Sentiment Analysis in an Affective Intelligent Tutoring System

[1]María Lucía Barrón-Estrada, [1]Ramón Zatarain-Cabada, [2]Raúl Oramas-Bustillos, [1]Francisco González-Hernández. [1]Instituto Tecnológico de Culiacán, [2]Universidad de Occidente, Unidad Culiacán, Culiacán, Sinaloa, México. lbarron@itculiacan.edu,mx

Abstract—This paper presents an implementation of Sentiment Analysis module that works in an Affective Intelligent Tutoring System. Later, we show the results of evaluating this module and some experiments we made with students. We consider that the main contribution of this work is that with this module, the system will be able to analyze the student's feedback using sentiment analysis techniques and provide useful information to the course administrators teachers for improving the teaching material.

Keywords— Student's feedback, Sentiment Analysis, Machine Learning, Naïve Bayes Classifier, Affective Computing.

I. Introduction

For Virtual Learning Environments feedback is a communication link between the teacher (tutor or facilitator) and the student. The teacher typically uses feedback survey as an evaluation instrument for their virtual courses because feedback is a key element in learning and teaching [1][2].

Students feedback can help to understand the student learning behavior [3][4]. This feedbacks contain open-ended student opinions and would provide meaningful insights for the teachers to incorporate changes and improve their courses contents and other teaching elements.

The growth of user-generated content on websites and social networks, such as Facebook, Twitter or Amazon, has changed the way that people express their opinions and has turned online opinions into a very valuable asset with many practical applications.

Sentiment Analysis (SA) is one of the most interesting applications of Natural Language Processing (NLP) used in social media applications, whose goal is the evaluation and classification of text with emotionally charged language which expresses or implies positive or negative sentiments [5].

In this paper, we describe a Sentiment Analysis module that works inside Java Sensei [6], an Affective Intelligent Tutoring System (AITS) to find out student's opinions about the course contents. The module keeps track the student's opinions (feedbacks) and by using sentiment analysis techniques it finds their positive or negative feelings that they have towards the current material teaching. With this feedback, the course administrators of Java Sensei can find out if the course contents need improvements.

This paper is organized as follows: Section 2 describes related work of Sentiment Analysis. Section 3 describes Sentiment Analysis Techniques, Section 4 presents the Sentiment Analysis module in an Intelligent Tutoring System. Section 4 discuss the results produced with the system. Finally, the conclusions and future work are discussed in Section 5.

II. Related Work

The sentiment classification research is mainly at the document or sentence level, clause, phrase, and word, depending on the specific objectives of applications. As mentioned in [5][7][8] to solve the sentiment classification problem, these SA approaches have been used: Machine Learning (ML), lexicon-based, and hybrid approach.

The ML approach frequently uses the Bag of Word (BOW) model, where a text (such as a sentence or document) is represented as a feature vector and then classified by machine learning techniques. The bag of word model omits grammar and word order but is interested in the number of occurrences of words within the text.

The lexicon-based approach relies on a sentiment lexicon, a collection of known and precompiled sentiment terms. It is divided into the dictionary-based approach and corpus-based approach which use statistical or semantic methods to find sentiment polarity. The hybrid approach combines both approaches and is very common with sentiment lexicons playing a key role in the majority of methods.

The text classification methods using ML approach can be divided into supervised and unsupervised learning methods. The supervised methods make use of a large number of labeled training documents. The unsupervised methods are used when it is difficult to find these labeled training documents.

The most common technique in Sentiment Analysis is supervised classifiers like Naïve Bayes Classifier (NB), Bayesian Network (BN), Maximum Entropy Classifier (ME) and Support Vector Machines Classifiers (SVM).

We found several works in the educational area that research students feedback and made Sentiment Analysis, for example, SA-E, Sentiment analysis for education [3], presents an architecture for analyzing student's feedback using sentiment analysis techniques using Twitter. Another work is Sentiment analysis in Facebook and its application to e-learning [9], presents SentBuk, a Facebook application that retrieves messages written by users in Facebook and classifies them according to their polarity. In Mining sentiments in SMS texts for teaching evaluation [10], it explores the

2161-377X/17 $31.00 © 2017 IEEE
DOI 10.1109/ICALT.2017.137

potential application of sentiment mining for analyzing short message service. Other related works are Feelings about feedback, the role of emotions in assessment for learning [1], Learning Sentiment from Students' Feedback for Real-Time Interventions in Classrooms [11] and Exploiting Sentiment Analysis to Track Emotions in Student's Learning Diaries [12].

