in tweets. In *Proceedings of the 12th international workshop on semantic evaluation* (pp. 1–17).

Rosenthal, S., Farra, N., & Nakov, P. (2019). SemEval-2017 task 4: Sentiment analysis in Twitter. arXiv preprint arXiv:1912.00741

Mikolov, T., Chen, K., Corrado, G., Dean, J., Sutskever, L., & Zweig, G. (2014). word2vec. Google Scholar.

Nguyen, H., & Nguyen, M.-L. (2017). A deep neural architecture for sentence-level sentiment classification in twitter social networking. 10.1016/B978-0-444-51747-0.50005-6.

Onan, A., Korukoğlu, S., & Bulut, H. (2016). A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification. *Expert Systems with Applications, 62*, 1–16. doi:10.1016/J.ESWA.2016.06.005.

Ortigosa, A., Martín, J. M., & Carro, R. M. (2014). Sentiment analysis in Facebook and its application to e-learning. *Computers in Human Behavior, 31*, 527–541.

Pereira, C.N., & Valcárcel, R.R. (2016). Emocionario. V&R EDS.

Poli, R., Langdon, W.B., McPhee, N.F., & Koza, J.R. (2008). A field guide to genetic programming. Lulu. com.

Poria, S., Cambria, E., Hazarika, D., & Vij, P. (2016). A deeper look into sarcastic tweets using deep convolutional neural networks.

Qian, Q., Huang, M., Lei, J., & Zhu, X. (2016). Linguistically regularized lstms for sentiment classification. 10.18653/v1/P17-1154.

Rangel, I. D., Guerra, S. S., & Sidorov, G. (2014). Creación y evaluación de un diccionario marcado con emociones y ponderado para el español. *Onomázein: Revista de Lingüística, Filología y Traducción de La Pontificia Universidad Católica de Chile, 29*, 31–46.

Rani, S., & Kumar, P. (2017). A sentiment analysis system to improve teaching and learning. *Computer, 50*(5), 36–43.

Rowe, A. D. (2017). Feelings about feedback: The role of emotions in assessment for learning. In *Scaling up assessment for learning in higher education* (pp. 159–172). Springer.

Rowe, A., & Fitness, J. (2018). Understanding the role of negative emotions in adult learning and achievement: A social functional perspective. *Behavioral Sciences, 8*(2), 27. doi:10.3390/bs8020027.

Sermanet, P., Chintala, S., & LeCun, Y. (2012). Convolutional neural networks applied to house numbers digit classification. 10.0/Linux-x86_64.

Smith, T. C., Smith, T. C., & Witten, I. H. (1995). A genetic algorithm for the induction of natural language grammars. *Proc IJCAI-95 Workshop on New Approaches to Learning for Natural Language Processing, 17*, 17–24. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.51.4685 .

Spector, L., Langdon, W. B., & O'Reilly, U.-M. (1999). *Advances in genetic programming*: 3. MIT Press.

Tellez, E. S., Miranda-Jiménez, S., Graff, M., Moctezuma, D., Suárez, R. R., & Siordia, O. S. (2017). A simple approach to multilingual polarity classification in Twitter. *Pattern Recognition Letters, 94*, 68–74. doi:10.1016/j.patrec.2017.05.024.

Tellez, E. S., Moctezuma, D., Miranda-Jiménez, S., & Graff, M. (2018). An automated text categorization framework based on hyperparameter optimization. *Knowledge-Based Systems, 149*, 110–123. doi:10.1016/j.knosys.2018.03.003.

Tul, Q., Ali, M., Riaz, A., Noureen, A., Kamranz, M., Hayat, B., & Rehman, A. (2017). Sentiment analysis using deep learning techniques: A review. *International Journal of Advanced Computer Science and Applications, 8*(6). doi:10.14569/IJACSA.2017.080657.

Wang, J., Yu, L.-C., Lai, K. R., & Zhang, X. (2016). Dimensional sentiment analysis using a regional CNN-LSTM Model. In *Proceedings of the 54th annual meeting of the association for computational linguistics (Volume 2: Short Papers), (January)* (pp. 225–230). doi:10.18653/v1/P16-2037.

Wang, X., Liu, Y., Sun, C., Wang, B., & Wang, X. (2015). Predicting polarities of tweets by composing word embeddings with long short-term memory. 1343–1353.

Wen, M., Yang, D., & Rose, C. P. (2014). Sentiment Analysis in MOOC Discussion Forums: What does it tell us? In J. Stamper, et al. (Eds.), *Proceedings of the 7th International Conference on Educational Data Mining* (pp. 130–137). Pearson.

Winkler, S., Schaller, S., Dorfer, V., Affenzeller, M., Petz, G., & Karpowicz, M. (2015). Data-based prediction of sentiments using heterogeneous model ensembles. *Soft Computing, 19*(12), 3401–3412. doi:10.1007/s00500-014-1325-6.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks, 5*(2), 241–259. doi:10.1016/S0893-6080(05)80023-1.

Zatarain Cabada, R., Rodriguez Rangel, H., Barron Estrada, M. L., & Cardenas Lopez, H. M. (2019). Hyperparameter optimization in CNN for learning-centered emotion recognition for intelligent tutoring systems. *Soft Computing*. doi:10.1007/s00500-019-04387-4.

Zhang, L., & Liu, B. (2016). Sentiment analysis and opinion mining. *Encyclopedia of Machine Learning and Data Mining*, (May), 1–10. doi:10.1007/978-1-4899-7502-7_907-1.

Zhang, L., Wang, S., & Liu, B. (2018). Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*. doi:10.1002/widm.1253.

Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. In C. Cortes, D. D. Lee, M. Sugiyama, & R. Garnett (Eds.), *Proceedings of the 28th International Conference on Neural Information Processing Systems*. MIT Press.