III. SENTIMENT ANALYSIS IN JAVA SENSEI ENVIRONMENT

In 2015, a Web-based environment (http://javasensei.ddns.net/javasensei) called Java Sensei was reported [6] that aims to provide adapted and individualized programming instruction to students by using modern learning technologies as a recommender and mining system, an affect recognizer, a sentiment analyzer, and an authoring tool. For the purposes of this paper, we will focus how we build the Sentiment Text Recognition module.

The idea behind Sentiment Text Recognition module is to take the students opinions about the system programming exercises, and then, the module will be made Sentiment Analysis to determine if the exercise's polarity is positive (like) or negative (not like). With this information, the course administrators and teachers will be able to evaluate the student's opinion and determine the programming exercises quality and then, they will be able to know how to improve the programming exercises based on student's opinion. Fig. 1 shows the idea about how to work Sentiment Analysis in Java Sensei.

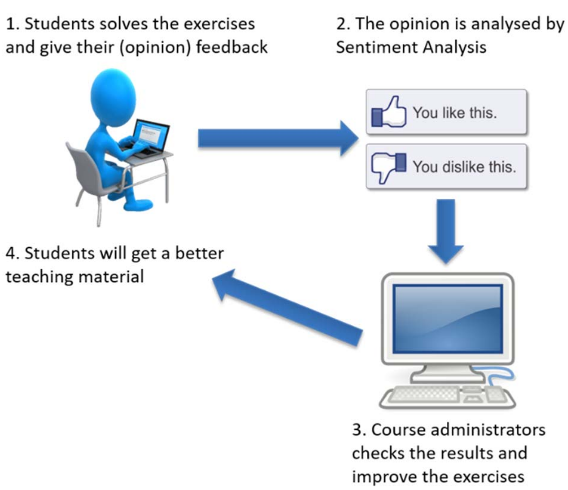

Fig. 1. Sentiment Analysis Flow.

IV. SENTIMENT ANALYSIS MODULE

The major challenges to design the Sentiment Analysis module was how to collect and manage the student's opinions, turning it into meaningful numbers of some sort and analyzing them. To achieve this goal, first, we get a text corpus (collection of texts), then pre-processing them, convert into numerical feature vectors (an n-dimensional vector) for using it in text classifier for training and predicting text polarity. Fig 2. shows this algorithm.

It is important to have a text corpus classified with text (opinion) and its polarity (positive, negative). For this work, TASS Corpus [13] was used. This corpus contains over 68 000 Twitter messages in XML format, written in Spanish by about 150 well-known personalities and celebrities of the world of politics, economy, communication, mass media, and culture, between November 2011 and March 2012.

The TASS corpus needs to be pre-processed in order to transform text into numbers and get the representative vectors for each tuple (polarity, opinion). The steps of the corpus preprocessing are:

- Slang-terms: translate slang terms and emoticons to equivalent text. For example "+1" are translated as "I like" (it means: "me gusta" in Spanish).
- Tokenization: convert sentences into words, removes punctuations, and symbols
- Stemming: reduces the word to its root, e.g., runner to run.
- Filter stop words: remove useless words –as, from, on, the, to, in, etc., from the corpus

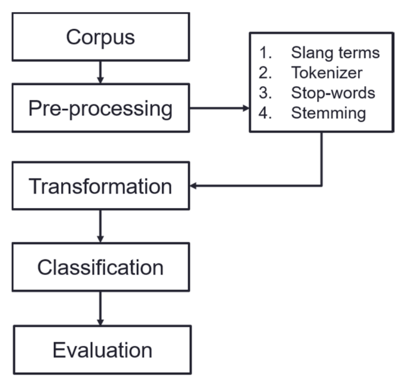

Fig 2. Sentiment Analysis Algorithm.

Fig. 3 shows and example of the corpus preprocessing.

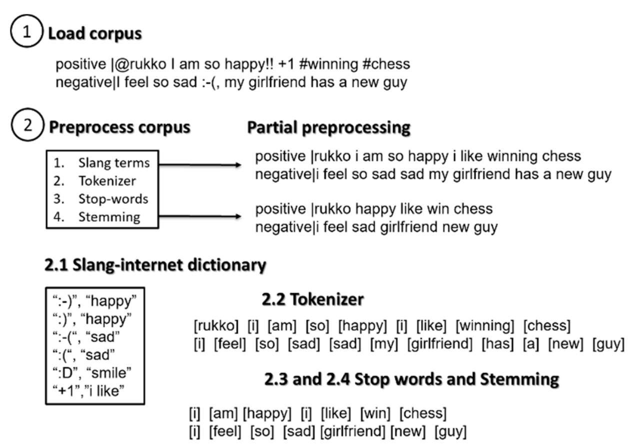

Fig. 3. Corpus preprocessing.

For transformation, we generated TF-IDF matrix that computes term frequency and inverse document frequency for each word in the corpus. This is called Weighting features extraction. Term Frequency (TF) is the number of times that a given term appears in a document or dataset, which allows being represented. Inverse Document Frequency (IDF) computes the number of documents in which a given term appears. This measure allows determining how discriminative a given term is. Rare terms are more discriminative than common terms. Next, we constructed the model learning, known as supervised learning where, given a labeled dataset, the task is to learn a function that will predict the label given the input. We used the Naïve Bayes Classifier because is the simplest and most commonly used classifier. As mentioned in [8], a Naïve Bayes classification model computes the posterior probability of a class, based on the distribution of the words in the document. The model works with the Bag of Bows (BOWs) feature extraction which ignores the position of the word in the document. It uses Bayes Theorem (1) to predict the probability that a given feature set belongs to a particular label:

$$P(label\,|features) = \frac{\mathrm{P(label)*P(features|label)}}{\mathrm{P(features)}} \qquad (1)$$

P(label) is the prior probability of a label or the likelihood that a random feature set the label. P(features|label) is the prior probability that a given feature set is being classified as a label. P(features) is the prior probability that a given feature set occurs.

V. Results

A. Evaluation of machine learning model.

First, we have evaluated the machine learning model for predict the polarity (positive, negative) of an input text. The result of this evaluation it is important because it helps us to find how trustworthy the model is. We used cross-validation model technique to estimate how accurately our model is.

We have got a dataset of 26661 tweets after processing TASS corpus. We use 60% of these data to train our model with Naïve Bayes Classifier and 40% for testing. Fig 4. Show the results of applying the cross-validation test.

```
Init predictor...
('Score:', 0.80750909568283258)
                  precision    recall    f1-score    support
   negative          0.82       0.88       0.85       16647
   positive          0.77       0.69       0.73       10014

   avg/total         0.81       0.81       0.81       26661

Accuracy  80.75%
End predictor...
```

Fig.4 Results of testing the machine learning model.

We got an accuracy of 80.75% that it means that we got a correct prediction about text polarity in the 80.75% of the cases. The precision measure 0.81 it indicates us how many of the tweets that we identified were relevant is true-positive. The recall measure 0.81 it indicates us how many of the relevant tweets that we identified is true-positive. Finally, the f1-score that combines the precision and recall give us a 0.81 value that can be considered as a good measure.

B. Testing the Sentiment Analysis module.

The system has been tested with 43 students in our research lab in Instituto Tecnológico de Culiacan in Mexico. In Java, Sensei students must solve several exercises to learn and practice the topics of Java language. Fig. 5 shows the Web environment that represents the student's graphical user interface (GUI) with a menu organized by lessons.

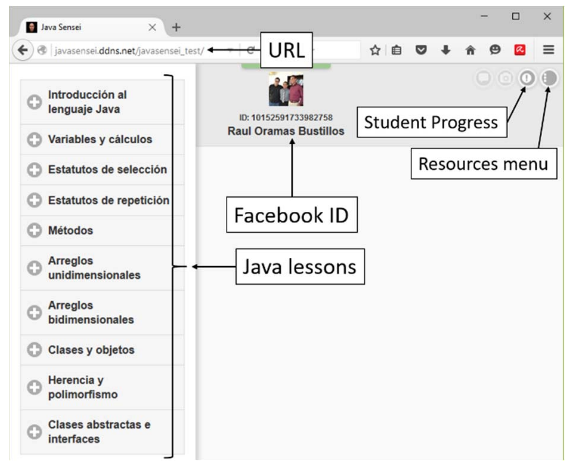

Fig. 5 Java Sensei Environment in a Web browser.

Each lesson has 15 exercises. At the end of each exercise, the system asks for student opinion. Table 1, shows some opinions recollected.

We recollected 178 students opinions were 71 texts were evaluated as positive and 107 as negative by the Sentiment Text Recognition Module. This results will be kept for further study by course administrators and teachers and it will help to improve the teaching material of the Java Sensei.

TABLE I
Opinions Collected.

Opinion in Spanish	Opinion in English	Eval Polarity
No entendí este ejercicio	I don't understand this exercise	Negative
Me gusto este ejercicio	I like this exercise	Positive
No tenia mucho conocimiento sobre el tema	I don't have a lot of knowledge about this topic	Positive
El ejercicio es confuso	This exercise is confuse	Negative
Este ejercicio es sencillo de entender	This exercise is easy to understand	Positive

VI. Conclusions and Future Work

In this paper, we have described a general framework to integrate Sentimental Analysis in the AITS Java Sensei. Evaluation results with students give us a lot of information to improve exercises quality of Java Sensei.

For future work, we want to make new improvements in the software to include multimodal affect recognition, identification of emotion by text, voice, and EEG devices. Also, we are working to develop a better corpus that includes typical programming phrases and more Spanish slang terms. Finally, we want to test the AITS with the new improvements.

Acknowledgment

The work described in this paper is fully supported by a grant from the TECNM and CONACYT in México.

References

[1] A. D. Rowe, "Feelings About Feedback: The Role of Emotions in Assessment for Learning," in *Scaling up Assessment for Learning in Higher Education*, D. Carless, S. M. Bridges, C. K. Y. Chan, and R. Glofcheski, Eds. Singapore: Springer Singapore, 2017, pp. 159–172.

[2] A. Poulos and M. J. Mahony, "Effectiveness of feedback: the students' perspective," *Assess. Eval. High. Educ.*, vol. 33, no. 2, pp. 143–154, Apr. 2008.

[3] N. Altrabsheh, M. M. Gaber, and M. Cocea, "SA-E: Sentiment analysis for education," *Front. Artif. Intell. Appl.*, vol. 255, pp. 353–362, 2013.

[4] S. Cummins, L. Burd, and A. Hatch, "Using Feedback Tags and Sentiment Analysis to Generate Sharable Learning Resources Investigating Automated Sentiment Analysis of Feedback Tags in a Programming Course," in *2010 10th IEEE International Conference on Advanced Learning Technologies*, 2010, pp. 653–657.

[5] Z. Yan-Yan, Q. Bing, and L. Ting, *Sentiment Analysis*, vol. 21, no. 8. Cambridge University Press, 2015.

[6] R. Z. Cabada, M. L. B. Estrada, F. G. Hernandez, and R. O. Bustillos, "An Affective Learning Environment for Java," *2015 IEEE 15th Int. Conf. Adv. Learn. Technol.*, pp. 350–354, 2015.

[7] G. Wang, J. Sun, J. Ma, K. Xu, and J. Gu, "Sentiment classification: The contribution of ensemble learning," *Decis. Support Syst.*, vol. 57, no. 1, pp. 77–93, 2014.

[8] B. Liu, *Web Data Mining*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.

[9] A. Ortigosa, J. M. Martín, and R. M. Carro, "Sentiment analysis in Facebook and its application to e-learning," *Comput. Human Behav.*, vol. 31, no. 1, pp. 527–541, 2014.

[10] C. K. Leong, Y. H. Lee, and W. K. Mak, "Mining sentiments in SMS texts for teaching evaluation," *Expert Syst. Appl.*, vol. 39, no. 3, pp. 2584–2589, 2012.

[11] N. Altrabsheh, M. Cocea, and S. Fallahkhair, "Learning Sentiment from Students' Feedback for Real-Time Interventions in Classrooms," in *Adaptive and Intelligent Systems: Third International Conference, ICAIS 2014, Bournemouth, UK, September 8-10, 2014. Proceedings*, A. Bouchachia, Ed. Cham: Springer International Publishing, 2014, pp. 40–49.

[12] M. Munezero, C. S. Montero, M. Mozgovoy, and E. Sutinen, "Exploiting Sentiment Analysis to Track Emotions in Students' Learning Diaries," in *Proceedings of the 13th Koli Calling International Conference on Computing Education Research*, 2013, pp. 145–152.

[13] J. Villena-román, E. Martínez-cámara, S. Lana-serrano, and J. C. González-cristóbal, "TASS - Workshop on Sentiment Analysis at SEPLN TASS - Taller de Análisis de Sentimientos en la SEPLN," pp. 37–44, 2013.